FactoryTalk View Machine Edition User's Guide

Version 15.00.00

Rockwell Automation

Important User Information

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

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

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

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

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

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

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

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

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

ATTENTION: Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence.

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

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

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

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

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

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

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

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

Contents

Preface

FactoryTalk View ME User's Guide provides comprehensive information about FactoryTalk View ME, procedures for creating and running an automation application, and reference information.

Find information on the Internet

If you cannot find the answer in the User's Guide or Help, you can find information on the Internet.

- Rockwell Automation Home Page
- Rockwell Automation Downloads includes the latest add-on profiles (AOPs), drivers, macros, electronic data sheets, sample code, and software activations.
- Rockwell Automation Knowledgebase provides a comprehensive and searchable database of support information for all Rockwell Automation products.
- FactoryTalk View Series Home Page

To connect to web sites from FactoryTalk View Studio

- From the **Help** menu, select **Rockwell Automation on the Web**, and then select the name of the web page you want to view.

Contact Rockwell Automation Technical Support

If you cannot find the answer to your question in the documentation or on the Internet, contact Rockwell Automation.

Customer Support Telephone — 1.888.382.1583 or 1.440.646.3434

When you call, it is recommended that you be at your computer and ready to provide the following information:

- The product's serial number and product key, which are printed on the Activation Certificate enclosed with the product software DVD
- FactoryTalk View version number
- FactoryTalk Services Platform version number
- FactoryTalk Linx version number
- The type of hardware you are using
- The names of other Rockwell Automation products installed on the computer
- The exact wording of any messages shown on the computer
- A description of what happened and what you were doing when the problem occurred
- A description of how you tried to solve the problem

To view the list of installed add-ons and updates

1. In FactoryTalk View Studio, select **Help** > **About FactoryTalk View Studio**.
2. To view the list of installed add-ons, click **Add-Ons**.

Legal Notices

Rockwell Automation publishes legal notices, such as privacy policies, license agreements, trademark disclosures, and other terms and conditions on the Legal Notices page of the Rockwell Automation website.

Software and Cloud Services Agreement

Review and accept the Rockwell Automation Software and Cloud Services Agreement here.

Open Source Software Licenses

The software included in this product contains copyrighted software that is licensed under one or more open source licenses.

You can view a full list of all open source software used in this product and their corresponding licenses by opening the oss_license.txt file located in your product's OPENSOURCE folder on your hard drive. This file is divided into these sections:

- Components
  Includes the name of the open source component, its version number, and the type of license.
- Copyright Text
  Includes the name of the open source component, its version number, and the copyright declaration.
- Licenses
  Includes the name of the license, the list of open source components citing the license, and the terms of the license.

The default location of this file is:

C:\Program Files (x86)\Common Files\Rockwell\Help\<*product name*>\Release Notes\OPENSOURCE\oss_licenses.txt.

You may obtain Corresponding Source code for open source packages included in this product from their respective project web site(s). Alternatively, you may obtain complete Corresponding Source code by contacting Rockwell Automation via the **Contact** form on the Rockwell Automation website: http://www.rockwellautomation.com/global/about-us/contact/contact.page. Please include "Open Source" as part of the request text.

Commercial Software Licenses

This software also includes these commercially licensed software components:

Component	Copyright
PDFControls.NET 2.0.114	Copyright 2006-2021 TallComponents BV.
Symbol Factory 3.01.94	Copyright 1996-2023 Software Toolbox, Inc.

Other Third-Party Free Software Licenses

This software also includes these free software components:

Component	Copyright
SQL Server LocalDB 2022 U13	Copyright Microsoft. All rights reserved.
FarPoint Spread 3.0	Copyright @1991-1999 FarPoint Technologies Incorporated
Microsoft Visual C++ 2015-2022 Redistributable	Copyright (c) Microsoft Corporation. All rights reserved.
Microsoft .NET Framework 4.8	Copyright Microsoft. All rights reserved.
VBA 7.1.11	Copyright @2018 Microsoft Corporation. All rights reserved.

Get Started

FactoryTalk® View Machine Edition (ME) software is a versatile HMI application that provides a dedicated and powerful solution for machine-level operator interface devices. As an integral element of the Rockwell Automation visualization solution, FactoryTalk View ME provides excellent graphics, runtime user management, language switching, and faster commissioning time through a common development environment.

FactoryTalk View ME allows for a consistent operator interface across multiple platforms, including PanelView Plus and Windows desktop solutions, and consists of:

- *FactoryTalk View Studio*. Configuration software for developing and testing HMI applications.
- *FactoryTalk View ME Station*. A traditional standalone HMI solution that provides an integrated operator interface that runs the same HMI application on PanelView Plus 7, PanelView Plus 6, MobileView terminal, Rockwell Automation Industrial Computers, or other industrial PC.

The FactoryTalk View ME software package also includes:

- FactoryTalk Services Platform
- FactoryTalk Administration Console
- FactoryTalk Diagnostics
- FactoryTalk Activation Manager
- FactoryTalk Linx
- FactoryTalk Updater

For more information about each component, see FactoryTalk View ME Installation Guide.

FactoryTalk View Studio tools

These tools are installed with FactoryTalk View Studio:

- **Application Documenter** is software that provides detailed information about FactoryTalk applications.
- **FactoryTalk View ME Application Manager** is software for renaming, copying, deleting, backing up, and restoring applications.
- **DeskLock** is software that prevents Windows operating system users from exiting the FactoryTalk View application at runtime. You can use this tool to set up a customized desktop and to prevent access to Windows functions. You cannot use DeskLock if you are running your application on a PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal.
- **FactoryTalk View File Viewer** is software for viewing the following types of log files:
  - SE datalog files
  - ME datalog files
  - ME alarm log files
  - dBase files
  - ME RecipePlus files

  The log information can be shown in the form of list views or trend plots. You can save the files as CSV or dBase files so that it is easier to do more analysis.
- **Legacy Tag Database Conversion** is software for converting your legacy tag database to a new format to work with Microsoft SQL database.
- **ME Firmware Upgrade Wizard** is software for installing new firmware releases on PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, and PanelView Plus CE terminals.

- **ME Transfer Utility** is software for transferring a runtime application file (.mer) between the development computer and a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, transferring Windows True Type font files (.ttf and .ttc) to a PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, and comparing a runtime application on a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal with a runtime application on a desktop computer.
- **Tag Import and Export Wizard** is software for importing or exporting the FactoryTalk View tag database.
- **Rockwell Automation CSV Data Log Tamper Detection Utility** is software used to detect the integrity of a CSV data log file that contains tamper detection strings, generate a detection report, and save the report to a text file.
  You can install this utility independently from the path `FTView-DVD\<version>-FTView\Tools\CSV Data Log Tamper Detection Utility`. It doesn't have any version dependencies on FactoryTalk View ME.

To start FactoryTalk View Studio tools

- From Windows **Start** menu, select **All apps > Rockwell Software >** *ToolName*.

You can also select a tool from the **Tools** menu in FactoryTalk View Studio.

Diagnostics Viewer

Diagnostics Viewer allows you to view FactoryTalk Diagnostics messages using the FactoryTalk Diagnostics Viewer.

To start Diagnostics Viewer

- From Windows **Start** menu, select **All apps** > **Rockwell Software** > **Diagnostics Viewer**.

You can also select the tool from the **Tools** menu in FactoryTalk View Studio.

FactoryTalk tools

FactoryTalk tools are installed when you install FactoryTalk Services Platform.

To start FactoryTalk tools

- From Windows **Start** menu, select **All apps > Rockwell Software >** *ToolName*.

FactoryTalk Activation Manager

All FactoryTalk View software must be activated using FactoryTalk Activation Manager.

FactoryTalk Activation Manager is automatically installed when you install FactoryTalk View.

To start FactoryTalk Activation Manager

- From Windows **Start** menu, select **All apps > Rockwell Software >** *ToolName*.

Explore FactoryTalk View Studio

This chapter describes how to:

- Start FactoryTalk View Studio
- Open sample applications
- Explore the FactoryTalk View Studio main window
- Use the **Explorer** window
- Work with editors
- Print information in editors

For information about working with editors' components, see Work with components on page 243.

Start FactoryTalk View Studio from the Windows Start menu

You can start FactoryTalk View Studio from the Windows **Start** menu.

To start FactoryTalk View Studio

- From the Windows **Start** Menu, select **All apps** > **Rockwell Software** > **FactoryTalk View Studio**.

FactoryTalk View comes with several sample applications. We suggest you open the InstantFizz_ME sample application now to use while you try out the instructions in the remainder of this chapter. See Open sample applications on page 35.

Start FactoryTalk View Studio using a command

You can start FactoryTalk View Studio using a command.

To start FactoryTalk View Studio using a command

1. In the Windows Run command, enter a command with the following syntax:
   (64-bit OS) `"C:\Program Files (x86)\Rockwell Software\RSView Enterprise\VStudio.exe" /o /type:application_type /app:application_name [/lang:startup_language]`
   where *C:* is the drive to which FactoryTalk View is installed.
2. Press **Enter**.

Command-line parameters and examples

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

Parameters	Descriptions
/O	Starts FactoryTalk View Studio.
/Type:*value*	Specifies the application type. The *value* is one of: • *sed* – SE Network Distributed • *sen* – SE Network Station

Parameters	Descriptions
	• *sel* – SE Local Station • *med* – Machine Edition
/App:*value*	Specifies the application to be opened.
/Lang:*value*	Specifies the startup language. The value is a Locale ID in Decimal.

The following examples show how to use the commands.

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

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

- To start FactoryTalk View Studio for ME and open the *Objects 5 Screen Demo* application in French (France), type the following command:

  ```
  "C:\Program Files\Rockwell Software\RSView Enterprise\VStudio.exe" /o /
  type:med /app:"Objects 5 Screen Demo" /Lang:1036
  ```

Open sample applications

When you first install FactoryTalk View, sample applications are accessible to all users. For information about setting up users and granting them access to applications, see Set up security on page 143.

To open the InstantFizz_ME sample application

1. Start FactoryTalk View Studio.
2. On the **Existing** tab, select **InstantFizz_ME**.
3. Specify a language for the application and click **Open**.

FactoryTalk View Studio shows the sample application in the **Explorer** window.

HMI projects folder settings

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

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

- NT AUTHORITY\IUSR: Full control

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

- NT AUTHORITY\SYSTEM: Full control
- NT AUTHORITY\LOCAL SERVICE: Full control

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

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

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

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

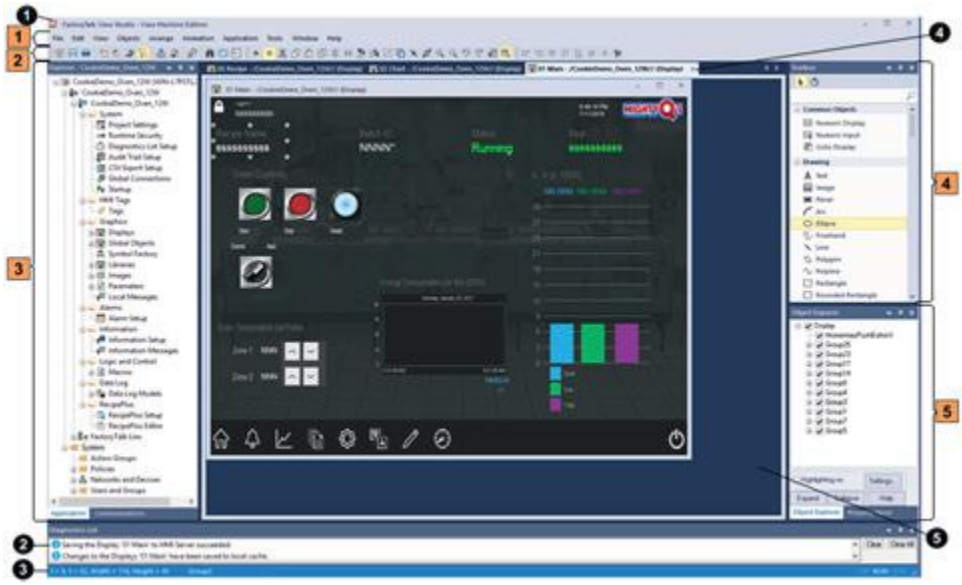
- Remove the **INTERACTIVE** group from the folder's security properties.
- Add specific users or user groups and assign their permissions to this folder as needed.

For more information about security, see Set up security on page 142.

For more information about customizing the location of the HMI projects folder, see Help.

FactoryTalk View Studio main window

When you start FactoryTalk View Studio, you can open an existing application or create a new one using the items from the **File** menu. Once you open an application, the application appears in the **Explorer** window on the left side of the main window.



Item	Name	Description
1	Menu bar	It contains the menu items for the active window. Each editor has its own set of menus.
2	Toolbar	It contains commonly used menu items providing quick access to the items without opening the menu. All editors use the **Standard** toolbar. The **Graphic Displays**, **Graphic Libraries**, **Runtime Security**, and **Tags** editors have additional toolbars.
3	Explorer window	It includes the **Application** tab and the **Communications** tab.
4	Toolbox	It is a searchable toolbox containing all project elements. Right-click in the toolbox to add frequently used objects to Common Objects or to show the objects in a list view.
5	Object Explorer and Property Panel	It includes **Object Explorer** and **Property Panel**. **Object Explorer** provides a list of all the objects on the current graphic display. **Property Panel** provides the functions to configure graphic objects. You can group or ungroup **Object Explorer** and **Property Panel**.
❶	Title bar	It shows the program name.
❷	Diagnostics list	It shows messages about system activities. You can specify the types of messages to show in the Diagnostics List, move the list, resize it, and clear the messages in it.
❸	Status bar	It shows information about the active window, the selected tool or menu item, or the selected graphic object, such as its size and position.
❹	Workbook tabs	If you select **Workbook Mode** from the **View** menu, the workspace shows tabs on the top of each open editor or component.
❺	Workspace	It is the blank area of the FactoryTalk View Studio window. You can drag icons from the **Explorer** window to the workspace to open editors and components.

To show or hide the items in the main window (except for the menu bar), select the items from the **View** menu.

To undock Explorer, Diagnostics List, toolbox, Object Explorer, or Property Panel from the main window, drag the title bar of each window and move the window anywhere on the screen. When a window is undocked, you can resize it as needed. For example, to show more messages in the Diagnostics List at a time, drag an edge or corner of the window until it is the size you want.

To undock a toolbar, drag the grab bar on the left.

For more information about using the Explorer window, see Use the Explorer window on page 38.

For more information about specifying the types of messages to show, see Message routing on page 127.

Explorer window

The **Explorer** window is the main tool for working with FactoryTalk View Studio. It lists the editors you use to develop your application, as well as the components, such as graphic displays.

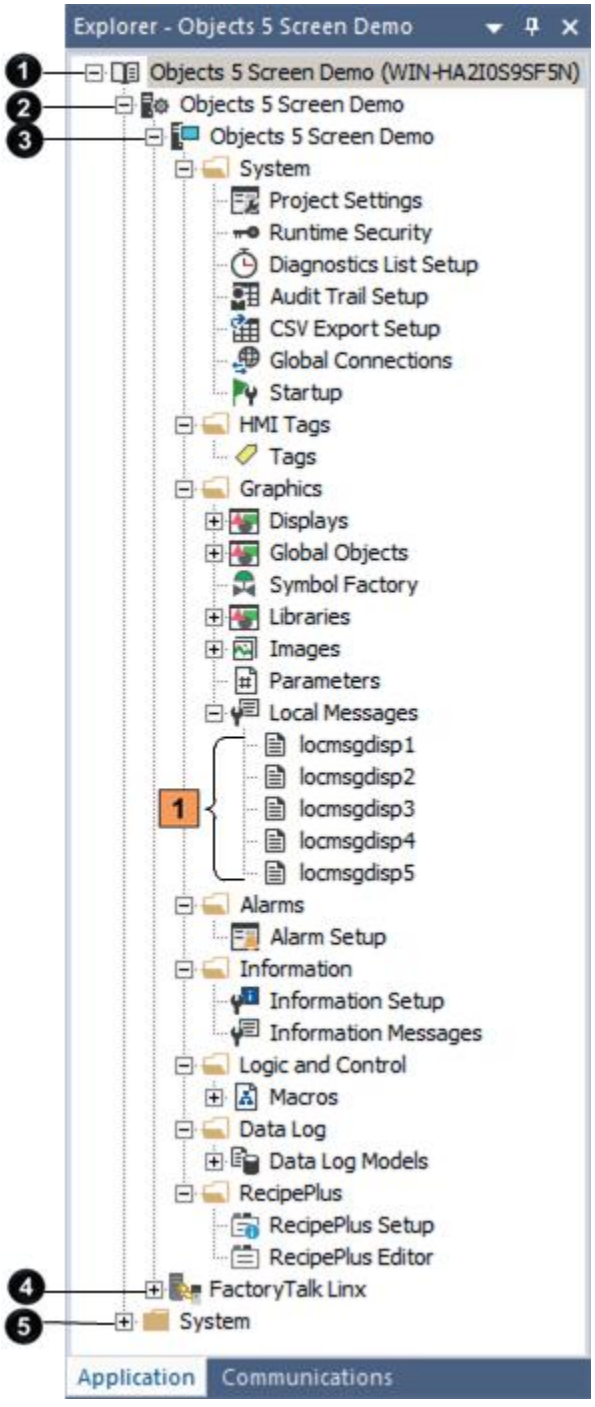
The **Explorer** window has two tabs:

- The **Application** tab contains the editors for creating and editing your application, and the **System** folder for setting up FactoryTalk Security.
- The **Communications** tab contains the communications tree for the development computer. For more information about using the communications tree, see Help for FactoryTalk Linx.

For information about setting up security, see Set up security on page 142.

An application consists of one or more data servers and an HMI project (also known as an HMI server). The data servers provide communications for the project. The project consists of graphic displays, alarm information, user information, and other settings.

This manual generally uses the term *application* to refer to both application-level and project-level procedures.



Item	Description
1	Components are listed below an editor's icon.
1	FactoryTalk Directory
2	Application
3	Application's HMI server. It contains the editors for setting up the application and components.
4	FactoryTalk Linx data server

Item	Description
5	This folder contains FactoryTalk Security policies and users.

Work with 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** menu, select the editor you want to open.

For information on working with particular editors, see the chapters later in this guide.

View an editor's components

With some editors, you enter information in a single window or a tabbed dialog box. Other editors allow you to create multiple components, such as graphic displays or message files. Each component is stored in a separate file, with its own name.

You can create components in these editors:

- **Graphics** (components include graphic displays, global object displays, and graphic libraries, each in their own folder)
- **Alarms**
- **Parameters**
- **Local Messages**
- **Information Messages**
- **Data Log Models**
- **Macros**
- **RecipePlus**

The **Explorer** window lists the components you create under the icon for the editor you used to create the component.

To view a list of components for an editor

- Click the + symbol to the left of the editor icon, or double-click the editor name.

Print an editor's contents

At design time, you can print an editor's contents.

Prerequisites

- From the **File** menu in any editor or component, select **Print Setup** and specify the printer.

To print an editor's contents

1. Open the component.
2. From the **File** menu, select **Print**, or on the toolbar, click  .
3. Click **OK**.

Print at runtime

Print graphic displays at runtime using the display print button or using a remote display print. You can also send diagnostics and alarm messages to a printer at runtime.

For information about	See
Display print buttons	Print displays at runtime on page 273
Print graphic displays	Print displays at runtime on page 273
Print displays remotely	Print displays on page 272
Print alarm messages	Methods for printing alarm information on page 102
Print diagnostics messages	Show and print diagnostics messages at runtime on page 135
Specify the printers to use at runtime for Windows computer applications	Specify the printers to use at runtime on page 191
The type of printer to use with a PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal	Install printers and software on a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal on page 199

Plan applications

This chapter describes how to:

- Understand the machines and processes you're automating
- Collect data
- Design an HMI tag database
- Plan graphic displays
- Plan languages
- Plan alarms
- Provide information for the operator
- Plan trends
- Plan recipes
- Design a secure system

Understand the process

Gather information from a variety of sources so you have a complete and detailed understanding of the machines and processes you are automating.

To gather information

- Talk to machine operators and other experts who are going to use the system. Find out what information they need to optimize machine operations and processes.
- Talk to management and management information systems staff to find out what information they need to support planning decisions.
- Break up each section of the process into its constituent parts.
- Determine what type of communications that you will use — which network types, data servers, and devices.
- Determine which process variables you need to have access to and identify their locations in the programmable controllers.

Collect data

When planning data collection, design your system so that only essential data is collected. Limiting data collection is important because collection activities require substantial processing power and generate much traffic on the network.

Keep data collection requirements in mind when designing the layout of the programmable controller data tables and the HMI tag database. Ideally, tag addresses reference contiguous blocks of programmable controller data tables to reduce highway traffic and optimize system response. If you're going to use data from an OPC server, read the server documentation to find out how to optimize traffic.

Design an HMI tag database

Take the time to plan your HMI tag database. A good design helps reduce the time required for maintenance and can improve programmable controller-to-FactoryTalk View response time.

Before you begin creating the tag database, collect:

- Flowcharts of your processes (or process and instrument diagrams).
- A list of programmable controller data table or register addresses your application will have access to.
- Machinery systems documentation.
- Alarm requirements (for more information, see Plan alarms on page 46).

Before creating tags:

- Develop naming conventions for tags, choosing names that are familiar and logical to everyone. This makes troubleshooting easier.
- Group related tags.
  Group tags in the way that makes the most sense for your application. For example, group all similar devices or group related areas of the plant floor.
  When creating tags, place related tags into folders. For greater organization, nest folders.

Tip: If you are going to use direct referencing for all your tags, you don't need to create HMI tags in FactoryTalk View.
Data Server tags usually update faster than HMI tags. To obtain the best system performance for your application, use Data Server tags where possible.

For information about creating HMI tags, see Use HMI tags on page 79.

For information about creating Data Servers to data server tags, see Work with tags on page 66.

Plan graphic displays

When planning displays, determine the best way for users to navigate through your displays and develop a template to establish a common look and feel for your displays.

Also consider factors about the runtime terminal that affect how the application is shown and used, such as:

- Screen size
- The maximum number of displays supported at runtime
- If the application runs on a personal computer, the maximum number of displays is based on the type of activation used.
- If the application runs on a MobileView, PanelView Plus 7, or PanelView Plus 6 terminal, the maximum number of displays is based on the amount of memory used by the application. Applications running on the terminal can contain as many displays as can fit in the memory of the terminal.
- PanelView Plus 7 Standard applications have a limit of 100 displays.
- PanelView Plus 6 Compact applications have a limit of 25 displays.

- Whether the operator will be using a keyboard, touch screen, mouse, or combination of navigation and data entry methods.

Tip:

Using FactoryTalk View Studio version 11.00.00 or later, you can create a v9.00 or v10.00 PanelView Plus 7 Standard runtime application (.mer) that contains more than 50 displays. The runtime application can be run on a PanelView Plus 7 Standard terminal installed with firmware version 9.00 or 10.00, and can be restored to a design time application (.med) using FactoryTalk View Studio version 9.00.00 or 10.00.00. However, from the restored application, you cannot create a PanelView Plus 7 Standard runtime application using FactoryTalk View Studio version 9.00.00 or 10.00.00, because the number of displays exceeds the restriction defined in FactoryTalk View Studio version 9.00.00 or 10.00.00. To maintain a PanelView Plus 7 Standard application that includes more than 50 displays, it is recommended to use FactoryTalk View ME v11.00.00 or later.

Using FactoryTalk View Studio version 9.00.00 or later, you can create a v8.20 or earlier version of PanelView Plus 7 Standard runtime application (.mer) that contains more than 25 displays or more than 200 alarms. The runtime application can be run on a PanelView Plus 7 Standard terminal installed with firmware version 8.20 or earlier, and can be restored to a design time application (.med) using FactoryTalk View Studio version 8.20.00 or earlier. However, from the restored application, you cannot create a PanelView Plus 7 Standard runtime application using FactoryTalk View Studio version 8.20.00 or earlier, because the number of alarms or displays exceeds the restriction defined in FactoryTalk View Studio version 8.20.00 or earlier.

For information about creating graphic displays, see Use graphic displays on page 252.

Develop a hierarchy of displays

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

Well-organized graphic displays present information clearly and consistently and guide users through the system. Before designing individual graphic displays, plan an overall display hierarchy and plan how users will navigate through the hierarchy.

For information about navigation methods and developing a hierarchy of displays, see Set up global connections on page 89.

Create a template to ensure consistency

It is possible to keep a consistent appearance among all the displays in an application by presenting the same pieces of information in the same place on each graphic display. To ensure uniformity, develop a display with common elements that acts as a template. Each time you develop a new display, start with a duplicate of the template.

For example, the template could contain:

- Your company logo
- A title

- The date and time
- Navigational buttons

Use global object displays to create template objects. You can copy the objects into your graphic displays, and update all the objects at once by updating the original object in the global object display. For information about creating global object displays, see Create global object displays and base objects on page 380.

Design displays

When designing displays, applying good visual design principles helps users and increases their efficiency.

Remember these important design principles:

Consistency

- Be consistent with your use of symbols and color.
- Be consistent with button labels and button placement.
  When you design several displays, place the same kinds of buttons at the same positions. For example, if you have 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 easily recognizable symbols. For example, use the conventional ISA symbols for tanks and valves.
- Don't overload the screen 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 North America, red and green usually mean stop and start. Keep color meanings consistent by assigning red only to Stop buttons, and green only to Start buttons. Some people are color blind to red and green so don't rely on color alone to establish meaning.
- Use high contrast color combinations, such as yellow on blue.

Display types

- On Top display type is used to position the display in front of any other display that is already open at runtime.
- Replace display type is used to have the display close any open displays on the screen at runtime. Replace displays do not close displays of the type On Top that have the **Cannot Be Replaced** option selected. Only one Replace display can be open at a time.

Usability

- If you're designing for a touch screen, don't place important buttons where they'll be blocked by an On Top display. The user can't press a covered button. Also, ensure that the button is large enough for users to touch easily.
- Ensure that there is always a clear way to move between displays.
- Keep the intended user in mind and design displays so they are easy to understand and use. Ask the users to test the displays.

Plan languages

Before setting up languages, plan:

- Which languages you need to use
- Which Windows fonts support these languages
  - For applications running on a personal computer or a MobileView terminal, which Windows fonts support these languages
  - For applications running on a PanelView Plus family terminal, which languages the fonts resident on the terminal support
- How different languages will affect design elements in your graphic displays, such as object size and message length
- Whether operators need to switch languages at runtime, and if so, under what conditions. This will help you determine where to locate **Language Switch** buttons in your application
- How to show operators which button to press to switch to their languages. For example, by using a text label in French or a French flag, to alert a French operator

For information about setting up language switching, see Set up language switching on page 159.

Plan alarms

Before setting up alarms, consider the following:

- What conditions will trigger alarms
- How operators will be notified of alarms
- What information you want alarm messages to contain
- How operators will respond to alarms
- The maximum number of alarm messages supported at runtime
- PanelView Plus 7 Standard applications have a limit of 500 alarms
- PanelView Plus 6 Compact applications have a limit of 200 alarms

Tip:

Using FactoryTalk View Studio version 9.00.00 or later, you can create a v8.20 or earlier version of PanelView Plus 7 Standard runtime application (.mer) that contains more than 25 displays or more than 200 alarms. The runtime application can be run on a PanelView Plus 7 Standard terminal installed with firmware version 8.20 or earlier, and can be restored to a design time application (.med) using FactoryTalk View Studio version 8.20.00 or earlier. However, from the restored application, you cannot create a PanelView Plus 7 Standard runtime application using FactoryTalk View Studio version 8.20.00 or earlier, because the number of alarms or displays exceeds the restriction defined in FactoryTalk View Studio version 8.20.00 or earlier.

For information about setting up alarms, see Set up alarms on page 95.

Provide information for the operator

In addition to notifying the operator of alarm conditions, you can provide information and instructions about plant processes, and inform the operator about system activities.

- Use local messages to give the operator information in a specific graphic display while the display is open. For information about setting up local messages, see Local messages on page 269.
- Use information messages to give the operator information no matter which display is open. For information about setting up information messages, see Information messages on page 392.
- Set up diagnostics messages to notify the operator of system activity such as tag reads and writes, running macros, communication problems, or problems opening displays. For information about setting up diagnostics messages, see Set up FactoryTalk Diagnostics on page 126.

Before setting up information notification, plan:

- What system activity the operator needs to be informed about
- What conditions will trigger information messages
- What information you want the messages to contain

Plan trends

When planning trends, consider how they will be used. For example, will the trend 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.
- Which tags need to be logged by using a data log model.

For information about setting up data logging, see Set up data logging on page 387.

For information about creating trend graphic objects, see Set up trends on page 398.

Plan recipes

When planning recipes, consider how they will be used. For example, will recipes be used to:

- Compare tag values to pre-set data values?
- Upload tag values to existing or new data sets?
- Download values from data sets to tags?
- Archive tag values to ensure compliance with government regulations?

Based on such considerations, you can determine:

- Which tag sets and data sets to pair together in a recipe file.
- Which **RecipePlus** button actions to use with the **RecipePlus** selector and **RecipePlus** table.

For information about using recipes and creating **RecipePlus** graphic objects, see Set up ReciplePlus on page 410.

Design a secure system

When deciding on your security requirements, consider whether to:

- Restrict access to every graphic display to prevent accidental changes.
- Restrict access to certain graphic displays.
- Have everyone log on.
- Use passwords, and if so, whether to have a minimum length, and whether to require that the passwords are changed periodically.
- Allow only authorized users to shut down the application.
- Use the DeskLock tool on Windows platforms to prevent users from switching to another application at runtime (for applications that will run on personal computers). You can use this tool to set up a customized desktop and to prevent access to Windows functions.

Based on these considerations, you can set up security for individual users or groups of users. For example, you might want to set up groups of users, such as a manager group and an operator group.

For more information about setting up security, see Set up security on page 142.

Work with applications

This chapter describes:

- Application files
- How to create, import, open, and close applications
- How to rename, copy, delete, back up, and restore applications
- How to specify project settings
- How to view application properties
- How to view HMI server properties

What is an application?

An application is the software application that you create in FactoryTalk View Studio to monitor and control your plant processes. An application consists of one or more data servers and an HMI project (also known as an HMI server). The data servers provide communications for the project. The project or HMI server consists of graphic displays, alarm information, user information, and other settings.

For information about viewing the application properties, see View application properties on page 61. For information about viewing the HMI server properties, see View HMI server properties on page 61.

Application versus project

This manual generally uses the term *application* to refer to both application-level and project-level procedures. There is an exception to using the **Project Settings** editor. See About project settings on page 57. In that section, the term *project* is used.

When you create an application, FactoryTalk View creates folders and files in various locations on the development computer. Some of the folders are empty until you start setting up your application. When you finish developing the application, FactoryTalk View uses the information in the various folders and files to create the runtime application.

Tip: By default, the HMI project file (.med) is in **\Users\Public\Public Documents\RSView Enterprise \ME\HMI projects**. You have the option to customize the location using the HMI Projects Folder Path Config Utility.

The runtime application consists of a file with the extension .mer. For information about creating the runtime application, see Create runtime applications on page 177.

Component files

Component files are located in folders in the HMI project folder, as described in the following table. For example, graphic display component files (with the extension .gfx) are stored in the Gfx folder. You might need to navigate to the location of these files. For example, if you are adding a component from one application into another application.

This folder	Contains	File extensions
Application name	HMI project file and the folders described later	.med
DLG	Data log models (component files)	.mdf

This folder	Contains	File extensions
Gfx	Graphic display component files	.gfx
Global Objects	Global object display component files	.ggfx
Images	Image files	.bmp, .png, or .jpg
Information	Information message component files	.ifm
Local	Local message component files	.loc
Macros	Macro component files	.mcr
PAR	Parameter component files	.par
RecipePlus	RecipePlus component files	.rpp

External folders

The following table lists some FactoryTalk View folders that are external to the HMI project folder, and the types of files they contain.

The folders are in **\Users\Public\PublicDocuments\RSView Enterprise**

This folder	Contains	File extensions
Images	Image files for use on graphic objects	.bmp
ME\Archives	ME archive application files	.apa
ME\Libraries	Graphic library component files and image files used in the library displays	.gfx .bmp
ME\Logs*Runtime Application Name*\Dlglog*Data Log Model Name*	Data log files for applications that have run on the development computer	.log .tag
ME\Logs*Runtime Application Name*\M_Alarms	Alarm log file for applications that have run on the development computer	.alm
ME\Runtime	Runtime application files	.mer
ME\HMI projects	Sample application folders, including all files and folders needed for the sample	

You can specify a different directory in which to store graphic library component files. For more information, see Location of library components on page 264.

Default log file locations

The following table shows the default log file locations.

PC and terminals	Default locations
PC and MobileView terminal	The alarm, audit, and data log folders are located in: **\Users\Public\Public Documents\RSView Enterprise\ME\Logs\<ApplicationName>**
PanelView Plus 7 and PanelView Plus 6 terminals	The alarm, audit, and data log folders are located in: **\Windows\Logs\<ApplicationName>**

PC and terminals	Default locations
PanelView Plus, PanelView Plus Compact, and PanelView Plus CE terminals	The alarm and data log folders are located in: **\Storage Card\Rockwell Software\RSViewME\Logs\<ApplicationName>**

You can also store data log files in a different location on the runtime computer, on a networked computer, or on a PC card. For more information, see Set up data logging on page 387.

Name files

File names, including the path, can be up to 200 characters long (file names can contain spaces). For example, the following path and file name contains 119 characters:

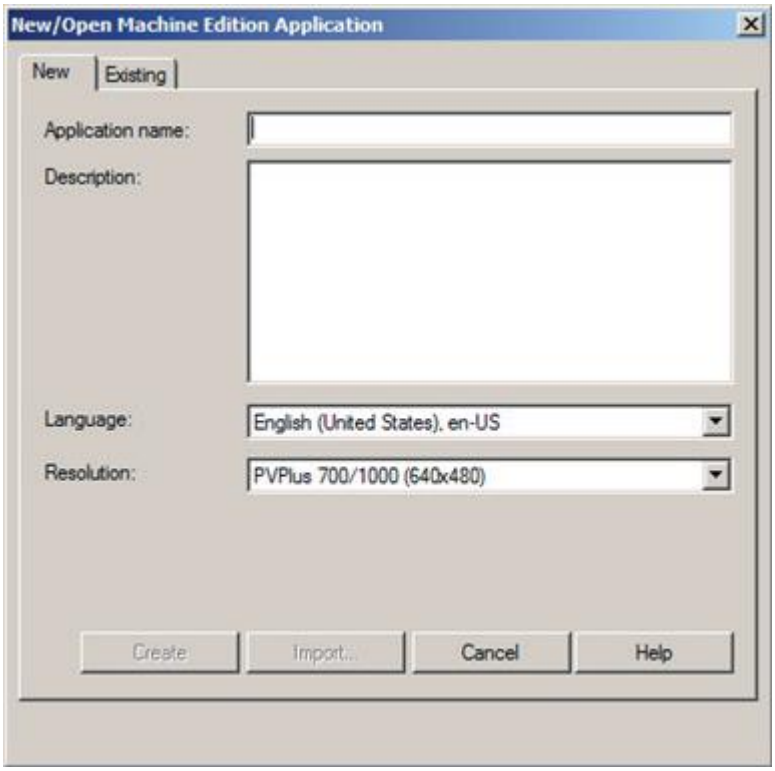
C:\Documents and Settings\All Users\Documents\RSView Enterprise\ME\HMI projects\Malthouse Clgx 640x480\Gfx\Steeping.gfx

Create an application

You can create a FactoryTalk View ME application when you start FactoryTalk View Studio or when FactoryTalk View Studio is open.

To create an application

1. If FactoryTalk View Studio is not started, start FactoryTalk View Studio.
2. If FactoryTalk View Studio is started, from the **File** menu, select **New Application** or click the **New Application** tool. If an application is already open, FactoryTalk View Studio asks you whether to close the application. Click **Yes**.



3. On the **New** tab of the **New/Open Machine Edition Application** dialog box, enter a name for the application in the **Application Name** box.
4. (Optional) Enter a description for the application. You can also modify the description in the application **Properties** dialog box after the application is created. See View application properties on page 61.
5. Select a language for the application. You can add more languages in the **Language Configuration** dialog box after the application is created. For information about using different languages, see Set up language switching on page 159.
6. Select a resolution for the application. You can modify the resolution in the **Project Settings** editor after the application is created. See Project window size and runtime screen resolution on page 57.
7. Click **Create**.

FactoryTalk View Studio creates the application's folders and files and then shows the new application in the Explorer window. The system automatically creates a display called **MAIN** and sets the initial graphic to **Main**.

By default, the application files are stored in:

\Users\Public\Public Documents\RSView Enterprise\ME\HMI projects\<ApplicationName>

The application file is given the extension .med.

You have the option to customize the location using the HMI Projects Folder Path Config Utility.

Import an application

You can import the following application files to FactoryTalk View Studio and convert them to the current version FactoryTalk View ME application files using Machine Edition Import Wizard:

- PanelBuilder® application (.pba or .pva)
- PanelBuilder32 application (.pba or .pva)
- PanelBuilder 1400e application (.pvc)
- RSView® Machine Edition application (.med)
- FactoryTalk View ME application (.med)

When importing an application:

- If the application includes alarm block tags, the conversion process creates direct references for each of the block tags imported. The direct reference is based on the original block tag name and address, with the modifier "L" appended. The "L" modifier is followed by the number of elements to be read. The number is arrived at by dividing the number of bits that is defined for the block tag by the number of bits in each address element (assumed to be a 16-bit word). For example, if the original block tag had 47 bits defined, using the address N7:0, it would be converted to a direct reference as N7:0,L3.
- If the application files are created with an earlier version FactoryTalk View, FactoryTalk View will create a copy of the application for the current version and will not alter the earlier version.
- PanelView Plus 7 Standard application will be converted to PanelView Plus 7 Performance or custom size application, and the application limits will be removed. If you want a PanelView Plus 7 Standard application, specify the project window size in the **Project Settings** dialog box after the application is imported.
- Compact Machine Edition application will be converted to a regular non-compact application, and the application limits will be removed. If you want a PanelView Plus 6 Compact application, specify the project window size in the **Project Settings** dialog box after the application is imported.

- The Import Wizard converts PanelBuilder objects to the most similar Machine Edition objects. When conversion is not possible, an error message is shown.
- A PanelBuilder application will be converted to a Machine Edition application that does not contain a border or a title bar.

You can use the current version application files to create previous version runtime files in FactoryTalk View Studio. For more information, see Open and edit applications from earlier versions of RSView or FactoryTalk View ME on page 56.

You cannot import applications that were created with later versions of FactoryTalk View Studio than the one you are presently running. For example, if you are running Version 7.0, you cannot import an application that is created with Version 8.0.

To import an application

1. If FactoryTalk View Studio is not started, start FactoryTalk View Studio.
2. If FactoryTalk View Studio is started, from the **File** menu, select **New Application** or click the **New Application** tool. If an application is already open, FactoryTalk View Studio asks you whether to close the application. Click **Yes**.
3. On the **New** tab of the **New/Open Machine Edition Application** dialog box, enter a name for the application in the **Application Name** box.
4. (Optional) Enter a description for the application. You can also modify the description in the application **Properties** dialog box after the application is imported. See View application properties on page 61.

   Tip: If the imported application contains an application file comment, the comment will overwrite your description when the application is imported.

5. Select a language for the application. You can add more languages in the **Language Configuration** dialog box after the application is imported. For information about using multiple languages, see Set up language switching on page 159.
6. Click **Import** to start the Machine Edition Import Wizard.
7. In the **Machine Edition Import Wizard - Import Type** dialog box, select the type of file that you want to import and click **Next**.
8. In the **Machine Edition Import Wizard - Import File** dialog box, enter the name of the file you want to import or click the browse button to select the file.
9. If you are importing a FactoryTalk View ME application (.med), click **Finish**.
10. If you are not importing a FactoryTalk View ME application (.med), click **Next**. In the **Machine Edition Import Wizard - Import Options** dialog box, specify graphic display conversion and touch property and click **Finish**.

The Machine Edition Import Wizard converts the imported application to a Machine Edition application and shows it in the Explorer window. The system automatically creates a display called **MAIN** and sets the initial graphic to **Main**.

By default, the converted files are stored in:

\Users\Public\Public Documents\RSView Enterprise\ME\HMI projects\<ApplicationName>

The application file is given the extension .med.

For more information about importing PanelBuilder 1400e applications, see Convert PanelBuilder 1400e applications on page 438.

For more information about importing PanelBuilder and PanelBuilder32 applications, see Convert PanelBuilder and PanelBuilder32 applications on page 451.

Open an application

Open an application when you start FactoryTalk View Studio or when FactoryTalk View Studio is open.

FactoryTalk View ME application files are converted to the current version of FactoryTalk View when you open them. Once the application has been converted, it cannot be opened by earlier versions of FactoryTalk View Studio. However, you can still use the current version of FactoryTalk View Studio to create runtime application files that will run on earlier versions of ME Station.

If an application was created in earlier versions of FactoryTalk View ME, when you open the application in the current version of FactoryTalk View Studio, ensure that the application's IP addresses do not contain a leading zero. FactoryTalk Linx interprets a leading zero as an octal notation. For example, the IP address 131.200.13.111 is valid, but 131.200.013.111 is invalid. Before opening the application, edit the application's Nodes.csv file in Notepad by removing leading zeroes in IP addresses.

To open an application in FactoryTalk View Studio, the logged-in user must have read/write access to the sub-folder that holds the HMI applications. If the security designation for FactoryTalk View Studio users is Power Users, they may not have write access.

To open an application

1. If FactoryTalk View Studio is not started, start FactoryTalk View Studio.
2. If FactoryTalk View Studio is started, from the **File** menu, select **Open Application** or click the **Open Application** tool . If an application is already open, FactoryTalk View Studio asks you whether to close the application that is currently open. Click **Yes**.
3. On the **Existing** tab, select the application to open.
   The resolution and version that was last used to open the application is shown.
4. If the application has more than one language, select a language to open the application in.
5. Click **Open**.

Tip: To open two applications at the same time, for example, to copy images and graphic objects between applications, start two instances of FactoryTalk View Studio.

For information about the features supported in different versions of FactoryTalk View, see Features supported in different versions of FactoryTalk View on page 424.

For information about opening a sample application, see Open sample applications on page 35.

For information about using different languages, see Set up language switching on page 159.

For information about the default language, see The default language on page 159.

Open a recently used application

You can open a recently used application from the **File** menu.

To open a recently used application

1. From the **File** menu, select **Recent Applications**.
2. Click the application name in the list.
3. If an application is already open, FactoryTalk View Studio asks you whether to close the application. Click **Yes**.

The application opens in the language that it was last opened.

HMI projects folder settings

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

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

- NT AUTHORITY\IUSR: Full control

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

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

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

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

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

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

For more information about security, see Set up security on page 142.

For more information about customizing the location of the HMI projects folder, see Help.

Open and edit applications from earlier versions of RSView or FactoryTalk View ME

You can open and edit applications from RSView 3.10, 3.20, and FactoryTalk View 4.0 and later. When you are finished editing, you can create runtime application files for ME Station version 6.0 and later. This allows you to continue running your applications on existing terminals, without upgrading all your software.

The application is converted to the current version of FactoryTalk View when you import it, but when you create the runtime application file you can save it back to the earlier version.

If the application uses a feature that the runtime version does not support, you will not be able to create the runtime application file. When you create the runtime application file, it uses the application's current language. You can create as many different runtime application files, in as many different languages, as you want.

For information about importing applications, see Import applications on page 52.

For information about the features supported in different versions of FactoryTalk View and RSView, see Features supported in different versions of FactoryTalk View on page 424.

For information about using multiple languages, see Set up language switching on page 159.

For information about using global objects, see Use graphic displays on page 252.

Close an application

To close an application

- From the **File** menu, select **Close Application**.
  If there are any unsaved changes in open editors, FactoryTalk View Studio asks you whether to save the changes before closing the application.

To close an application and exit FactoryTalk View Studio at the same time

- Click the **Close** button at the right end of the FactoryTalk View Studio title bar  .
  If there are any unsaved changes in open editors, FactoryTalk View Studio asks you whether to save the changes before closing.

Rename, copy, delete, back up, and restore applications

Use FactoryTalk View ME Application Manager to:

- Rename applications
- Copy applications

- Delete applications
- Create a compressed backup copy of an application, with the file extension .apa
- Restore an application from the backup file, with the file extension .apa
- Restore a legacy project backup file (with the file extension .mea) from an earlier release of FactoryTalk View Studio, and convert it to the current version
- Restore a runtime application file, with the file extension .mer, to a development application, with the file extension .med. For more information, see Convert runtime application files to development applications on page 181.

Start FactoryTalk View ME Application Manager

Use one of the following methods:

- From Windows **Start** menu, select **All apps > Rockwell Software > FactoryTalk View ME Application Manager**.
- In FactoryTalk View Studio, from the **Tools** menu, select **Application Manager**.

About project settings

Project settings determine the general appearance of your runtime application, such as the size and position of the project window and whether a title bar appears in the project window.

Tip: Project settings affect how your graphic displays look. Specify project settings before you create graphic displays.

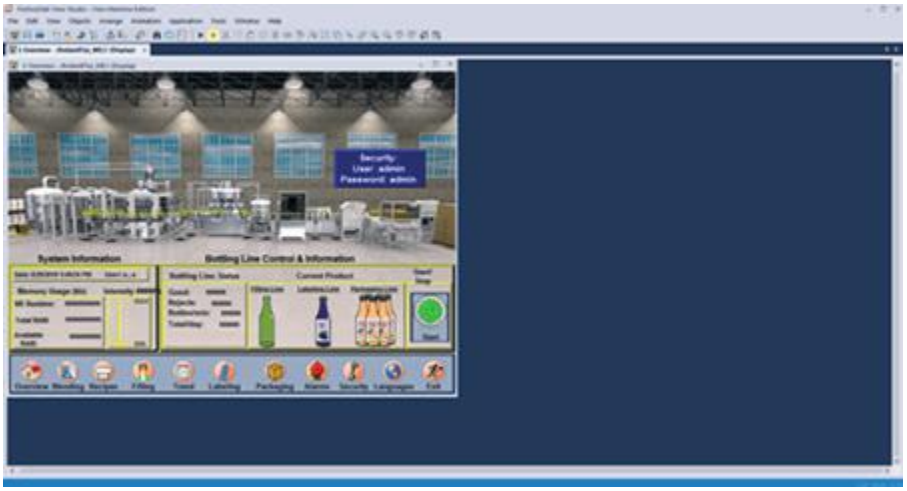
Use the **Project Settings** editor to specify project settings. For details about selecting options in the editor, see Help.

Project window size and runtime screen resolution

This section describes the difference between project window size and runtime screen resolution, and describes how these two settings can affect the behavior of your application at runtime.

The project window size is the amount of room your application occupies on the runtime terminal or computer screen, in pixels. The runtime screen resolution is the actual size of the runtime terminal or computer screen, in pixels. Therefore, the project window size must be less than or equal to the runtime screen resolution.

In the following example, the screen resolution is 1024 x 728 and the project window size is 800 x 600.



Usually, an application is designed to completely fill the runtime terminal or computer screen. If you want your application to do this, set the project window size to the resolution of the screen in pixels.

Screen resolution for MobileView, PanelView Plus 7, and PanelView Plus 6 terminals

If you are going to run your application on a MobileView, PanelView Plus 7, or PanelView Plus 6 terminal, select one of these project window sizes:

For this terminal or monitor	Select this project window size
PanelView Plus 6 400 or 600 PanelView Plus 6 Compact 400 or 600	320x240
PanelView Plus 7 Standard 4" Wide	480x272
PanelView Plus 700 or 1000 PanelView Plus 6 700 or 1000 PanelView Plus 6 Compact 1000 PanelView Plus 7 Standard 6" or 7" PanelView Plus 7 Performance 6" or 7"	640x480
PanelView Plus 7 Standard 9" Wide PanelView Plus 7 Performance 9" Wide	800x480
PanelView Plus 6 1250 PanelView Plus 7 Standard 10" PanelView Plus 7 Performance 10"	800x600
PanelView Plus 6 1500 PanelView Plus 7 Standard 15" PanelView Plus 7 Performance 15"	1024x768
MobileView PanelView Plus 7 Standard 12" Wide PanelView Plus 7 Performance 12" Wide	1280x800
PanelView Plus 7 Performance 19"	1280x1024

Based on the project window size selected, you may need to:

- Resize the default message displays so that the entire message is visible.
- Reposition the [ALARM] display from the default position (0, 0), as this position is below the viewing area of the terminal.

Change the project window size after creating graphic displays

The project window size is used for all graphic displays of the Replace type. If you change the window size after you have already created graphic displays, you have the option of scaling graphic displays.

If you choose to scale graphic displays, all Replace and On Top displays are resized and the objects in them are scaled to fit the new size. You can also specify whether to scale the font size and border size of the graphic objects, and the size of graphic images in the Images folder.

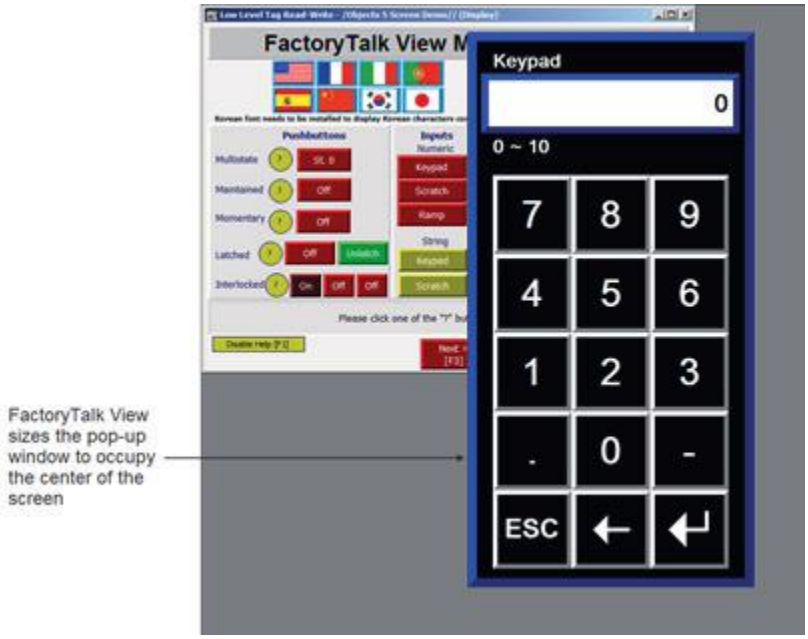
If you choose not to scale displays when you change the project window size, Replace displays are resized, but any objects on the displays remain the same size and in the same position as before. On Top displays are not resized.

For more information about Replace and On Top displays, see Replace displays on page 259 and On Top displays on page 259.

For more information about scaling displays, see Help.

How the runtime screen resolution affects the pop-up windows

FactoryTalk View comes with a pop-up **Login** window, as well as numeric and string pop-up windows for runtime data entry, and a **Change Password** window for changing user passwords at runtime.



How the project window size affects the default message displays

Your application comes with pre-configured, default graphic displays for diagnostics, alarm, and information messages. If the project window size is smaller than the default displays, the edges of the displays are cut off at runtime.

The following table shows the sizes of the default displays.

Graphic Display	Width in pixels	Height in pixels
[DIAGNOSTICS]	640	160
[ALARM]	640	210
[INFORMATION]	640	80

If your project window size is smaller than the default message display sizes, resize the default displays or use your own displays instead.

PanelView Plus 7 Standard applications and PanelView Plus 6 Compact applications

PanelView Plus 7 Standard and PanelView Plus 6 Compact applications have the following restrictions:

Restrictions	PanelView Plus 7 Standard applications	PanelView Plus 6 Compact applications
Maximum number of displays	100* (1)	25
Maximum number of alarm messages	500	200
Maximum screen resolution	1280x800	640x480
Maximum number of FactoryTalk Linx shortcut	1	1
Methods of communications	Ethernet If the application uses FactoryTalk Linx, only Ethernet drivers are supported.	Serial and Ethernet If the application uses FactoryTalk Linx, only Serial-DF1, Serial-DH485, and Ethernet drivers are supported.
Supported versions of FactoryTalk Linx* (2)	5.20 and later	5.20 and later

1. When using different versions of FactoryTalk View, you can create a PanelView Plus 7 Standard application containing a different number of displays.

When using FactoryTalk View version	You can create a PanelView Plus 7 Standard application containing up to
7.00.00, 8.10.00, and 8.20.00	25 displays
9.00.00 and 10.00.00	50 displays (25 On Top displays and 25 Replace displays)
11.00.00 and later	100 displays

2. If no FactoryTalk Linx shortcut is defined, a single KEPServer Enterprise data server can be used to communicate with a single device.

The Kepware Hilscher Universal and Modbus Plus communications drivers are not supported for PanelView Plus 7 Standard, PanelView Plus 6 Compact, or PanelView Plus Compact terminals. However, the drivers are supported on PanelView Plus 6 Compact 1000 terminal.

To change an application to a PanelView Plus 7 Standard or PanelView Plus 6 Compact application, on the **General** tab of the **Project Settings** dialog box, from the **Project window size** list, select a **PVPlus Compact** or a **PVPlus 7 Standard** option.

Title bar

In the **Project Settings** dialog box, if you select **Title bar** on the **Runtime** tab, a title bar is used for all Replace graphic displays at runtime. For information about graphic display types, see Replace displays on page 259 and On Top displays on page 259.

If you plan to set up security for your application and want to prevent unauthorized users from stopping the application, do not use a title bar. For information about setting up security, see Set up security on page 142.

If you change the Title bar or Border setting, you have the option of scaling graphic displays to accommodate the new display size. For details, see Help.

For On Top displays, you can use the **Display Settings** dialog box to specify a different name to use on the title bar. For details, see Help.

Enable auto logout

Use the **Project Settings Runtime** tab to log the current user out automatically if the terminal is inactive for a specified period. You can set up the terminal to open a particular graphic display after the automatic logout, and you can apply a parameter file or parameter list to the display you select.

If no display is selected, whatever display was open at logout will remain open.

Note that the auto-logout feature that returns to a specific display applies to ALL user accounts, including the DEFAULT user account.

View application properties

You can view the information about your application:

- Application name
- Application description, if any

To view application properties

- From the **Application** menu, select **Application Properties**.

View HMI server properties

You can use the **Explorer** window to view the information about your project or HMI server:

- Application name
- Application description, if any
- Location of the project file
- Number of graphic displays in the application
- Maximum number of displays you're licensed to use

To view HMI server properties

- In the **Explorer** window, right-click the HMI server name and select **Properties**.

Set up communications

This chapter describes:

- Data servers
- Open Platform Communications (OPC)
- How to create data servers and set up communications

What is a data server?

A data server provides access to devices on the network, making it possible to browse, read, and write values from FactoryTalk View applications.

FactoryTalk View ME supports the following types of data servers:

- **Rockwell Automation Device Servers**
  FactoryTalk Linx is the recommended data server for all Rockwell Automation devices and networks. FactoryTalk Linx does not require activation or licensing and can be installed as often as needed in any application.
- **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. KEPServer Enterprise is an OPC Data Access v2.05a server available with FactoryTalk View ME.
  KEPServer Enterprise is used to communicate with non-Rockwell Automation devices like Siemens or Modicon controllers.
  KEPServer Enterprise does not require activation or licensing for FactoryTalk View ME.

About OPC communications

FactoryTalk View ME supports OPC communication on the PanelView Plus family of terminals for use with only KEPServer Enterprise. FactoryTalk View ME Station running on a personal computer or a MobileView terminal supports any OPC Data Access v2.05a compliant server. KEPServer Enterprise is an OPC DA V2.05a compliant OPC server.

Create data servers

A FactoryTalk Linx data server is created automatically with a new application. It is located under the application's HMI server in the **Explorer** window, just above the FactoryTalk **System** folder.

You can create additional OPC data servers for your application. For example, you might want to use a FactoryTalk Linx data server and a KEPServer Enterprise data server to communicate with a third-party device.

Set up FactoryTalk Linx data servers

You can use the **Communication Setup** editor to add drivers, add devices, set up driver and device properties, and set up device shortcuts.

To set up communications in FactoryTalk Linx

1. In the **Explorer** window, open the FactoryTalk Linx data server.
2. Double-click the **Communication Setup** editor.

3. In a new application, the **FactoryTalk Linx Configuration Wizard** opens. Follow the instructions to create a new configuration or use an existing device configuration. Once you make a choice and click **Finish**, the **Communication Setup** editor opens.
   In an existing application, double-clicking **Communication Setup** opens the **Communication Setup** editor, with its two tabs.
   - The **Design (Local)** tab is for establishing the location of the tags/addresses for editing. The tags/addresses can be online with a controller or other data server, or an offline controller file. This will enable the **Tag Browser** to find the tags/addresses.
   - The **Runtime (Target)** tab is for identifying the connection from the runtime computer or terminal to the controller or other data server. If the paths are the same, use the **Copy** button to copy the Design configuration to the **Runtime** tab.

For more information, see Help for FactoryTalk Linx.

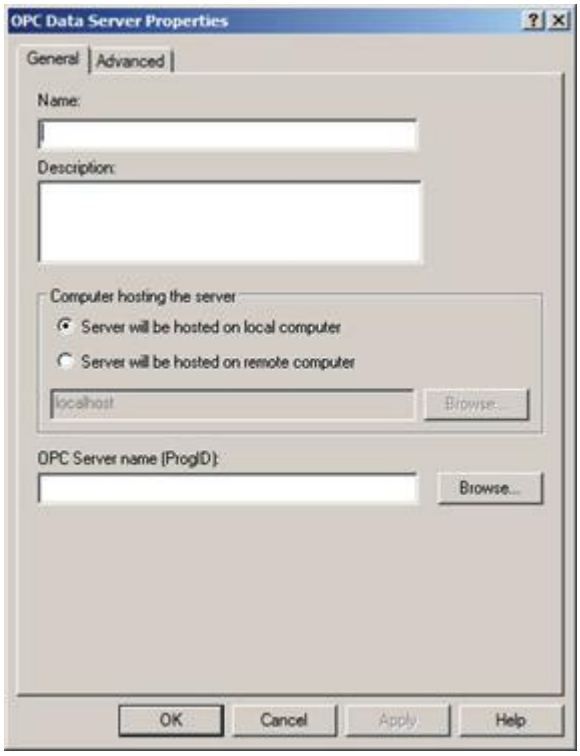
Set up an OPC data server

If you need to connect third-party devices, use a KEPServer Enterprise data server.

If you need to view or use tag names when you are not connected to a device, you can set up a cache that will read in tag names from the device and enable them to be used when the device is offline.

To create an OPC data server

- In the **Explorer** window, right-click the application, and select **Add New Server > OPC Data Server**.



For details about selecting options in the **OPC Data Server Properties** dialog box, see FactoryTalk Help.

Update data server caches

If tags are added, modified, or deleted on the data server, you must update (synchronize) the cache manually.

For information about updating the data server cache periodically after you create it, and editing and removing data servers, see FactoryTalk Help.

Set up communications

Use the following steps to set up communications.

To set up communications

1. Gather information about your network and the devices that are connected to it. You will need this information to set up FactoryTalk Linx or an OPC server. Do one of the following:
   - If you want to use KEPServer Enterprise as the OPC server that enables communications on your network, set up KEPServer Enterprise on the development computer. For details, see KEPServer Enterprise Help.
   - If you want to use an OPC server on the development computer. For details, see the documentation supplied with your OPC server.
2. In FactoryTalk View Studio, create a data server that points to the OPC server you have installed.
   If you are using FactoryTalk Linx for communications, you don't need to create a data server because it is created automatically when you create your FactoryTalk View ME application.

   Tip: KEPServer Enterprise is the only OPC Server supported on PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, and PanelView Plus CE terminals.

3. If you want to use HMI tags in your application, open the **Tags** editor, and then create device tags that point to addresses in devices. For details about creating HMI tags, see Use HMI tags on page 79.
   If you do not want to use HMI tags, but instead want to reference addresses in devices directly, skip this step. The best practice is to use direct reference tags.
4. In objects in graphic displays, or anywhere else you want access to values in tags, type the tag name using the syntax for your OPC server, or use the **Tag Browser** to select tags. For details about using tags, see Work with tags on page 66.
5. Create the runtime application file. For details, see Create runtime applications on page 177.
   All the necessary tag information is compiled with the runtime application file.
6. If you are using an OPC server other than FactoryTalk Linx, install the OPC server software on the runtime computer.
   For applications that will run on a personal computer, the OPC server can be on a different (remote) computer than the runtime computer, but the remote computer must be on the same network as the runtime computer.

Work with tags

This chapter describes:

- Types of tags
- When to use data server tags
- Steps for using data server tags
- When to use HMI tags
- Steps for using HMI tags
- How to browse for tags and offline tags
- How to use the **Tag Browser**
- How to use tags and expressions in your application
- How to log tag values
- How to use macros to assign values to tags

For information about creating HMI tags, see Use HMI tags on page 79.

What is a tag?

A tag is a logical name for a variable in a device or in local memory (RAM). For example, a tag can represent a process variable in a programmable controller.

Data server tags

FactoryTalk View conforms to the OPC DA (Data Access) 2.05a specification for information exchange among automation or control applications, field systems or devices, and business or office applications.

In FactoryTalk View, data servers such as FactoryTalk Linx and KEPServer Enterprise provide access to OPC-DA-compliant tags. In the FactoryTalk View documentation, the tags you use through a data server are called data server tags. For more information about data servers, see Set up communications on page 65.

Data server tags include tags found in Studio 5000 processors and tags from other OPC-compliant devices. You use data server tags by providing a direct reference to the tag's location, wherever you want your application to use the data.

Starting from version 21, Logix controllers support the extended tag properties that are configurable in Studio 5000 Logix Designer®.

Starting from FactoryTalk View version 9.00.00, you can use the following extended tag properties when a controller tag type can be referenced:

- **.@Description**: Description of the tag
- **.@EngineeringUnit**: A system of measurements of physical quantities
- **.@Min**: The minimum numeric value for the non-Boolean tag being measured
- **.@Max**: The maximum numeric value for the non-Boolean tag being measured
- **.@State0**: The false Boolean state of the tag
- **.@State1**: The true Boolean state of the tag

Starting from Logix Designer version 33 and FactoryTalk Linx version 6.20.00, you can also use the following extended tag properties:

- **.@Library**: Associates instructions with a display library for faceplate call-up.
- **.@Instruction**: Associates the instruction name for faceplate call-up.
- **.@Label**: A label that appears on the graphic HMI symbol.
- **.@Area**: The area name for area-based security in the HMI.
- **.@URL**: The Uniform Resource Locator (URL) for accessing the help content from HMI.
- **.@Navigation**: A path to another source of information. For example, a document link or website.
- **.@Name:** A text string ranging from 0 through 512 characters that defines the tag name in the client application.
- **.@Quality**: A string following the OPC DA standard to indicate the tag status. The following table lists the quality codes and their meanings:

Value	Description
0	Bad
4	Bad - Configuration Error in Server
8	Bad - Not Connected
12	Bad - Device Failure
16	Bad - Sensor Failure
20	Bad - Last Known Value Passed
24	Bad - Common Failure
28	Bad - Out of Service
64	Uncertain
68	Uncertain - Last Usable Value
80	Uncertain - Sensor not Accurate
84	Uncertain - Engineering Units exceeded
88	Uncertain - Value from Multiple Sources
192	Good
216	Good - Local Override, Value Forced

  To obtain the quality value, for example, `[New_shortcut]Tag1.@Quality`, the base tag `[New_shortcut]Tag1` must be added to the client application as well.
- **.@Timestamp**: A string that follows the format of Year/Month/Day/Hour/Minute/Second to define the time when FactoryTalk Linx data server polls the data to the client.
- **.@Scope:** A string that defines a tag's controller and program scope.

To read the values, use the format **TagName.@ExtendedProperty** where *TagName* is the controller tag and *ExtendedProperty* is the extended tag property, for example, *[Shortcut]SodaCIPTanks.Water.Value.@Min*.

HMI tags

In addition to direct referencing tags from data servers, FactoryTalk View allows you to create tags with additional properties such as minimum and maximum values, scale, and offset. These tags can reference values at an external

data source, or store values in the runtime computer's memory. Tags you create in FactoryTalk View are called HMI tags.

Data source

The FactoryTalk View documentation uses the term data source as a generic term that includes all possible sources of tag data, for both data server tags and HMI tags.

The data source can be memory or a device such as a programmable logic controller or OPC server. The application writes values to and reads values from the data source.

The data source is configured to exchange information (in the form of numeric or string values) between the application and the physical machine that your project is controlling.

Use a tag

Use the following steps to use a tag.

To use a tag

1. If the tag does not exist, create the tag.
   - To use a data server tag, you can use an existing tag in the processor (for example, a programmable controller), or you can create one in the processor or OPC server. For example, in a Studio 5000 processor, you could create the tag using your Studio 5000 Logix Designer® programming software.
   - To use an HMI tag, you must first create it in the **Tags** editor. For details, see Use HMI tags on page 79.
2. Browse for or type the name of the tag anywhere you want to connect an object to data at runtime.
   For example, to make a push button change a tag value when it is pressed, connect the push button to a tag by typing the tag name on the **Connections** tab of the push button's **Properties** dialog box.

If you don't know the names of the tags, you can browse for them. You can browse while online and connected to a device, or you can browse for tags from an offline file, for example, a Studio 5000 program file. To browse for tags, use the **Tag Browser**.

Use tag names that don't exist

You can type the name of a tag that does not exist. If you do this, ensure that you spell every instance of the tag name consistently, and that when you create the tag, you spell it the same way you did when you referred to it.

Performance considerations when using extended tag properties

ControlLogix extended tag properties provide application efficiencies and high integration between the control and HMI application programs. Before undertaking the design of an HMI application accessing extended tag properties, an HMI designer must consider the performance characteristics related to extended tag properties.

Extended tag properties are integrated into the ControlLogix application program. On a controller, the metadata is stored in FLASH memory. To maintain controller performance, the system design requires that each extended tag property that is used on an HMI display be accessed separately. This design affects the display loading time within the HMI application. When loading a display that uses extended tag properties, all objects (for example, numeric

display, button captions, and the like) will be wire-framed before all data and extended tag property data is read from the controller.

The time to load the extended tag property data depends on how many extended tag properties are used on a display. To improve overall HMI application performance, extended tag property data is cached in memory. This caching improves the display loading performance when a display is revisited.

Extended tag properties can be used with the runtime language switching feature of FactoryTalk View. When a language switch occurs, the HMI application will read, if the locale exists, the referenced string-type extended tag property values, for example, *.@Description*. Reading the extended tag property values delays the display refresh that occurs when a language switch occurs. The refresh delay is the same as the delay when a display is initially loaded.

The following tables show the performance data that was generated using an Allen-Bradley 1756-L75 controller running V28 firmware and an Allen-Bradley 1756-L85 controller running V28 firmware. You can use the data as a reference to estimate the impact on your applications that use extended tag properties.

Performance of a 1756-L85 controller running V28 firmware

The following table shows the time to load the extended tag property data when switching displays.

Terminal types	When a graphic display includes			
	50 extended tag properties	200 extended tag properties	500 extended tag properties	1000 extended tag properties
Computer	1.5 s	2.4 s	4.0 s	6.9 s
PanelView Plus 6 400/600	2.7 s	4.2 s	7.6 s	9.6 s
PanelView Plus 6 700-1500	1.7 s	2.5 s	4.3 s	5.4 s
PanelView Plus 7 Standard	3.6 s	6.1 s	11.0 s	14.3 s
PanelView Plus 7 Performance	1.4 s	2.3 s	4.1 s	5.1 s

The following table shows the time to load the extended tag property data when switching languages at runtime.

Terminal types	When a graphic display includes			
	50 extended tag properties	200 extended tag properties	500 extended tag properties	1000 extended tag properties
Computer	0.6 s	1.1 s	2.2 s	4.1 s
PanelView Plus 6 400/600	1.1 s	2.8 s	5.5 s	6.4 s
PanelView Plus 6 700-1500	1.2 s	2.0 s	3.4 s	4.5 s
PanelView Plus 7 Standard	1.6 s	3.6 s	7.8 s	10.5 s

Terminal types	When a graphic display includes			
	50 extended tag properties	200 extended tag properties	500 extended tag properties	1000 extended tag properties
PanelView Plus 7 Performance	1.2 s	2.2 s	3.7 s	5.1 s

Performance of a 1756-L75 controller running V28 firmware

The following table shows the time to load the extended tag property data when switching displays.

Terminal types	When a graphic display includes			
	50 extended tag properties	200 extended tag properties	500 extended tag properties	1000 extended tag properties
Computer	2.3 s	4.5 s	9.6 s	17.4 s
PanelView Plus 6 400/600	3.2 s	6.3 s	12.4 s	14.1 s
PanelView Plus 6 700-1500	3.4 s	4.9 s	10.2 s	11.9 s
PanelView Plus 7 Standard	4.8 s	8.5 s	16.8 s	20.2 s
PanelView Plus 7 Performance	3.0 s	4.6 s	9.8 s	11.4 s

The following table shows the time to load the extended tag property data when switching languages at runtime.

Terminal types	When a graphic display includes			
	50 extended tag properties	200 extended tag properties	500 extended tag properties	1000 extended tag properties
Computer	1.2 s	3.5 s	8.2 s	16.0 s
PanelView Plus 6 400/600	2.0 s	5.1 s	10.1 s	11.2 s
PanelView Plus 6 700-1500	1.5 s	3.8 s	8.9 s	10.0 s
PanelView Plus 7 Standard	2.7 s	6.1 s	13.2 s	16.8 s
PanelView Plus 7 Performance	1.6 s	3.9 s	8.8 s	10.8 s

Memory considerations when using extended tag properties

The HMI system caches extended tag property values to improve performance. The HMI system caches extended tag property values for each runtime language viewed while the HMI application is executing. The cache is stored in RAM. On PanelView Plus 6 and PanelView Plus 7 terminals, there is a limited amount of RAM. To prevent the PanelView Plus 6 or PanelView Plus 7 terminal from encountering an out-of-memory condition, the system is designed to empty the

extended tag property cache after eight hours. The system will remove all cached extended tag property values from the RAM cache except for the value associated with the HMI default language and the last runtime language viewed.

Example: Extended tag property

Assume that an HMI application supports English (default language), French, and Chinese languages. When the application is launched, the English language extended tag property values will be read and cached as the operator navigates to different displays. After four hours of operation, the operator switches to French. After two additional hours of operation in French, the operator switches to Chinese. If nothing else changes, after eight hours passes, the French language extended tag property values will be removed from the cache. The English language and Chinese language extended tag property values will be retained in the cache.

Use a data server tag

For some of the things you want to do in your application, you might not need to create HMI tags. Use direct references to tags located in devices instead, for example, tags located in an OPC server's database.

Using data server tags allows you to add, modify, or delete tags in a device without having to duplicate the changes in the FactoryTalk View HMI tag database.

Some devices, for example, Studio 5000 processors, support complex data types, such as arrays and structures. Your controller might have structures that contain hundreds of member elements.

Referencing tag values directly eliminates the need for creating an HMI tag for each member.

Direct reference tags usually update faster than HMI tags. To obtain the best system performance for your application, use direct reference tags where possible.

To use a data server tag

1. Create the tag in the OPC server or processor, or use an existing tag in the processor.
   The server must understand the syntax of the device it is connected to. You need only create a tag if you want to browse the server.
   Examples of OPC servers include KEPServer Enterprise. For information about creating tags in OPC servers that have their own tag database, see the documentation provided with the server.
   For information about creating tags in a Studio 5000 processor, see the documentation for your device programming software.
2. Create a data server in FactoryTalk View, if your application does not already have one. Every new FactoryTalk View Machine application includes a FactoryTalk Linx data server automatically.
   For more information, see Set up communications on page 63.
3. In graphic displays, data log models, or anywhere else you want to use a tag's value, create a direct reference to the tag. At runtime, the tag's value is passed to the graphic display or data log model.

Use an HMI tag

Use HMI tags to manipulate tag data and store tag values in the runtime computer's memory.

To use an HMI tag

1. In the **Tags** editor in FactoryTalk View, create a tag. HMI tags can be used as an internal storage location. Optionally you can map the tag name to an OPC tag or device address.
   For more information, see Use HMI tags on page 79.
2. In graphic displays, alarm triggers, or anywhere else you want to use a tag's value, assign the tag. At runtime, the tag's value is passed to the graphic display or alarm system.

Scale, offset, or provide a range for data

Use HMI tags if the data server you are using does not provide for:

- **Scale or offset a value.**
  The scale and offset modify the *raw data* that comes from and goes to the programmable controller before the data is saved in the computer's memory (called the *value table*). The scale and offset also modify the value specified in FactoryTalk View before it is written to the programmable controller.
  The scale is a multiplication factor — the value from the programmable controller is multiplied by the scale.
  The offset is a fixed value — after the value from the programmable controller is multiplied by the scale, the offset amount is added.
- **Set minimum or maximum limits on a value (that is, providing a range of values).**
  HMI tags allow you to set a minimum value and a maximum value that can be written to the programmable controller or server.
  These values do not affect what is read from the programmable controller 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 programmable controller and store it in the value table, but would not be able to write this value to the programmable controller.
  For more information about minimum and maximum, and scale and offset, see Help.

Store values in FactoryTalk View memory

A memory tag can be used to store values without the need for an attached or accessible device.

For example, you might need to store a value in memory:

- To store the result of a calculation.
- Temporarily, for example, a counter or index.
- To maintain information about the system's current state, for example, which graphic display was last opened.

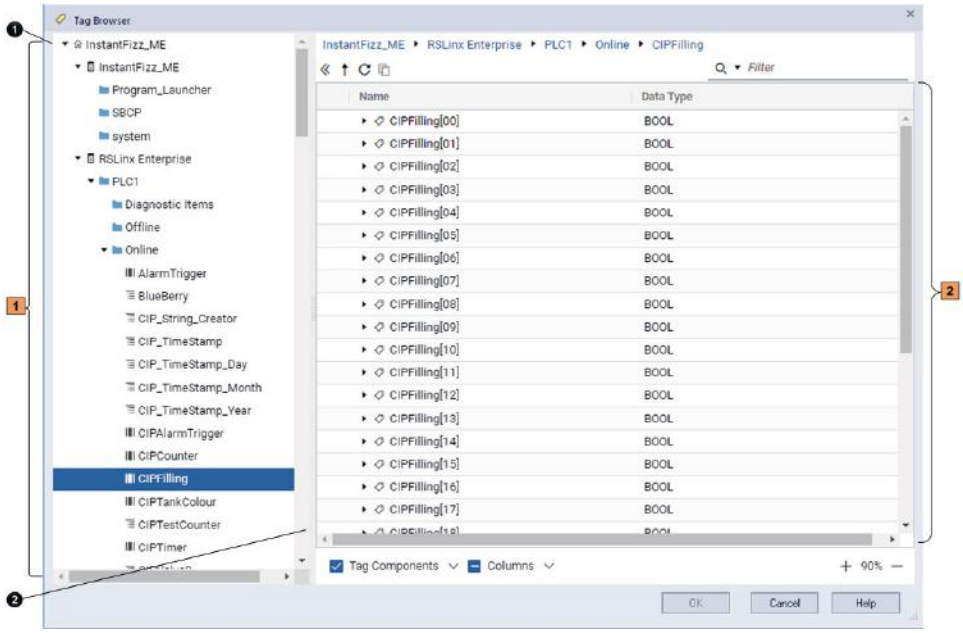
For information about creating HMI memory tags, see Use HMI tags on page 79.

Use the Tag Browser

In FactoryTalk View, when you need to use a tag, you can type its name and path, or you can browse for it.

Use the **Tag Browser** to browse the tag database and select a tag or folder from the editor you are working in.

The **Tag Browser** is divided into two sections. The left side shows the hierarchy of the folders for the HMI tag database. Each device shortcut in FactoryTalk Linx is represented by a folder.



Item	Description
❶	Root folder
❷	For a better view of the folders or tags, drag this bar to the left or the right.
1	Folders pane
2	Tags pane

For information about global object parameters and controller instruction faceplates, see Use parameters and global objects on page 374.

To open the Tag Browser

Use one of the following methods:

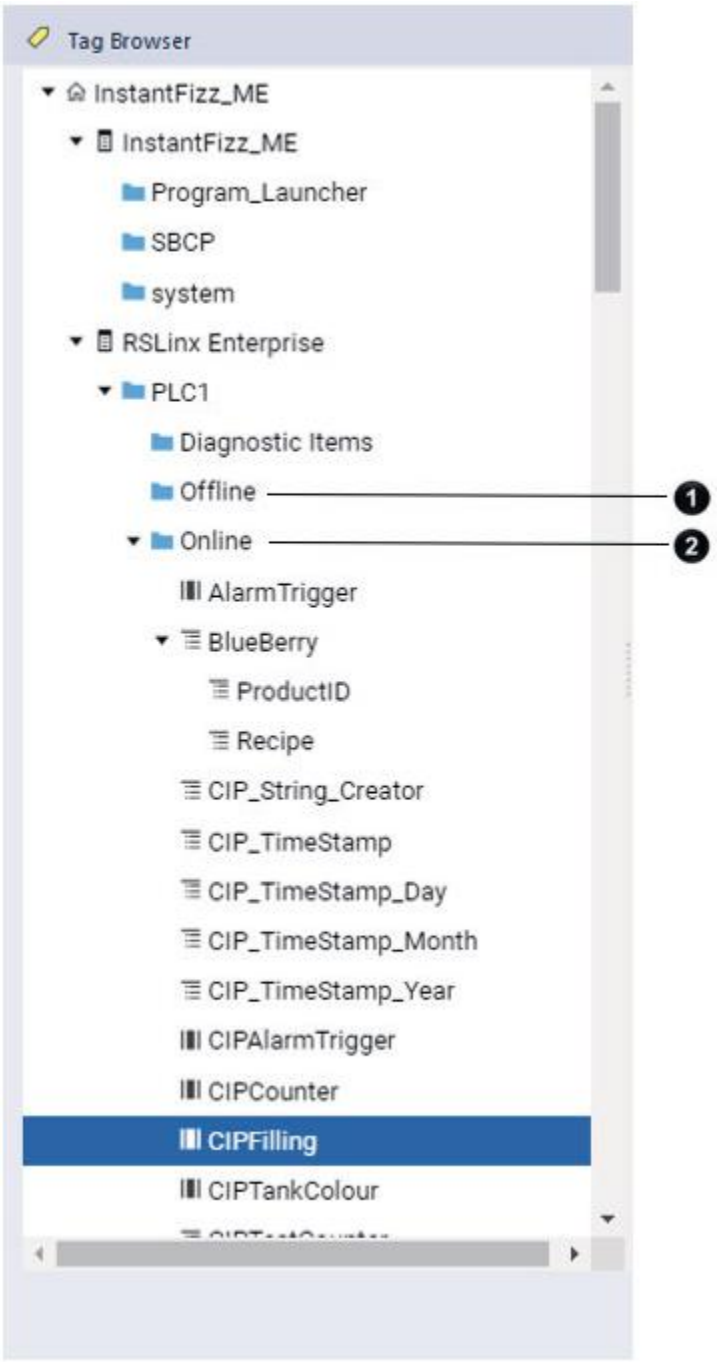
- In the editors that use tag assignments, click the **Browse** button in the **Tags** column.
- In the **Expression** editor, position the cursor where you want to insert a tag name, and then click **Tags**.
- In the **Parameters** editor, double-click where you want to insert a tag name.

For more information about using the **Tag Browser**, see Help.

Browse for offline tags

For each FactoryTalk Linx device shortcut in your application, an **Offline** folder and an **Online** folder are shown in the **Tag Browser**.

In FactoryTalk Linx, you can use the **Offline** folder to browse tags in an RSLogix.acd file. For details about making the .acd file available to browse offline, see FactoryTalk Linx Help.



Item	Description
1	Tags that are available when you are not connected to the controller are located in the **Offline** folder.
2	Tags that are available when you are connected to the controller are located in the **Online** folder.

Use tags and expressions in your application

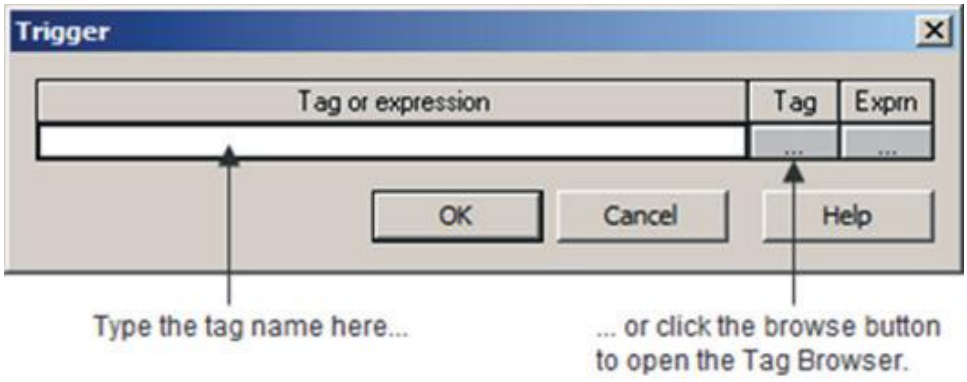
Once you have created tags (either in data servers or in FactoryTalk View), assign them to connections in your application to allow the application to interact with the data source and perform actions based on the tag values.

You can assign tags in these editors:

- **Expressions**
- **Global Connections**
- **Graphic Displays**
- **Global Object Displays**
- **Graphic Libraries**
- **Parameters**
- **Alarm Setup**
- **Information Setup**
- **Macros**
- **Data Log Models**
- **RecipePlus Editor**

Assign a tag

This example shows the **Trigger** dialog box that opens when you click the **Add** button on the **Triggers** tab of the **Alarm Setup** editor:



To assign a tag, use one of these methods:

- In the **Tag or expression** column, type the tag name.
- In the **Tag** column 

click the **Browse** button and then select a tag from the **Tag Browser**.

For information about using the **Tag Browser**, see Help.

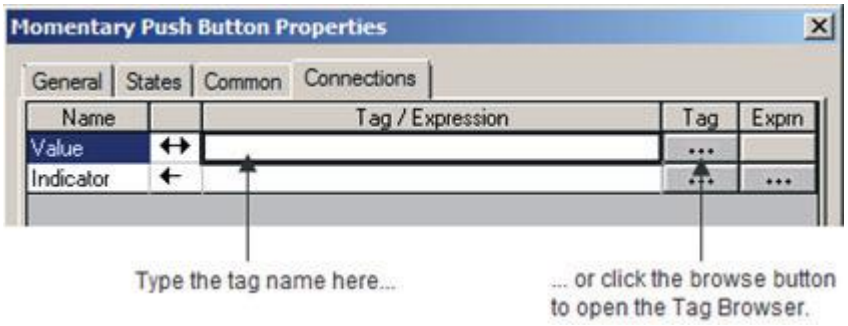
Assign a tag to a graphic object

In the **Graphics** editor, you can assign tags to many of the graphic objects you create.

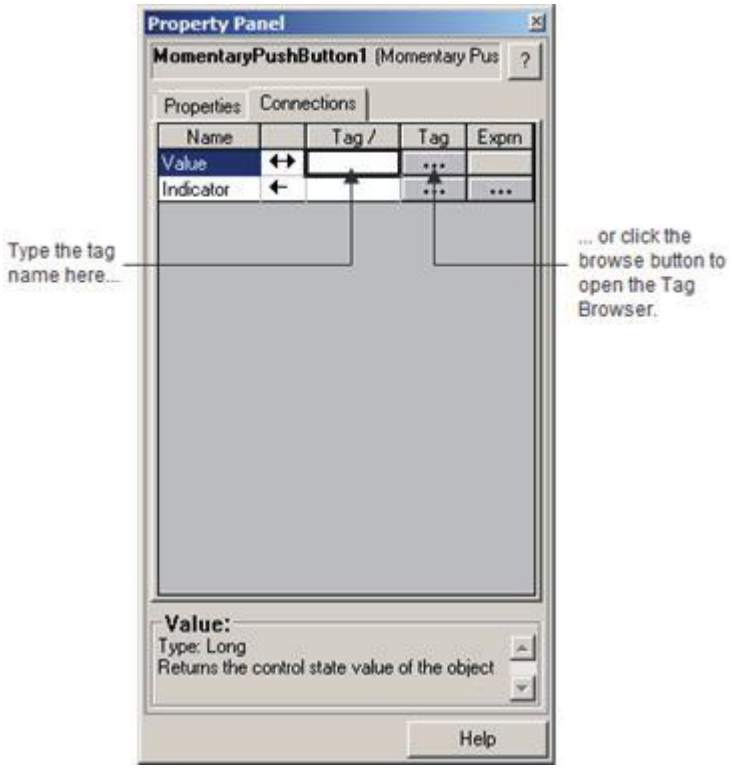
You can use up to 1,000 tags per graphic display. This limit includes the tags contained in embedded variables and expressions. For example, if an expression references the same tag twice, this counts as two tag references.

To assign tags to a graphic object, use one of these methods:

- Double-click the object to open the object's **Properties** dialog box, and then click the **Connections** tab.



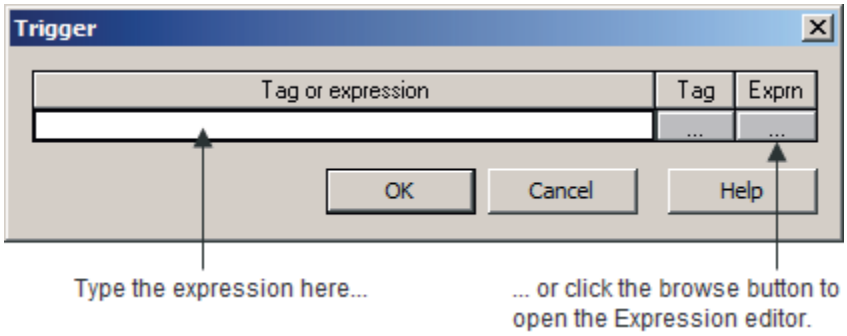
- Select the object, and then assign tags on the **Connections** tab of the **Property Panel**.



Assign an expression

Many of the connections to which you can assign a tag also permit the use of expressions to perform logical or mathematical calculations on tag values. If you assign an expression, FactoryTalk View monitors the expression value rather than the original tag value. For example, your machine might send values to the data source in the form of temperature in degrees Celsius. You could use an expression to convert the value to degrees Fahrenheit, and then monitor the expression result rather than the original tag value.

If you can assign an expression, a **Browse** button is present in the **Exprn** column on the **Connections** tab.



To assign an expression, use one of these methods:

- In the **Exprn** column, click the **Browse** button, and then create an expression in the **Expression** editor. Use this method if you want to check the expression syntax.
- In the **Tag or expression** column, type the expression. The expression syntax is not checked if you use this method.

For more information about expressions, see Use expressions on page 352.

Substitute tag names used in graphic objects

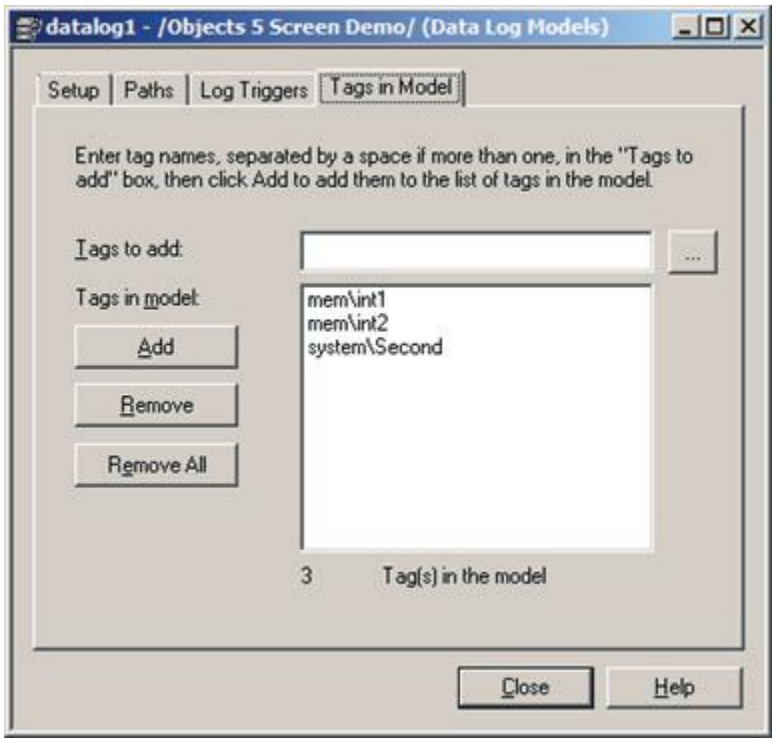
You can use tag substitution or Find and Replace to replace tag names and expressions assigned to the connections for graphic objects. For example, if you assign a tag to multiple objects on a graphic display, and then decide to use a different name for the tag, you can create a new tag and then use tag substitution or Find and Replace to search for all tag references with the old name, and replace them with the new tag name.

You can also use tag substitution to replace the tags contained in embedded variables.

For more information about tag substitution and Find and Replace, see Replace tags using tag substitution on page 303.

Log tag values

Analog and digital tag values can be logged, including both HMI and data server tags. To set up data logging, use the **Data Log Models** editor. For more information, see Set up data logging on page 387.



Assign values to tags

A macro is a list of tag assignments stored in a text file, in the format <tag>=<value>. Each assignment assigns a value to a tag. The value can be in the form of another tag, an expression, a numeric constant, or a string. Whenever you run the macro, the specified values are written to the tags.

For more information about macros, see Use macros on page 421.

You can also use recipes to assign values to multiple tags with a button press. For information about using recipes to assign values to tags, see Set up RecipePlus on page 410.

Use HMI tags

This chapter describes:

- HMI tag types
- How to use the **Tags** editor
- Data sources
- Address syntax for device tags
- How to organize HMI tags
- How to view tag statistics
- Other methods for creating HMI tags
- How to import HMI tags

HMI tag types

You can create and use three types of HMI tags based on the data that they can store:

- *Analog tag* stores a numeric value from a range of values defined for the tag. Use them to represent devices that can have a range of values such as temperature, pressure, flow, or the position of a rotary control.
- *Digital tag* accepts any numeric value. Zero will be written as False (0) and any non-zero number will be written as True (1). Use them to represent devices that have two states: on or off, such as switches, contacts and relays.
- *String tag* stores ASCII strings that can be a string of characters or whole words. The maximum size string allowed is 82 characters. Use string tags to represent devices that use text, such as a bar code scanner that uses an alphanumeric product code.

In addition to the tags you create, the application provides a set of *system tags*. System tags store information generated while the system is running, such as the name of the logged-in user, system time and date, and the like. FactoryTalk View creates system tags when you create an application. The system tags are stored in the folder called *system* in the HMI tag database, and their data source is system. You can use system tags anywhere you would use any other type of tag, but cannot create or edit them. For example, you can create a string display object on a graphic display and assign the system tag, *system\Time*, to the object. At runtime, the object shows the information stored in the system tag, which is the current system time.

Open the **system** folder in the **Tags** editor to see the system tags.

For a list of system tags, see System tags on page 460.

For information about when to use HMI tags versus when to use data server tags, see Work with tags on page 66.

Analog tags that use floating-point values

You can assign analog tags to most of the connections in FactoryTalk View, including both HMI and data server tags. If the analog tag uses a floating-point data type but an integer value is required, the floating-point value is rounded.

For information about the data types available for analog tags, see Help.

How values are rounded

The floating point number is rounded to an integer before it is sent to the connection when an analog tag of the floating point data type is assigned to a:

- Write connection.
- Read connection that will be used to trigger a state, alarm, information message, or local message.

When a floating-point value is rounded to an integer:

- If the number after the decimal is 4 or less, the value is rounded down. For example, 8.495 is rounded to 8.
- If the number after the decimal is 6 or more, the value is rounded up. For example, 8.6 is rounded to 9.
- If the number after the decimal is 5, the value is rounded using a "banker's algorithm" to average out the rounding over time:
  - If the number before the decimal is 0, the value is rounded down. For example, 10.5 is rounded to 10.
  - If the number before the decimal is an odd number, the value is rounded up. For example, 11.5 is rounded to 12.
  - If the number before the decimal is an even number, the value is rounded down. For example, 12.5 is rounded to 12.
  - Floating point values are rounded to a maximum of six digits. Therefore, using a floating-point data type and adding to it a value that exceeds the six significant digits will result in a rounding error. For example, adding any number to 999,999 will result in a rounding error.

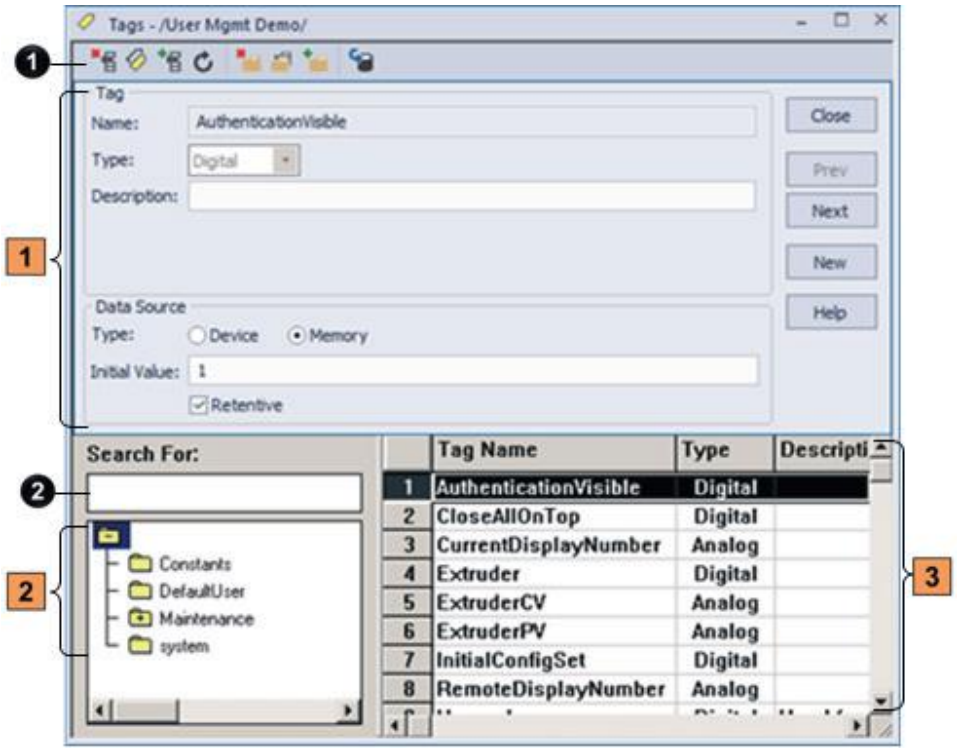
Tip: Floating point values (the default values for numeric input objects) are rounded to six digits. Therefore, using a floating point data type and adding to it a value that exceeds the six significant digits will result in a loss of precision.

This rounding method is also used for Numeric Input Enable buttons that use the implicit decimal position if an integer tag is assigned to the button's **Value** connection. For more information about the Numeric Input Enable button, see Help.

Tags editor

Use the **Tags** editor to:

- Create folders
- Duplicate, nest, and delete folders
- Create and view tags
- Edit, duplicate, and delete tags



Item	Name	Description
❶	Toolbar	Use the tools on the toolbar to: • Delete or duplicate tags and folders • Import PLC tags • Insert rows in the spreadsheet • Refresh the contents of the database with changes from other editors • Create new folders
❷	Query box	Use this section to search for and select a particular tag, or group of tags (using query patterns), that you want to view in the spreadsheet.
1	Form area	In the **Tag** area, specify the tag name, type, description, and other details that vary depending on the tag type. In the **Data Source** area, specify the tag's data source, device, or memory.
2	Folder hierarchy	This section shows the folders that make up the tag database.
3	Spreadsheet	This section shows a list of tags. Use the query box or the folder hierarchy to select the tags you intend to show.

For details about using the **Tags** editor, see Help.

Update the contents of the Tags editor

You can create tags:

- in the **Tags** editor
- as you work in other editors
- by importing them from an existing tag database

To see the result of changes to the database you make in other editors while the **Tags** editor is open, update the contents.

To update the contents of the editor

- From the **View** menu, select **Refresh**, or click the **Refresh** tool.
  When you select **Refresh**, the undo buffer is cleared and recent changes can't be automatically reversed.
  Refresh is not available if you haven't accepted the current tag.

For information about creating tags in other editors, see Other methods for creating HMI tags on page 86.

Search for HMI tags

Use the **Tags** editor's query box to select the tags you want to show in the spreadsheet. This allows you to edit tags in different folders without browsing the folder hierarchy. You can:

- Select a single tag by typing the tag name.
- Select multiple tags by typing wildcard characters.

These are the wildcard characters:

This character	Does this
?	Matches any single character.
*	Matches any number of characters, including the backslash (\) character. Use this wildcard character by itself to show all the tags in the HMI tag database.

For example, to search for all the tags that contain the string "pump" in their name, type *pump* in the **Search For** box. When you press **Enter**, a list of tags containing the string "pump" appears in the spreadsheet.

When you do searches, remember that the backslash in a folder name counts as a character in the tag name.

You can also use **Find** to search for every occurrence of a specific tag or text string throughout different components within FactoryTalk View ME applications. Find also locates all the text strings within each component that refer to the tag.

Find a tag or text string

Follow the steps below to find a tab or text string.

To find a tag or text string

1. From the **Tools** menu, select **Find**, or on the toolbar, click  .
2. Type the text string in **Find what**, or identify the tag to search for:
   - Click the browse button () next to **Find what**.
   - From the **Tag Browser**, navigate to the correct folder in the left pane, select the tag in the right pane, and 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):
   - Click the browse button next to **Find within**.
   - From the **Find within** dialog box, select the product component types to be searched and click **OK** to save the selection and close the dialog box.
4. Identify the word search limitations in **Find Options**.
5. Select the **Direction** in which to search:
   - Select **Up** to search for the tag or text string in all components before the open one.
   - Select **Down** to search for the tag or text string in all components after the open one.
6. 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.
7. 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.

Find all references to a tag or a text string

Follow the steps below to find all references to a tag or text string.

To find all references to a tag or text string

1. From the **Tools** Menu, select **Cross Reference**, or on the toolbar, click  .
2. Type the text string in **Find what**, or identify the tag to search for:
   - Click the browse button () next to **Find what**.
   - From the **Tag Browser**, navigate to the correct folder in the left pane, select the tag in the right pane, and 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):
   - Click the browse button next to **Find within**.
   - From the **Find within** dialog box, select the product component types to be searched and click **OK** to save the selection and close the dialog box.
4. Click **Search** to show all the found items in a spreadsheet form.
5. You can double-click any of the found items to open it with the appropriate editor.

HMI tag data source

When creating an HMI tag, you must specify a data source. The data source determines whether the tag receives its values externally or internally. For more information about data sources, see Data source on page 68.

Device

A tag with a device as its data source receives its data from a source external to FactoryTalk View. The data can come from a programmable controller or another device via FactoryTalk Linx, or from another OPC server. You can use up to 5,000 device tags.

For analog tags with Device as the data source, the data type determines the format used when reading and writing tag data to the data server.

Do not create HMI Tags that are pointing to a System Tag. When a project contains an HMI tag pointing to a System Tag, users may fail in creating an MER or have runtime performance issues.

Memory

A tag with memory as its data source receives its data from (and stores values in) the FactoryTalk View internal value table. Tags with memory as the data source do not count toward the total tag limit.

For analog tags with Memory as the data source, select the data type that matches the format of the data you will store in the tag. Choosing **Default** is the same as choosing **Floating Point**.

Address syntax for device tags

For device tags, you must provide the address and path to the OPC server that is connected to the data source. For example, the OPC server could be connected to a device such as a programmable controller.

The syntax for OPC tag addresses is:

```
[AccessPath]Address
```

or

```
::[AccessPath]Address
```

where :: indicates that the address resides in a data server in the application. The two colons are necessary when the address contains a colon, for example, N7:0.

The square brackets are part of the syntax. They do not indicate optional parameters.

The access path is one of the following:

- For communications with a FactoryTalk Linx data server, the access path is the name of the device shortcut in FactoryTalk Linx.
- For communications with other OPC servers, the access path might be optional. For information about the syntax for the access path, see your OPC server documentation.

Example: Studio 5000 addressing

We recommend that you use the Tag Browser or Find and Replace to select OPC items. When you use the Tag Browser the correct syntax and data types are selected automatically.

However, if you want to define tags before setting up communications, you can type the address manually, as shown in this example. If you type the address manually, be sure to select the data type for the tag that matches the data type used by the Studio 5000 processor. If you use the Default data type, values are treated as floating-point values.

This example shows you how to specify the OPC topic, item name, and bit level of an item that uses the integer data type in a Studio 5000 processor.

OPC topic and item addressing

Studio 5000 uses this addressing syntax:

[*Device shortcut*]*item name* for FactoryTalk Linx.

[*OPC topic name*]*item name* for RSLinx Classic.

In this example, the FactoryTalk Linx device shortcut name for the Studio 5000 processor is *CLGX*. To connect to an OPC item called *Motor*, type this address:

```
[CLGX]Motor
```

Bit level addressing

You can gain access to the bit level of an item that uses the integer data type by appending .# to the end of the tag address, where # is the number of the bit you want access to.

For example, to gain access to the first bit of the integer item *Motor*, type this address:

```
[CLGX]Motor.0
```

Organize HMI tags

Organized tags make finding and using tags faster and simpler. To organize your tags, try these tips:

- Establish naming conventions on page 85
- Use folders to group tags on page 85.

Establish naming conventions

Naming conventions enable you to use wildcard characters most effectively when searching for and showing tags during development.

A tag name can be up to 255 characters long, including a folder name. If you create a folder, the folder name becomes part of the tag name. The backslash (\) after a folder name counts as a character.

A tag name can contain these characters:

- A to Z
- 0 to 9
- underscore (_) and dash (-)

A tag name cannot contain spaces.

A tag name can be mixed case. A tag name preserves upper and lower cases for readability but is not case-sensitive. For example, the tag name *MixerValve1* is the same as *mixervalve1*.

When a tag name starts with a number or contains a dash, enclose the name in curly brackets { } when you use it in an expression, for example, {N33-0}. For more information about using tags in expressions, see Use expressions on page 352.

Use folders to group tags

To organize tags, create a folder to store tags that are related to one another. To separate the folder name from the rest of the tag name, use a backslash (\). For example, tags in the folder called *Pump* would start with *Pump*\.

For greater organization, you can nest folders. For example, you can organize the HMI tag database first by area, then by machines in the area, and finally by devices in each machine. The result might be *Area1**Machine1**Pump*.

To create similar groups of tags quickly, create one folder with its tags and then duplicate the folder.

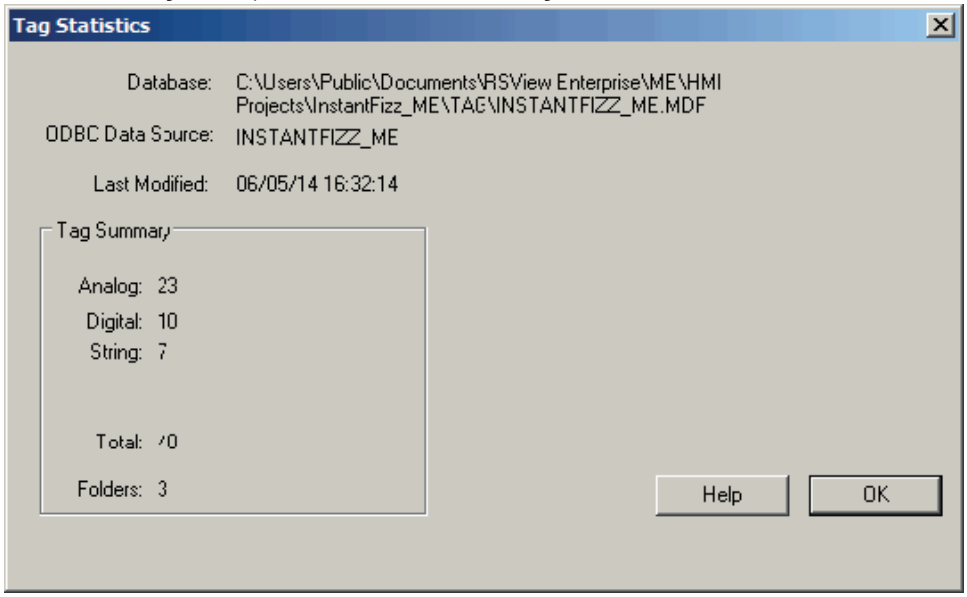
View tag statistics

The **Tag Statistics** dialog box provides the following information for the current database:

- HMI tag database path
- The ODBC data source
- The date the tag database was last modified
- The number of analog, digital, and string tags
- The total number of HMI tags
- The number of folders in the database

To view tag statistics

- With the **Tags** editor open, from the **View** menu, select **Tag Statistics**.



Other methods for creating HMI tags

In addition to creating tags in the **Tags** editor, you can import tags from a PLC or SLC database.

For details, see Import tags from a PLC database on page 87.

Create tags as needed in other FactoryTalk View editors

You can create tags as needed in any editor or dialog box that uses tags. You can add a tag to the database by doing the following:

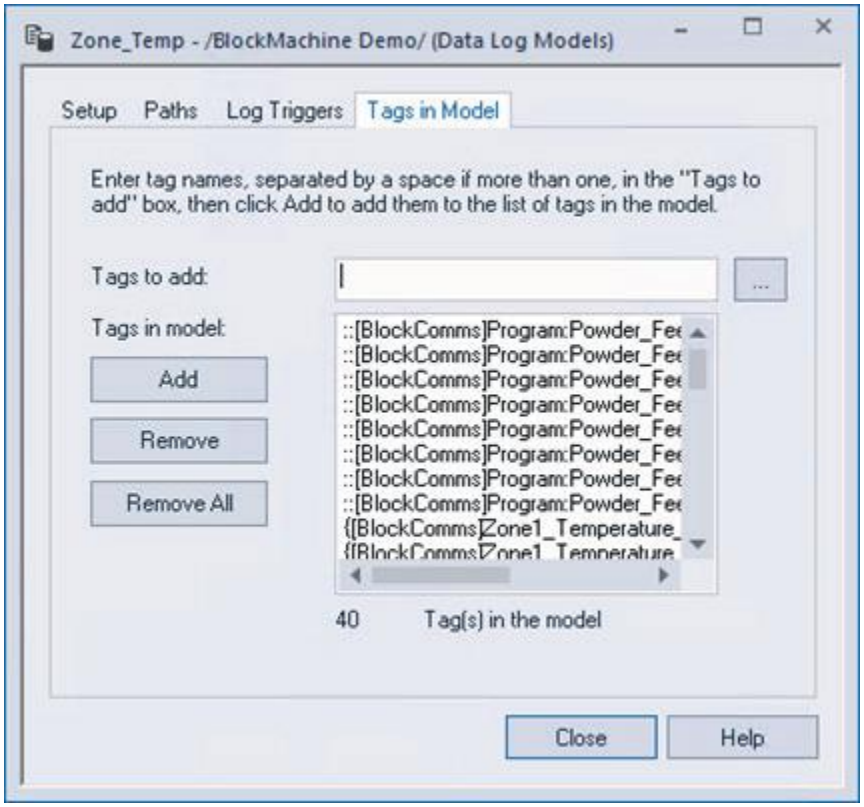
- Type a new tag name in the **Tag or expression** column. Track the tag name, and add the tag in the **Tags** editor when convenient.

Create tags as needed in the Data Log Models editor

In the **Data Log Models** editor, on the **Tags in Model** tab, you can add a tag to the database by doing the following:

- Type a new tag name in the **Tags to add** box and click **Add**. You are warned that the tag does not exist. Click **Yes** to add the tag to the list of tags in the datalog model. Track the tag name and add the tag in the **Tags** editor when convenient.

For more information about the **Data Log Models** editor, see Set up data logging on page 387.



Import tags from a PLC database

Use the **Import PLC Tags** dialog box to selectively import tags from a PLC or other database into your application's HMI tag database. Tags imported in this way are copied into the database. They are not shared with the source database. This means changes to tags in your application do not affect the database from which they have been imported, and vice versa.

You can import tags from any of these databases:

- RSLogix 500© internal database, with file extension .rss
- RSLogix 5 internal database, with file extension .rsp
- RSLogix 5 or RSLogix 500, saved as an external database, with file extension .ctd (before importing, explicitly export to the .ctd format to be sure of including the latest changes)

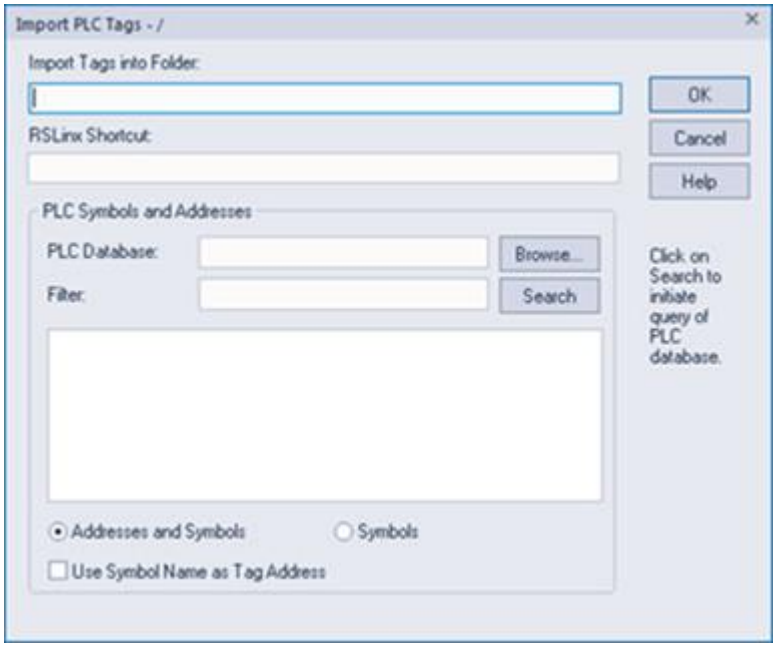
For PLC and SLC addresses, the **Import PLC Tags** dialog box shows only addresses that are used in the symbol or address list of the PLC programming software.

Open the Import PLC Tags dialog box

Use the following steps to open the **Import PLC Tags** dialog box.

To open the Import PLC Tags dialog box

- In the **Tags** editor, from the **Edit** menu, select **Other Databases**, or from the editor's toolbar, select  .



For more information about using the **Import PLC Tags** dialog box, see Help.

Use the Tag Import and Export Wizard

You can also import PLC or SLC databases into FactoryTalk View using the Tag Import and Export Wizard.

You can use the wizard to:

- Export your application's tags to a .csv file.
- Import tags from another FactoryTalk View application's tag .csv file.
- Merge tags from another FactoryTalk View application (that is, import them directly from the application, without first creating a tag .csv file).
- Import tags from RSLogix 5 or RSLogix 500, saved as an external database, with file extension .ctd (before importing, explicitly export to the .ctd format to be sure of including the latest changes).

Open the Tag Import and Export Wizard

Use the following steps to open the Tag Import and Export Wizard.

To open the Tag Import and Export Wizard, use one of these methods:

- In FactoryTalk View Studio, from the **Tools** menu, select **Tag Import and Export Wizard**.
- From the Windows **Start** menu, select **All apps** > **Rockwell Software** >**Tag Import and Export Wizard**.

For details about using the wizard, see the Tag Import and Export Wizard Help.

Set up global connections

This chapter describes:

- Global connections
- How to update the date and time
- How to change displays
- How to control display changes remotely
- How to use parameter files or parameter lists with display changes
- How to close On Top displays
- How to print displays
- How to run macros
- How to set up the backlight intensity remotely
- How to monitor runtime RAM usage

What is a global connection?

Global connections are connections that apply to your entire runtime application. Global connections allow the data source to control or interact with your application at runtime.

Use global connections to:

- Update the date and time at the data source using the runtime terminal's date and time.
- Update the date and time on the runtime terminal using the data source date and time.
- Notify the data source of the current display number.
- Notify the data source when a display is printed.
- Change the display on the runtime terminal remotely (from the data source).
- Close On Top displays
- Apply parameter files or parameter lists to the tag placeholders in the changed displays.
- Print the current display from the data source.
- Run up to five macros from the data source, when a tag or expression result changes.
- Set the intensity of the backlight on a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal.
- Monitor runtime RAM usage for troubleshooting purposes.

The use of each global connection is optional.

Use the **Global Connections** editor to set up global connections. For more information, see Help.

Update the date and time

Use the date and time connections on the **Date and Time** tab of the **Global Connections** editor to update the date and time. You can assign any or all of the date and time connections, or leave them unassigned.

Normally, you would assign only one set of date and time connections, either to update the date and time at the data source from the terminal, or to update the terminal's internal clock from the data source.

The values of assigned date and time connections are updated in the order in which the connections are listed in the **Global Connections** editor.

Update the date and time at the data source from the terminal

To update the date or time at the data source from the terminal's internal clock, assign a tag to one or more of these connections:

This connection	Contains this range of values
Year	00-99
Day of Week	1-7, with 1 being Sunday
Month	1-12
Day of Month	1-31
Hour	0-23
Minute	0-59
Second	0-59

Specify how frequently to update the values of the assigned connections at the data source (from the terminal's internal clock) by typing a time in the **Date and time output interval** box.

Update the date and time at the terminal from the data source

This option is especially useful if you have several terminals connected to the same data source. Instead of the operator resetting the date and time manually on each terminal, the data source can reset all internal clocks at once.

To update the date or time in the terminal's internal clock from the data source, assign a tag or an expression to the Remote Date and Time connection. Also assign a tag or an expression to one or more of these connections:

This connection	Contains this range of values
Remote Year	00-99 or 0000-9999
Remote Month	1-12
Remote Day of Month	1-31
Remote Hour	0-23
Remote Minute	0-59
Remote Second	0-59

When the Remote Date and Time connection value changes to a new nonzero value, the terminal resets its internal time and date to the time and date read from the individual remote date and time connections.

The date and time are updated at the rate specified in the **Maximum update rate** box, but changes are not made until the Remote Date and Time connection changes to a new nonzero value.

Change displays

Use these global connections on the **Display** tab of the **Global Connections** editor to monitor and control display changes:

- **Replace Display Number** — notifies the data source of the number of the Replace display that's currently open.
- **Remote Display Number** — allows the data source to change the display on the runtime computer.

Control display changes remotely

To control display changes remotely, you can set up the data source to open graphic displays using global connections.

For example, the **Remote Display Number** connection is a global connection that you can use to control display changes from the data source.

For more information, see About global connections on page 89.

Remote display changes and security

If you set up the data source to open graphic displays remotely, remote display changes occur whether the logged-in user has security access to a given display.

Set up remote display changes

This section outlines the steps for setting up remote display changes. For more detailed information about the **Global Connections** editor, see Help.

To set up remote display changes

1. In the **Graphics** editor, in the **Display Settings** dialog box, every new display is assigned the number 1 by default. Therefore, to use remote display changes, you must assign a unique display number to each Replace and On Top display that you want to change remotely.
   For more information about graphic displays, see Use graphic displays on page 252.
2. In the **Global Connections** editor, assign a tag or an expression to the **Remote Display Number** connection.
3. If you assign a tag to the **Remote Display Number** connection, set up the data source to send a display's number to the connection when you want the display to open.
4. If you assign an expression to the **Remote Display Number** connection, FactoryTalk View monitors the values of the tags used in the expression, and uses the expression result to determine which display to open.
5. If you assign a tag to the **Close An On Top Display** or **Close all On Top Displays** connection, or both, set up the data source to send a display's number to the connection when you want one or more displays to close.
6. If you assign an expression to the **Close An On Top Display** or **Close all On Top Displays** connection, FactoryTalk View monitors the values of the tags used in the expression, and uses the expression result to determine which display to close.

Print displays through global connections

Use these global connections on the **Display** tab of the **Global Connections** editor to monitor and control display printing:

- **Display Print Active** — notifies the data source that a display print is in progress.
- **Remote Display Print** — allows the data source to trigger a display print.

The operator can also print the current display by pressing the display print button. For information about setting up this button, see Help.

When the data source or the operator prints a display, everything on the screen is printed, including the current display, pop-up windows, and any visible background applications.

If you plan to print graphic displays remotely, turn off the screen saver. If the screen saver is on when a remote display print is triggered, only the screen saver image is printed.

For information about turning off the screen saver on personal computers, see your Windows documentation.

For information about turning off the screen saver on a MobileView or PanelView Plus family terminal, see the terminal's *User Manual*.

Close On Top displays

Use these global connections to close displays remotely:

- **Close an On Top display** — notifies the data source of the number of the On Top display to be closed.
- **Close All On Top Displays** — notifies the application to close all On Top displays whose display number is not zero.

Only replaceable On Top displays can be closed remotely.

Apply parameters to changed displays

Use these global connections to replace tag placeholders in displays that use parameters when they open:

- **Parameter file** — identifies the parameter file for the display being opened remotely.
- **Use Parameter List** — notifies the application that a parameter list is to be used to provide the parameters to the display being opened remotely.
- **Parameter Tag #1-10** — the tag values assigned to these global connections will replace the tag placeholders on the display being opened remotely.

Run macros

Use these global connections on the **Macro** tab of the **Global Connections** editor to run macros:

This connection	Runs this macro
Remote Macro1	Macro1
Remote Macro2	Macro2
Remote Macro3	Macro3
Remote Macro4	Macro4
Remote Macro5	Macro5

Create the macros in the **Macros** editor. Make sure you name them exactly as shown in the right column of the preceding table (with no space in the name).

When the tag or expression assigned to one of these connections changes to a new nonzero value, FactoryTalk View runs the associated macro.

Leave time for a macro to run before starting a new macro. Otherwise, FactoryTalk View will queue the macros, which consumes memory and could lead to system instability. When you shut down FactoryTalk View ME Station, if a message about executing macros is shown for more than a few seconds, this means that macros have been queued.

For information about creating macros, see Use macros on page 421.

Set up backlight intensity remotely

For applications that will run on a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, use the Remote Backlight Intensity connection on the **Backlight** tab of the **Global Connections** editor to set the backlight intensity of a runtime terminal remotely from a data source.

To update the backlight intensity of the runtime terminal from the data source, assign a tag or an expression to this read-only connection.

This connection	Contains this range of values
Remote Backlight Intensity	0-100

At runtime, the application reads the value at the data source and sets the backlight to the percentage of its maximum intensity corresponding to the value. For instance, if the value of the connection is 25, the backlight will be set to 25 percent of its maximum intensity. This value will override the terminal's own backlight intensity setting.

If the value of the connection is greater than 100, the backlight will be set to its maximum intensity. If the value is less than 0, the backlight is set to its minimum intensity. That is, it will be turned off.

Monitor runtime RAM usage

Sometimes it is helpful to monitor runtime RAM usage for troubleshooting purposes. If your application seems to be running slowly, consuming too much memory, or interfering with the operation of other applications on the same personal computer or terminal, it is useful to monitor the exact amount of memory the applications are consuming.

Tip: To reduce runtime RAM usage, reduce the number of displays or tags in your application, and restart the application regularly (for example, once a week). Adding more RAM to the terminal will not help. For more information, see Knowledgebase Document ID: QA4385 - FactoryTalk View ME Terminal Memory Usage Best Practices.

The PanelView Plus, PanelView Plus Compact, and PanelView Plus CE terminals' operating systems have a 32 MB RAM process limit. On these terminals, FactoryTalk View ME Station has a tag connection threshold of 26 MB. Screen change performance will slow down significantly as the system adds and removes tag connections from the cache.

Use these global connections on the **Memory** tab of the **Global Connections** editor to monitor runtime RAM usage:

This connection	Provides this information, in kilobytes (Kb)
ME Runtime RAM Usage	The amount of memory being used by your application (and by the MERuntime.exe process)
Total RAM Usage	The total amount of memory being used by all applications (and processes) on the personal computer or terminal
Available RAM	The remaining amount of memory available
Remote RAM Usage	Assign a tag or expression to this connection. When it has a nonzero value, data is sent to the other three connections, if assigned.

You can monitor the values of the **MERuntime RAM Usage**, **Total RAM Usage**, and **Available RAM** connections using data logging or using graphic objects such as numeric displays. The connections are updated every ten seconds as long as the **Remote RAM Usage** connection has a nonzero value. Set up the data source to set this connection to a nonzero value when you want memory monitoring to occur, or assign the same tag or expression to a graphic object such as a **Maintained** push button that the operator can use to turn memory monitoring on and off.

These connections are available when you run FactoryTalk View ME Station on a personal computer and on the MobileView and PanelView Plus family terminals.

For PanelView Plus family terminals, you can also turn on the display of RAM usage statistics at runtime. To do this, in the **Configuration Mode** window, select **Runtime RAM Usage**, and then make your selections in the **Runtime RAM Usage** window. For more information, see the terminals' *User Manual*.

Set up alarms

This chapter describes:

- Steps for setting up alarms
- How to prepare to set up alarms
- How alarms work
- Alarm trigger data types
- Tips for using array tags
- Optional alarm connections.
- How to create alarm messages in multiple languages
- The [ALARM], [ALARM BANNER], [ALARM MULTI-LINE], [STATUS], and [HISTORY] displays
- How to use displays from the library in your application
- How to create your own alarm display
- How to open and close the alarm display
- How the alarm list, alarm banner, and alarm status list graphic objects work
- How to use buttons to acknowledge, silence, clear, and delete alarms
- How to use buttons to sort alarms and reset alarm status

What is an alarm?

An alarm occurs when something goes wrong or is about to go wrong. Alarms can signal that a device or process has ceased operating within acceptable, predefined limits, and can indicate breakdown, wear, or process malfunctions. Alarms are also used to indicate the approach of a dangerous condition.

Alarms are an important part of most plant control applications because an operator must know the instant something goes wrong. It is often equally important to have a record of the alarm and whether the alarm was acknowledged.

Use alarms to notify the operator when a situation requiring immediate attention occurs. When an alarm condition occurs, the application can open an alarm graphic display, set off an audible signal, send a message to a printer or a display, send the alarm trigger value to the data source, or do any combination of these. All triggered alarms in the application are logged in the alarm history.

PanelView Plus 7 Standard applications have a limit of 500 alarm messages. PanelView Plus 6 Compact applications have a limit of 200 alarm messages.

Multiple language alarm messages

A key component of any alarm system is the messages that appear when alarms occur. FactoryTalk View supports alarm messages in multiple languages. For information about setting up alarm messages in multiple languages, see Create alarm messages in multiple languages on page 108.

Other multiple language alarm features

FactoryTalk View supports using multiple languages for all captions on alarm objects. In addition, these alarm features can be translated into multiple languages:

- Title of the alarm history report
- Title of the alarm status report

Alarm objects and reports use the time formats for the current application language. For more information about using multiple languages, see Set up language switching on page 160.

Prepare to set up an alarm

As your application is running, information is continually sent to the data source about the state of the various processes. For example, your application might be monitoring whether a valve is open or closed, or the temperature in a boiler. Values representing the status of these processes are sent to the data source. For more information about data sources, see Data source on page 68.

The first step in setting up alarms is determining which processes to monitor for alarm conditions.

Tags and expressions

Before setting up alarms, you must set up a data server or HMI tags corresponding to the addresses at the data source that will store the values you want to monitor for alarm conditions. You can monitor analog and digital tags for alarm conditions, including both HMI and data server tags. You cannot monitor string tags.

For information about setting up data server tags, see your OPC data server documentation. For information about creating HMI tags, see Use HMI tags on page 79.

You can also use expressions to perform logical or mathematical calculations on tag values, and then monitor the expression value rather than the original tag value. For example, you could use an expression to monitor whether a tag value has increased or decreased beyond a threshold value: If Tag1 > 90 then 1 else 2.

For information about creating expressions, see Use expressions on page 352.

Identify alarm conditions

Once you have identified the processes you want to monitor for alarms, and the tags or expressions that will store the values representing the status of the processes, you must determine the acceptable range of values for each tag or expression. Then you can set up alarms to notify the operator when a value is outside the normal operating range.

Import and export alarm setup files

The Alarm Import Export Wizard in FactoryTalk View Studio allows you to export alarm information to an XML file, and to import an alarm setup XML file.

For example, you can export an application's alarm setup information, import the setup to another FactoryTalk View application, and then modify it in the new application to suit your requirements. Or, you can modify the file before importing it to the new application.

You can use the Alarm Import Export Wizard to import alarm setup information that has been created using an external programming tool or editor, or you can import a FactoryTalk View XML file.

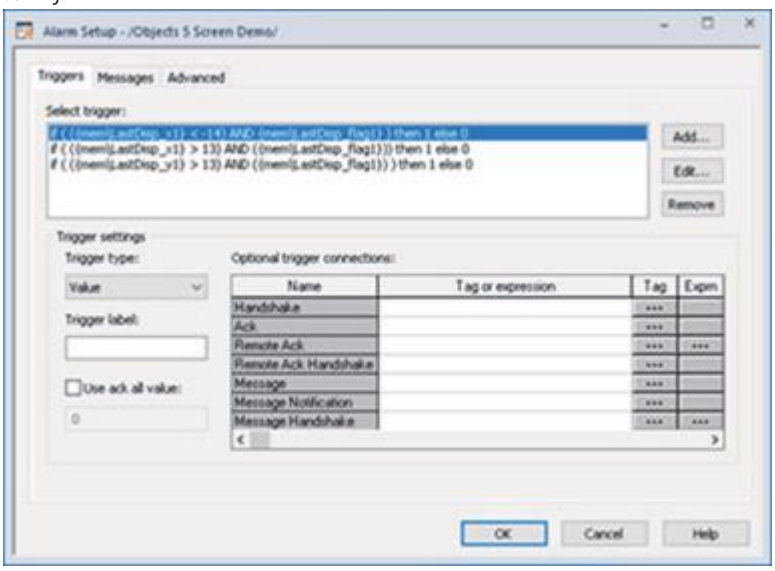
For more information about importing and exporting alarm setup files, see Import and export alarm XML files on page 430.

Set up an alarm

After the preparation on page 96, use the following steps to set up an alarm.

To set up an alarm

1. In the **Alarm Setup** editor:
   - On the **Triggers** tab, set up alarm triggers (the tags or expressions to monitor).
   - On the **Messages** tab, define the alarm messages and their trigger values.
   - On the **Advanced** tab, specify the graphic display to show alarm messages at runtime and specify time settings.

   Also use this editor to specify trigger types, **Acknowledge all** values, the maximum alarm log file size, the time settings (hold time and update rates), and optional connections.
2. In the **Startup** editor, verify that the **Alarms** box is selected.
3. (optional) In the **Graphics** editor, modify the default [ALARM] display or create your own graphic display to use for alarms. For example, if you won't be using audible alarm signals, edit the default display to remove the silence alarms button.
   For information about graphic displays, see Use graphic displays on page 252.
4. Test alarms on the runtime system.

How alarms work

This section provides an overview of the key components of your application's alarm system, and describes how the different parts work together.

Alarm triggers and trigger values

You specify the tags and expressions (also known as connections) to monitor for alarm conditions by creating an alarm trigger for each connection.

Each alarm trigger can generate one or more alarm messages, associated with different trigger connection values. For each alarm trigger, you specify the trigger values that will generate alarm messages, and create the messages to show for the trigger values.

The trigger value can be a nonzero integer value or a bit position, depending on which trigger data type you assign. For more information about trigger data types, see Alarm trigger data type on page 103.

An alarm's **Trigger** and **Remote Ack** settings should not be configured to use the same tag. Doing so may cause unpredictable alarm acknowledgment behaviors.

Filter alarm triggers in multiple languages

FactoryTalk View 5.00 and later supports alarm triggers in multiple languages. When you create an alarm trigger, its name is in the current application language. You can export alarm triggers for translation and then import them back to the application. For details, see Set up language switching on page 159.

Various graphic objects allow you to specify the alarm triggers, to which the object's action applies, using the name of the alarm trigger. This is called filtering alarms. When you translate trigger names into other languages, the default language is used to show the trigger name in the object's selection list. These graphic objects allow you to specify alarm triggers using a selection list called the Trigger Label Selector:

- Alarm list
- Alarm status list
- Alarm banner
- Clear alarm history button
- Acknowledge all alarms button
- Print alarm history button
- Print alarm status button

If the trigger label name has not been set up in the default language, the Trigger Label Selector shows a question mark (?) instead of the trigger label name. To correct the display, you must import the trigger labels for the default language.

You can also manually type in the trigger label names to use to filter alarms. If you manually type in the trigger names, specify the names in the default language.

If you change the default language, you must re-specify the trigger labels for each of the graphic objects listed earlier, using the trigger label names in the new default language. For more information about the default language, see The default language on page 159.

Alarm notification methods

You can use any combination of these methods to notify the operator that an alarm condition has occurred:

- Open an alarm graphic display containing an appropriate alarm message.
- Set off an audible signal (if the application is running on a personal computer).

- Send a message to a printer.
- Send a message to the data source.

If you set up alarm messages in multiple languages, make sure that the data source can either receive the message in Unicode format or convert it into ASCII characters for all the languages you will be using.

To use these methods, select the **Display**, **Audio**, **Print**, and **Message to Tag** options when you create your messages.

If you want to keep a permanent record of alarms, be sure to select the **Print** option. You can also print the contents of the alarm log file periodically, as described on Methods for printing alarm information on page 102.

Show alarm information

FactoryTalk View comes with the following alarm displays:

- [ALARM], in the **Displays** folder, contains an alarm banner graphic object that is set up to show the newest active alarm in the alarm log file.
- [ALARM BANNER], a display in the **Libraries** folder, contains an alarm banner graphic object that is set up to show the newest active alarm in the alarm log file.
- [ALARM MULTI-LINE], a display in the **Libraries** folder, contains an alarm list graphic object that is set up to show all the alarms in the alarm log file: active, inactive, acknowledged, and unacknowledged.
- [STATUS], a display in the **Libraries** folder, contains an alarm status list graphic object that is set up to show active alarms. It contains a Goto display button for opening the [HISTORY] display. This display is available in two sizes.
- [HISTORY], a display in the **Libraries** folder, contains an alarm list graphic object and a Goto display button for opening the [STATUS] display. The alarm list is set up to show all the alarms in the alarm log file: active, inactive, acknowledged, and unacknowledged. This display is available in two sizes.

By default, the [ALARM] graphic display opens automatically at runtime when an alarm is generated.

You can accept the default, edit the [ALARM] display, use one of the other alarm displays, or create your own display. If you select not to automatically open a graphic display for alarm notification, an alternative is to provide the operator with a way to open an alarm display when desired.

For more information, see:

- The [ALARM] display on page 114
- The [ALARM BANNER] display on page 115
- The [ALARM MULTI-LINE] display on page 115
- The [STATUS] display on page 116
- The [HISTORY] display on page 117
- Create your own alarm display on page 118
- The alarm list graphic object on page 119
- The alarm banner graphic object on page 120
- The alarm status list graphic object on page 121

Interact with alarms

Depending on which alarm objects you set up, the operator can:

- Acknowledge one or all alarms, or the alarms for a specific alarm trigger
- Clear and delete alarms
- Silence alarms (if the application is running on a personal computer)
- Sort alarms
- Reset the status of all alarms
- Print alarms

For information about the alarm button graphic objects the operator can use to work with alarms, see Use buttons with the alarm history and alarm objects on page 122.

Use the data source to interact with alarms

You can also set up remote acknowledgment, silencing, and resetting of alarms, to allow the data source to perform these functions. And you can set up the data source to determine which types of alarms to show (active, inactive, acknowledged, and unacknowledged), and to close the alarm display. These topics are discussed later in the chapter.

Methods for acknowledging alarms

A single alarm is acknowledged when:

- The operator selects an alarm in the alarm list and presses the acknowledge alarm button, an enter button object, or the **Enter** key on an external keyboard or keypad.
- The alarm banner has focus and the operator presses the acknowledge alarm button.

A single alarm for a particular trigger is acknowledged when:

- The trigger's **Remote Ack** connection value changes to the alarm's trigger value (or bit position, for bit triggers).

All alarms for a particular trigger are acknowledged when:

- The trigger's **Remote Ack** connection value changes to the **Acknowledge all** value. The **Acknowledge all** value is specified in the **Use ack all value** box in the **Alarm Setup** editor.
- The operator presses an acknowledge all alarms button that is set up to acknowledge the alarms for a specific alarm trigger.

All alarms in the alarm log file are acknowledged when:

- The operator presses an acknowledge all alarms button that is set up to acknowledge all alarms.
- The **Remote Ack All** connection value changes to a new nonzero value.

For more information about acknowledging alarms, see Use alarm buttons to acknowledge, silence, clear, and delete alarms on page 123.

Tip: The operator can send button presses to objects that do not have focus. For information about linking buttons to objects, see Link buttons to objects on page 332.

Methods for clearing and deleting alarms

Alarms are deleted from the alarm log file and cleared from all alarm lists and alarm banners when the operator presses the clear alarm history button. You can set up the button to clear alarms for all alarms, or for a specific alarm trigger.

The alarm in the alarm banner is cleared when the operator presses the clear alarm banner button. It is not deleted from the alarm log file.

For more information about clearing and deleting alarms, see Create and delete messages on page 124.

Methods for silencing alarms

All alarms are silenced when:

- The operator presses the silence alarms button.
- An alarm whose **Audio** property is turned on is acknowledged.
- The **Remote Silence** connection value changes to a new nonzero value.
- The operator presses the clear alarm history button.

Method for sorting alarms

When the operator presses the sort alarms button, the sort order toggles from time to trigger value or from trigger value to time. The sort order applies to all alarms in all alarm lists and in the alarm log file.

Methods for resetting alarms

Resetting alarms affects how alarms are shown in the alarm status list. When alarms are reset:

- The accumulated time an alarm has been in alarm is reset to 0. If the alarm is still active, the time begins accumulating again.
- The number of times the alarm has been triggered is reset to 0. If the alarm is still active, the number changes to 1.
- The date and time are sent to the system tag *AlarmResetDateAndTimeString*.

The alarms for a specific alarm trigger are reset when:

- The operator presses a clear alarm history button that has been set up to clear and reset alarms for a specific alarm trigger.

All alarms are reset when:

- The operator presses a reset alarm status button.
- The operator presses a clear alarm history button, if the button has been set up to clear and reset all alarms.
- The **Remote Status Reset** connection value changes to a new nonzero value.

For more information about resetting alarms, see Reset alarm status on page 125.

Methods for printing alarm information

In addition to automatically printing each alarm message as it occurs, you can keep a printed record of alarms:

- Use the Print Alarm History button. When the operator presses the button, a report of the contents of the alarm log file is printed. The report can include the time alarms occurred and were acknowledged.
  You can set up this button to print the alarm history for a selected alarm trigger, or for all alarms.
  If you set up alarm messages in multiple languages, messages and trigger labels are printed in the language that they were originally logged in. (Messages are logged in the application language that is current at the time the message is triggered.) All alarm times and acknowledge times are printed in the current application language.
- Use the Print Alarm Status button. When the operator presses the button, a report of the status of alarms is printed. The report can include active alarms, past alarms, or all alarms that have been set up in the **Alarm Setup** editor. The report can include how many times each alarm was triggered, and the accumulated time in alarm.
  You can set up this button to print the alarm status for a selected alarm trigger, or for all alarms.
  If you set up alarm messages in multiple languages, all messages, trigger labels, and times are printed in the current application language.

For information about specifying which printer to use at runtime for applications that will run on a personal computer or a MobileView terminal, see Specify the printers to use at runtime on page 191.

For information about the type of printer to use with a PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, see Install printers on a PanelView Plus 7 or PanelView Plus 6 terminal on page 200 and Install printers on a PanelView Plus terminal on page 200.

Alarm log file

Alarm log file is also referred to as the alarm history. For example, the operator presses the clear alarm history button to remove all alarms from the log file.

If the alarm monitoring function is turned on, as soon as the application starts running, FactoryTalk View begins monitoring trigger connections for alarm conditions. When an alarm occurs, FactoryTalk View adds the associated message to the alarm log file. When the file is full, the oldest messages are deleted to make room for new messages. Specify the maximum number of messages to store in the **Alarm Setup** editor.

For each alarm, the log file records the time the alarm was triggered and the time the alarm was acknowledged (if it was acknowledged). From FactoryTalk View ME version 10.0, the time in a log file is Coordinated Universal Time (UTC). In FactoryTalk View ME version 9.0 or earlier, the time in a log file is local time.

The log file is saved every 30 seconds, if there are new alarms. The log file is also saved when the application is shut down.

The log file is retained when you restart an application after a shutdown or power loss.

If you download a newer version of an application to the runtime computer, the log file for the older version is deleted automatically.

You can delete the log file from the runtime computer at application startup. For information about deleting the log file, see Delete log files on the runtime computer on page 196.

Alarm trigger data types

When you create alarm triggers, you must specify the type of data the trigger's tag or expression uses. The type of data affects how alarms are triggered.

- Value trigger on page 103
- Bit trigger on page 104
- Least Significant Bit trigger on page 105

Value trigger

To trigger alarms based on integer or floating-point values, use the Value trigger type. When the value of the trigger connection equals the trigger value of a message, an alarm is generated.

If you use an analog tag (an HMI tag or a data server tag) or an expression, you can use any nonzero integer or floating-point value to trigger an alarm. Floating-point values are rounded to the nearest integer. For information about how values are rounded, see How values are rounded on page 80.

Trigger values cannot be 0. Digital tags have two possible values, 0 and 1. Therefore, if you use a digital tag (an HMI tag or a data server tag), you can only use the value 1 to trigger a message. If you want to use a digital tag to trigger two different messages, create an expression that adds 1 to the digital tag's value. That way, you can use the trigger values 1 and 2.

Example: Integer trigger values

This example shows how to set up alarms using integer trigger values to monitor the level of acid in a 100-gallon tank. Alarms are generated when the tank is 75 percent full and 90 percent full.

1. Create an alarm trigger for the tag Acid_tank_level. This tag points to an address in a programmable controller that is linked to a sensor in the tank. The tag's minimum value is 0 and its maximum value is 100. Use this expression for the alarm trigger:

   ```
   if ((Acid_tank_level > 74) AND (Acid_tank_level < 90))
   then 1
   else if (Acid_tank_level > 89)
   then 2
   else 0
   ```
2. Use the Value trigger type for the alarm trigger.
3. Specify these trigger values and alarm messages for the alarm trigger:

Trigger Value	Message
1	Warning! The acid tank has exceeded safe fill levels.
2	Danger! The level in the acid tank is too high.

At runtime, when the value of Acid_tank_level is 75 or greater, FactoryTalk View generates the first alarm message. The second message is generated when the tag value is 90 or greater.

Using the expression in step 1 allows a range of values to trigger each alarm. This is useful in case the tag is not being read at the exact moment it reaches a threshold level.

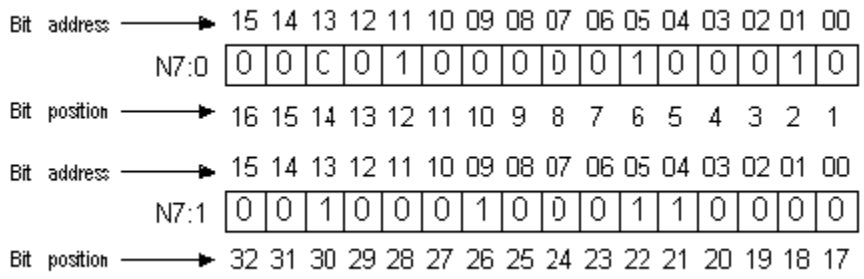
Bit trigger

Use the Bit trigger type to generate multiple simultaneous alarm messages. You can assign an array tag (consisting of up to 1,024 bits) to the trigger connection. Each bit in the array whose value changes from 0 to 1 triggers an alarm (if a message is set up for the bit position).

Example: Trigger values using bit positions

This example shows how to use bit positions to generate multiple simultaneous alarms. The array tag uses 32 bits, from N7:0/00 to N7:1/15.

This diagram shows the bit addresses and bit positions for the array tag:



The tag is currently generating alarms at bit positions 2, 6, 12, 21, 22, 26, and 30.

To set up alarms for this array tag

1. Create an alarm trigger for the array tag. Use the Bit trigger type.
   You must use a direct reference for the array tag. The tag uses the Integer data type (16 bits per element). Append L2 to the tag's starting address to indicate that the trigger will monitor 2 tag elements (32 bits), as follows:
   `{::[PLC5]N7:0,L2}`
   PLC5 is the FactoryTalk Linx device shortcut name. The two colons at the beginning are necessary when the tag reference contains a colon.
2. Specify trigger values and alarm messages for the alarm trigger. Each trigger value corresponds to a bit position (not a bit address).

Bit in the array (bit address)	Bit position	Trigger value	Message
00	1	1	Line 1: Conveyor has stopped.
01	2	2	Line 1: Power failure.
02	3	3	Line 2: Conveyor has stopped.
03	4	4	Line 2: Power failure.
..	..	..	..
31	32	32	Oven door open.

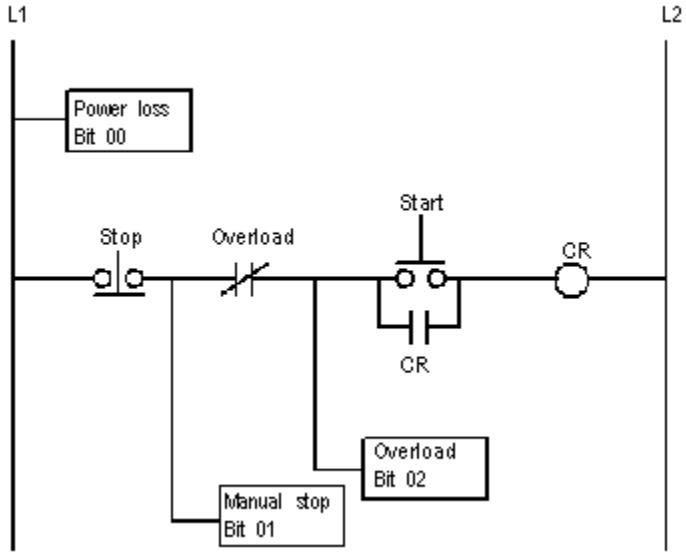
Least Significant Bit trigger

Use the Least Significant Bit (LSBit) trigger type to generate alarm messages based on a priority sequence that is determined by bit position. You can assign an array tag (consisting of up to 1,024 bits) to the trigger connection. When multiple bits in the array change from 0 to 1, only the alarm with the lowest bit position is generated.

Example: Trigger values using least significant bits

This example shows how to use the Least Significant Bit trigger type to prioritize which alarm is generated when multiple alarm conditions occur.

In this example, a programmable controller monitors a motor's power, manual stop, and overload status. The diagram indicates the location of sensors that are linked to an array tag in the programmable controller.



If the power to the motor is lost, the manual stop and overload switch also lose power, generating alarms for all three motor parts. The only alarm that the operator must see is the alarm for the motor, since the cause of the alarm is power loss to the motor, not a problem with the other two motor parts.

If someone stops the motor manually, power to the overload switch is lost. Alarms are generated for the manual stop and the overload switch, but the only alarm that the operator must see is the alarm for the manual stop.

1. Create an alarm trigger for the tag Motor_starter. The tag uses 16 bits, and points to the programmable controller address N7:61, although only the first three bit positions are used in this example. Use the LSBit (Least Significant Bit) trigger type.
   Because you are using only the first three bits in a single tag element, there is no need to specify a length for the array tag in this example. You can use a simple tag type.
   However, if you were also monitoring bits in N7:62, you would need to add L2 to the address, as follows:
   `{::[PLC5]N7:61,L2}`
2. Specify these trigger values and alarm messages for the alarm trigger:

Bit in the array (bit address)	Bit position	Trigger value	Message
00	1	1	The motor has lost power.

Bit in the array (bit address)	Bit position	Trigger value	Message
01	2	2	The motor has stopped.
02	3	3	The overload switch has lost power.

At runtime, if power to the motor is lost, the programmable controller changes all three bit values from 0 to 1, but FactoryTalk View generates the first alarm message only (since bit 00 is the lowest — that is, least significant — bit in alarm). If the operator acknowledges the first alarm and power is not yet restored, the second alarm is generated, and so on.

Tips for using array tags

To create an array tag, specify how many elements to use, beginning with the first element at the tag's address.

Use these tag data types for array tags. Use this table to determine how many data elements to address to provide the number of bits you need.

Data Type	Number of bits per element
Character	8
Byte	8
Short Integer	16
Word	16
Integer	32
Double Word	32

For example, if you want to monitor 128 bit positions for an alarm trigger, and you are using the Integer data type, you would append the modifier L4 to the tag name (because 4 x 32 = 128).

You cannot use HMI tags (digital or string tags) for array tags. You must use a direct reference (data server tags) for array tags.

Equivalent data types

Different processors and OPC servers use different terminology for tag data types. This section provides information about equivalent data types that you can use for array tags, with the associated number of bits for each.

Analog HMI tags created in FactoryTalk View

You can view a tag's data type in the Tag Browser.

Data type	Number of bits
Byte	8
Unsigned Integer	16
Integer	16
3-Digit BCD	16
4-Digit BCD	16

Data type	Number of bits
Long Integer	32

PLC and SLC tags

When creating array tags, you can use PLC and SLC tags with this data type:

Data type	Number of bits
Short Integer	16

Data server tags with the Short Integer data type are shown in the **Tag Browser** with the Item Canonical Data Type of Integer.

Studio 5000 tags

When creating array tags, you can use Studio 5000 tags with these data types:

Table Heading	Table Heading
SINT	8
INT	16
DINT	32

Data server tags with the data types described in the table are shown in the **Tag Browser** with the Item Canonical Data Types of Character, Short Integer, and Long Integer, respectively.

FactoryTalk Linx tag syntax

To create a FactoryTalk Linx array tag, use this syntax to specify the length of the array:

{tagname,Larraylength}

For example, {::[PLC5]N7:0,L5}

The tag N7:0 uses the Short Integer data type (16 bits), so the array tag consists of 80 bits (16 x 5).

When the tag name contains a colon (:), put two colons before the FactoryTalk Linx device shortcut name.

KEPServer Enterprise tag syntax

To create a KEPServer Enterprise array tag, use this syntax to specify the length of the array:

tagname [*arraylength*]

For example, Chan1.Dev1.Tag1 [3]

The tag Tag1 uses the Long Integer data type (32 bits), so the array tag consists of 96 bits (32 x 3).

Alternatively, you can define the tag length in KEPServer Enterprise when you create the tag, and then select the tag name in FactoryTalk View Studio. For more information, see KEPServer Enterprise Help.

Create alarm messages in multiple languages

FactoryTalk View 5.00 and later supports alarm messages in multiple languages. When you create alarm messages, they are in the current application language. You can export the alarm messages for translation and then import them back into the application. For details, see Set up language switching on page 160.

Optional alarm connections

When an operator acknowledges an alarm, the acknowledge time is recorded in the alarm log file and the alarm is silenced. This might be all you require of your alarm system.

However, you can use the optional alarm connections in the **Alarm Setup** editor to set up more sophisticated interactions with the data source when alarms are triggered, acknowledged, silenced, and reset. For example, you can set up the data source to acknowledge, silence, and reset alarms. You can also use optional alarm connections to close the alarm display or to allow the data source to close the alarm display.

There are two sets of optional alarm connections:

- Connections that work with a specific alarm trigger. You can assign a different set of connections to each alarm trigger.
- Connections that apply to all alarms

You can also assign connections to alarm lists, to allow the data source to determine which types of alarms to show in the lists. For more information, see How the alarm list graphic object works on page 119.

Connections that work with a specific alarm trigger

You can assign a different set of these connections to each alarm trigger, or to only one or some alarm triggers:

- **Handshake** — Assign a tag to this connection to notify the data source when the trigger's value changes.
- **Ack** — Assign a tag to this connection to notify the data source when the operator acknowledges an alarm (or all alarms).
- **Remote Ack** — Assign a tag or an expression to this connection to allow the data source to acknowledge alarms.
- **Remote Ack Handshake** — Assign a tag to this connection to notify the data source when a remote acknowledgment occurs.
- **Message** — Assign a tag to this connection to send alarm messages to the connection. Make sure that the tag supports the type of data sent in the alarm message. For example, if the message is a string of text, assign a string tag to the **Message** connection.
- **Message Notification** — Assign a tag to this connection to notify FactoryTalk View that an alarm message has been sent to the connection and FactoryTalk View must wait before sending a new message.
- **Message Handshake** — Assign a tag to this connection if you want the data source to notify FactoryTalk View when it has read the message.

Assign tags or expressions to these connections when you create your alarm triggers. For more detailed information about how these connections work, see the following sections.

How the Handshake connection works

The **Handshake** connection is useful for triggers that use the Value data type. Program the data source to queue alarm notifications when multiple alarms are generated for the same trigger. Use the **Handshake** connection to notify the data source that FactoryTalk View has detected the alarm notification. Then the data source can send the next alarm to FactoryTalk View.

When the application starts, the **Handshake** connection has a value of 1. When the trigger connection's value changes, the **Handshake** connection's value toggles from 1 to 0. The next time the trigger value changes, the **Handshake** connection's value toggles from 0 to 1.

A change in the trigger connection's value does not necessarily indicate an alarm. The value might still be within acceptable limits.

How the Ack connection works

When the application starts, this connection has a value of 0.

How the value changes

- When the operator acknowledges an alarm by pressing the acknowledge alarm button, FactoryTalk View sends the alarm's trigger value (or bit position, for bit triggers) to this connection, and holds the value as long as the operator presses the button, or for the duration of the hold time, whichever is longer. Then the connection's value is reset to 0.
- When the operator acknowledges all alarms, or the alarms for this trigger only, by pressing the acknowledge all alarms button, FactoryTalk View sends the trigger's **Acknowledge all** value to this connection, and holds the value as long as the operator presses the button, or for the duration of the hold time, whichever is longer. Then the connection's value is reset to 0.
- When the **Remote Ack connection** changes to a new non-zero value that matches the trigger value of an unacknowledged alarm, FactoryTalk View sends the trigger's value (or bit position, for bit triggers) to this connection, and holds the value for the duration of the hold time. Then the connection's value is reset to 0.

To avoid confusion about whether a specific trigger's alarm individual alarm has been acknowledged or all alarms for the trigger have been acknowledged, make sure the **Acknowledge all** value is not the same as one of the trigger values.

How the Remote Ack connection works

The Remote Ack connection allows for the remote acknowledgment of alarms. The data source writes the alarm's trigger value (or bit position, for bit triggers) to this connection, and the unacknowledged alarm with this trigger value is acknowledged.

If you assign this connection and specify an Acknowledge all value for the trigger, when the data source sends the Acknowledge all value to this connection, all unacknowledged alarms for the trigger are acknowledged.

Make sure that the Acknowledge all value is not the same as one of the trigger values.

Set up the data source to send the correct value to the Remote Ack connection when you want a remote acknowledgment to take place.

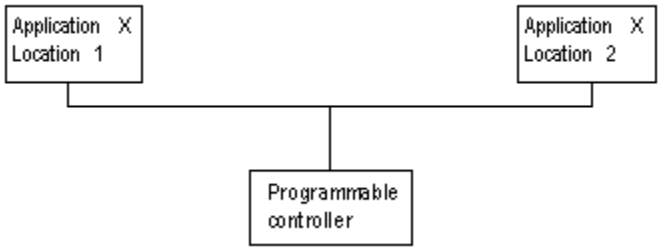
Remote acknowledgment does not take place if the hold time is in effect for the Ack connection.

Tip: An alarm's **Trigger** and **Remote Ack** settings should not be configured to use the same tag. Doing so may cause unpredictable alarm acknowledgment behaviors.

Example: Use the Remote Ack connection

This example shows how to use the Remote Ack connection to acknowledge an alarm.

In this example, the same application is run at two different locations on the plant floor, and both terminals are connected to the same programmable controller. All the tags in the example refer to addresses in the programmable controller.



1. Create an alarm trigger for a tag called Alarm. Use the Value trigger type.
2. Assign a tag called Ack to the Ack connection.
3. Assign a tag called Remote_ack to the Remote Ack connection.
4. Create an alarm message for the alarm trigger, with the trigger value 1.
5. Set up the data source to write the value at the Ack connection to the Remote Ack connection whenever the value of the Ack connection changes from 0 to a non-zero value.

At runtime, when an alarm with the trigger value 1 is generated, the alarm message appears at both locations simultaneously. If the operator acknowledges the alarm at Location 1, FactoryTalk View writes the value 1 to the Ack connection.

Since the Ack connection is write-only (not read), the acknowledgment does not appear at Location 2. However, the programmable controller writes the value from the Ack connection (1) to the Remote Ack connection, and the alarm at Location 2 is acknowledged.

How the Remote Ack Handshake connection works

This connection is used to notify the data source that the remote acknowledgment has been detected by FactoryTalk View.

When the application starts, the **Remote Ack Handshake** connection has a value of 1. When the **Remote Ack** connection value changes, the **Remote Ack Handshake** connection value toggles from 1 to 0. The next time the **Remote Ack** value changes, the **Remote Ack Handshake** value toggles from 0 to 1.

The **Remote Ack Handshake** value toggles whether the new **Remote Ack** value matches a trigger value.

Ensure that alarm messages are read by the data source before sending new messages

This section explains how the **Message**, **Message Notification**, and **Message Handshake** connections work.

To ensure that an alarm message is read by the data source before FactoryTalk View sends a new message, use message handshaking. While message handshaking is in effect for an alarm message, FactoryTalk View cannot send a new message to the alarm trigger's **Message** connection.

Methods of alarm message handshaking

Alarm handshaking works by setting the alarm trigger's **Message Notification** connection to 1. As long as the **Message Notification** connection is set to 1, new messages cannot be sent to the **Messages** connection. How the **Message Notification** connection is reset to 0 depends on how you set up alarm message handshaking.

There are two ways that you can use alarm message handshaking:

- To hold the message at the data source for a specific period of time.
- To hold the message at the data source until the data source notifies FactoryTalk View that the message has been read.

Choose the method that best suits your application needs and communication system.

Hold the message for a specific period of time

To set up an alarm trigger's alarm message handshaking so that the message at the **Message** connection is held for a specific period of time, assign a tag to the **Message Notification** connection and specify a hold time on the **Advanced** tab of the **Alarm Setup** editor.

This method of alarm message handshaking works as follows:

1. When an alarm occurs, the alarm message is sent to the **Message** connection.
2. The **Message Notification** connection is set to 1.
   As long as the **Message Notification** connection is set to 1, FactoryTalk View cannot send new messages to the data source.
3. The **hold time** timer begins timing.
4. When the hold time has expired, the **Message Notification** connection is reset to 0, and FactoryTalk View can send a new message to the **Message** connection.

Hold the message until the data source acknowledges that it has read the message

To set up an alarm trigger's alarm message handshaking so that the message at the **Message** connection is held until the data source notifies FactoryTalk View that it has read the message, use two connections: the **Message Notification** connection and the **Message Handshake** connection.

Set up the data source to send a new nonzero value to the **Message Handshake** connection when it has read the message at the **Message** connection.

If you use the **Message Handshake** connection, handshaking works like this:

1. When an alarm occurs, the alarm message is sent to the **Message** connection.
2. The **Message Notification** connection is set to 1.

As long as the **Message Notification** connection is set to 1, FactoryTalk View cannot send new messages to the data source.

3. When the data source has read the message, it sends a new nonzero value to the **Message Handshake** connection.
4. The **Message Notification** connection is reset to 0 and FactoryTalk View can send a new message to the **Message** connection.

How messages are queued

FactoryTalk View can queue up to 128 messages while it waits for the **Message Notification** connection to be reset to 0. Queued messages are sent to the data source on a first in, first out basis.

If the alarm message queue fills before the **Message Notification** connection is reset to 0, an error message is sent to FactoryTalk Diagnostics.

How the Message connection works

This connection is used to send the message associated with a trigger value to the data source.

The message is sent only if the alarm message's **Message to Tag** option is selected.

If you are going to generate text alarm messages in multiple languages, make sure the tag assigned to this connection can accommodate the string length of the messages in each language. For information about using multiple languages, see Set up language switching on page 159.

How the Message Notification connection works

This connection is used to notify FactoryTalk View that an alarm message has been sent to the data source and FactoryTalk View must wait to send another message.

When the application starts, the **Message Notification** connection is set to 0. When a message is written to the **Message** connection, the **Message Notification** connection value changes to 1. When the hold time expires, or the **Message Handshake** connection changes to a new nonzero value, the **Message Notification** connection value changes from 1 to 0.

Messages for an alarm trigger are queued if the **Message Notification** connection is set to 1.

How the Message Handshake connection works

Use the **Message Handshake** connection to reset the **Message Notification** connection when the data source has read the alarm message.

Set up the data source to send a new nonzero value to the **Message Handshake** connection when it has read the message at the **Message** connection.

Connections that apply to all alarms

These connections apply to all alarms:

- **Silence** — Assign a tag to this connection to notify the data source that all alarms have been silenced (for applications that will run on a personal computer only).
- **Remote Silence** — Assign a tag or an expression to this connection to allow the data source to silence all alarms (for applications that will run on a personal computer only).
- **Remote Ack All** — Assign a tag to this connection to allow the data source to acknowledge all alarms at once.
- **Status Reset** — Assign a tag to this connection to notify the data source that the alarm status has been reset.
- **Remote Status Reset** — Assign a tag or expression to this connection to allow the data source to reset the alarm status.
- **Close Display** — Assign a tag to this connection to notify the data source that the alarm display has closed.
- **Remote Close Display** — Assign a tag or expression to this connection to allow the data source to close the alarm display.

Assign tags or expressions to these connections on the **Advanced** tab of the **Alarm Setup** editor. For more detailed information about how these connections work, see the following sections.

How the Silence connection works

When the application starts, this connection has a value of 0. When alarms are silenced, this connection is set to 1 for the hold time and the internal beeper is turned off. At the end of the hold time, the connection is reset to 0.

If the silence event is triggered by a button press, this connection is set to 1 for the hold time or for the duration of the button press, whichever is longer. At the end of this time, the connection is reset to 0.

If subsequent alarms (that have been set up to sound the internal beeper) are generated, the alarms sound the beeper regardless of the Silence connection value or hold time status.

While the hold time is in effect, new alarms cannot be silenced.

How the Remote Silence connection works

When this connection changes to a new nonzero value, all alarms are silenced. Alarms can be silenced using other methods regardless of this connection's value.

Set up the data source to send a new nonzero value to the **Remote Silence** connection each time you want a remote silencing of alarms to take place.

How the Remote Ack All connection works

When this connection changes to a new nonzero value, all unacknowledged alarms are acknowledged.

Set up the data source to send a new nonzero value to the Remote Ack All connection each time you want a remote acknowledgment of all alarms to take place.

How the Status Reset connection works

When the application starts, this connection has a value of 0. If a reset alarm status or clear alarm history button is pressed, this connection is set to 1 for the hold time or for the duration of the button press, whichever is longer. At the end of this time, the connection is reset to 0.

The **Status Reset** connection is also set to 1 if the **Remote Status Reset** connection changes to a new nonzero value, and remains set for the hold time.

While the hold time is in effect, the alarm status cannot be reset again. The status of alarms keeps updating during the hold time.

How the Remote Status Reset connection works

When this connection changes to a new nonzero value, the status of all alarms is reset, as described in Methods for resetting alarms on page 101. Alarms can be reset using other methods regardless of this connection's value.

Set up the data source to send a new nonzero value to the Remote Status Reset connection each time you want a remote resetting of alarm status to take place.

How the Close Display connection works

When the application starts, this connection has a value of 0. If the alarm display (assigned in the **Alarm Setup** editor) closes, this connection is set to 1 for the hold time. At the end of the hold time, the connection is reset to 0.

If subsequent alarms are generated, the alarm display opens again regardless of the **Close Display** connection value or hold time status.

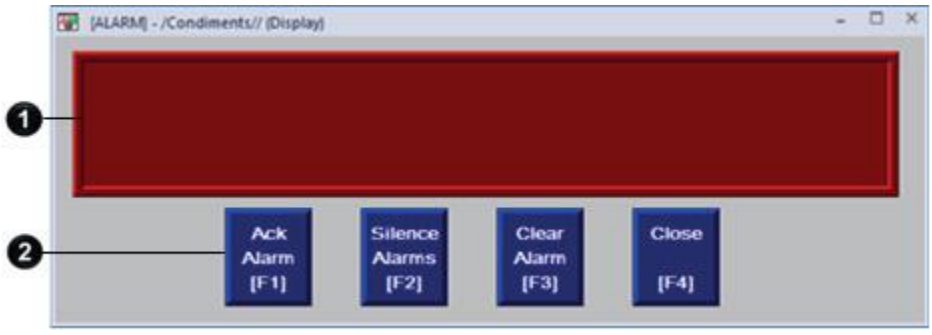
How the Remote Close Display connection works

When this connection changes to a new nonzero value, the alarm display closes. The alarm display can be closed using other methods regardless of this connection's value.

Set up the data source to send a new nonzero value to the **Remote Close Display** connection each time you want a remote closing of the alarm display to take place.

The [ALARM] display

When you create an application, it comes with a graphic display called [ALARM]. The [ALARM] display is the default display for showing alarm messages at runtime.



Item	Name	Description
1	Alarm banner graphic object	The alarm banner object on the [ALARM] display is set up to show new alarms as they occur and to show active alarms only. It shows one alarm at a time. You can also set up the alarm banner to queue new alarms until the shown alarm has been acknowledged. For more information about the alarm banner, see How the alarm banner graphic object works on page 120.
2	Buttons	Use these buttons to acknowledge, silence, and clear alarms, and close the display. For information about how the buttons work, see Use buttons with the alarm history and alarm objects on page 122.

You can:

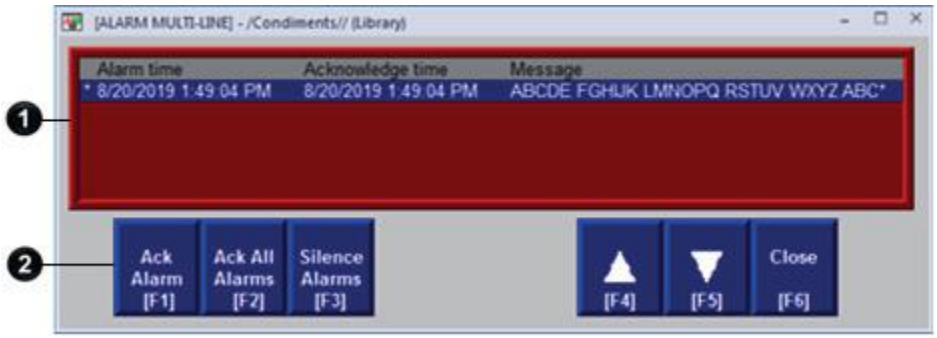
- Use the [ALARM] display as is or modify the display. For example, you can change the color of the objects, or add and remove buttons.
- Create your own graphic display as described in Create your own alarm display on page 118.
- Use one of the alarm displays in the graphic library. A copy of the [ALARM] display is included in the graphic library, with the name [ALARM BANNER].

The [ALARM BANNER] display

The Libraries folder contains a graphic display called [ALARM BANNER]. For more information about this display, see The [ALARM] display on page 114.

The [ALARM MULTI-LINE] display

The Libraries folder contains a graphic display called [ALARM MULTI-LINE]. The [ALARM MULTI-LINE] display allows the operator to see multiple alarms at once.



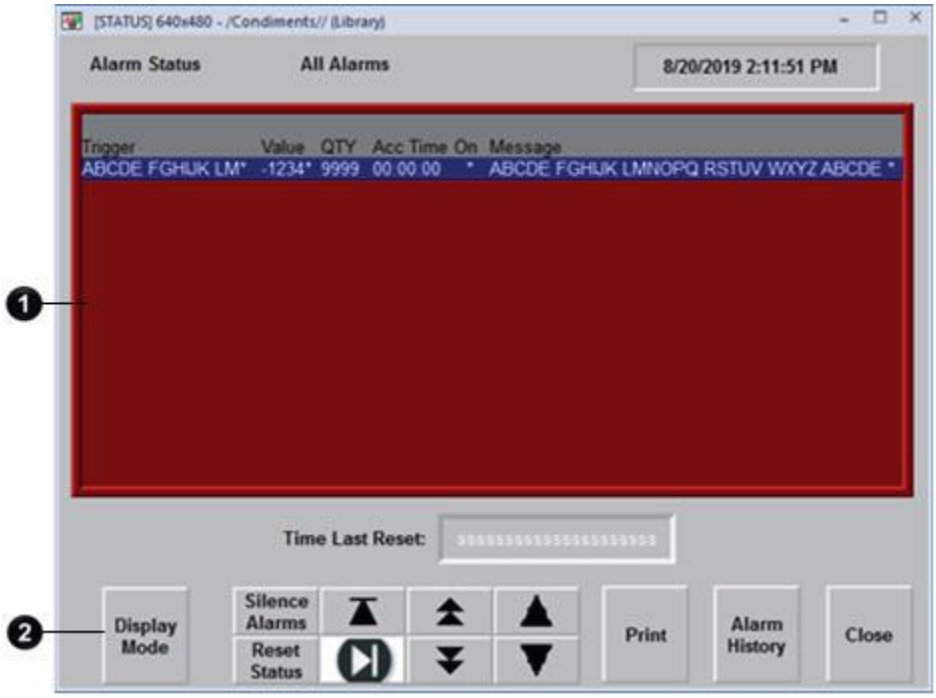
Item	Name	Description
1	Alarm list graphic object	The alarm list graphic object on the [ALARM MULTI-LINE] display shows the time when the alarms occurred and were acknowledged, and the alarm messages. It is set up to show all alarms. You can also set up the alarm list to show any combination of active, inactive, acknowledged, and unacknowledged alarms. For more information about the alarm list, see How the alarm list graphic object works on page 119.

Item	Name	Description
2	Buttons	Use these buttons to acknowledge and silence alarms, scroll the list, and close the display. For information about how the buttons work, see Use buttons with the alarm history and alarm objects on page 122.

Use the [ALARM MULTI-LINE] display as is or modify the display. For example, you can select which alarm triggers' alarms to show in the list.

The [STATUS] display

The Libraries folder contains a graphic display called [STATUS]. The [STATUS] display allows the operator to see the status of all the alarms that have been set up in the **Alarm Setup** editor. The [STATUS] display comes in two sizes, 640x480 and 800x600.

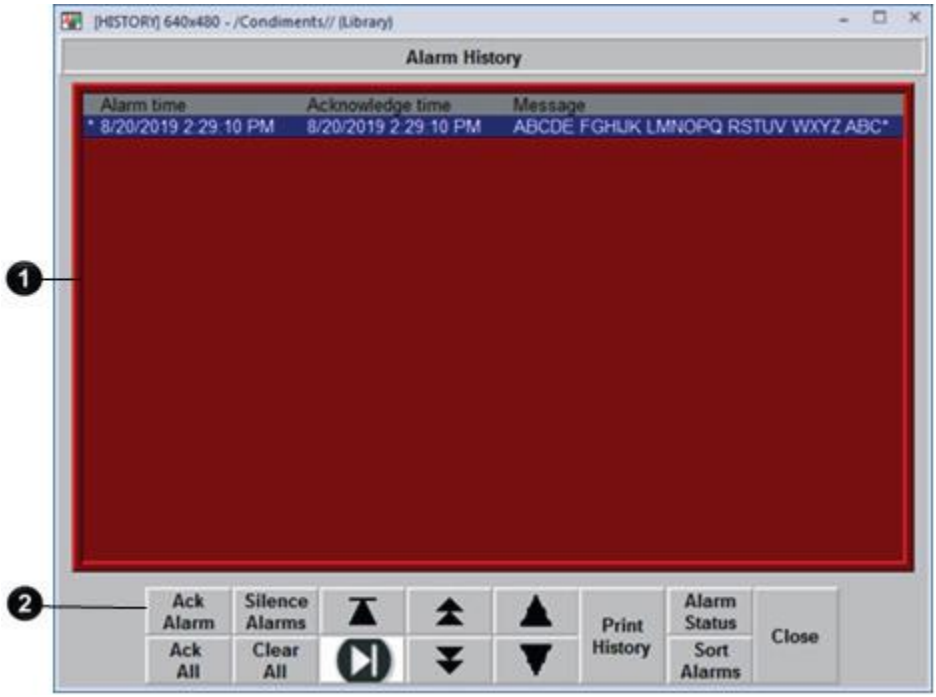


Item	Name	Description
1	Alarm status list graphic object	The alarm status list graphic object on the [STATUS] display shows the alarm messages. It is set up to show the status of active alarms. You can also set up the alarm status list to show the status of all alarms that have been set up, active alarms only or all alarms that have occurred since the alarm status was last reset. For more information about the alarm status list, see How the alarm status list graphic object works on page 121.
2	Buttons	Use these buttons to reset and silence alarms, change the display mode, print the status of alarms, open the [HISTORY] display, scroll the list, and close the display. For information about how the buttons work, see Use buttons with the alarm history and alarm objects on page 122.

Use the [STATUS] display as is or modify the display. For example, you can select which alarm triggers' alarms to show in the list, or remove buttons that you don't want to use.

The [HISTORY] display

The Libraries folder contains a graphic display called [HISTORY]. It is similar to the [ALARM MULTI-LINE] display, but the alarm list graphic object is larger (to show more alarm messages at a time), and the [HISTORY] display contains more buttons. The [HISTORY] display comes in two sizes, 640x480 and 800x600.



Item	Name	Description
1	Alarm list graphic object	The alarm list graphic object on the [HISTORY] display shows the time when the alarms occurred and were acknowledged, and the alarm messages. It is set up to show all alarms. You can also set up the alarm list to show any combination of active, inactive, acknowledged, and unacknowledged alarms. For more information about the alarm list, see How the alarm list graphic object works on page 119.
2	Buttons	Use these buttons to acknowledge, silence, and clear alarms, sort the list of alarms, print the alarm history list, open the [STATUS] display, scroll the list, and close the display. For information about how the buttons work, see Use buttons with the alarm history and alarm objects on page 122.

Use the [HISTORY] display as is or modify the display. For example, you can select which alarm triggers' alarms to show or remove buttons that you don't want to use.

Use displays from the library in your application

The previous four sections described the [ALARM BANNER], [ALARM MULTI-LINE], [STATUS], and [HISTORY] displays that are included in the Libraries folder. You can copy objects from the displays to your own display, or you can copy entire displays into your application.

For information about copying objects from a library into your graphic display, see Copy objects on page 307.

The graphic libraries are available on the development computer, but do not appear at runtime. To use a display in the library as a graphic display at runtime, you must add the library display into your application's folder of graphic displays. For details, see Use libraries as displays in your application on page 263.

Example: Use the [ALARM] and [STATUS] displays together

To use the [ALARM] display to alert the operator about alarms as they occur, and the [STATUS] display to show how many times and for how long alarms have been active:

1. In the **Alarm Setup** editor, on the **Advanced** tab, select the [ALARM] display as the display to open when alarms are triggered, if it is not already selected. (It is selected by default.)
2. Add the [STATUS] display from the library.
3. Edit the [ALARM] display by adding a Goto display button that opens the [STATUS] display when pressed.

Create your own alarm display

You can create your own graphic display for displaying and responding to alarms. For example, you could create a display with a static text object that notifies the operator that an alarm has occurred, and a Goto display button that the operator can press to open a display that lists the actual alarm message.

If you create your own graphic display, use an On Top display and select the **Cannot Be Replaced** option.

For information about the graphic objects you can use to show alarm information, see the information starting on How the alarm list graphic object works on page 119. For information about creating graphic displays and graphic objects, see Use graphic displays on page 252 and Use graphic objects on page 274.

Open the alarm display

The alarm display you specify on the **Advanced** tab of the **Alarm Setup** editor (the default [ALARM] display, a display you copy from the library, or your own display) is automatically opened whenever a trigger connection's value matches a trigger value (if you selected the **Display** option for the trigger value).

You can also create a Goto display button that the operator can press to open the alarm display. For information about setting up a Goto display button and specifying the display to open, see Help.

Close the display

The operator can close the display by pressing a **Close Display** button.

The display closes automatically under these circumstances:

- When the operator presses an acknowledge all alarms button and causes the newest alarm in the alarm history to be acknowledged
- When the operator presses a clear alarm history button and causes the newest alarm in the alarm history to be deleted
- When the **Remote Close Display** connection changes to a new nonzero value
- When the **Remote Ack All** connection changes to a new nonzero value
- When the newest alarm is acknowledged

If you don't want the display to close when the newest alarm is acknowledged, you can turn off this option in FactoryTalk View ME Station.

To change how the display closes

1. In FactoryTalk View ME, click **Terminal Settings**.
2. Double-click **Alarms**.
3. Specify how you want the display to close.

How the alarm list graphic object works

The alarm list graphic object shows the time an alarm was triggered, and the time it was acknowledged (if you set up the object to show acknowledged alarms).

At runtime, when a trigger connection at the data source matches a message's trigger value, an alarm appears in the alarm list. The alarm list can be in the [ALARM MULTI-LINE] or [HISTORY] display, in an alarm display you have created, or can be placed on any display in your application.

You can use multiple alarm lists, in the same display or in different displays. Each alarm list displays information from the same alarm log file, although you can set up different lists to display different information.

For information about setting up alarm lists, see Help.

What is shown?

- If the list is set up to show the alarms for specific alarm triggers, only alarms for those triggers are shown.
- If the list is set up to show unacknowledged alarms only, the alarms shown in the list are unacknowledged and the acknowledge time column, if any, is always blank.
- If the list is set up to show both acknowledged and unacknowledged alarms, acknowledged alarms can show a symbol (which you can specify) at the left end of the row, and the acknowledge time appears in the acknowledge time column, if shown.
- If the list is set up to show both active and inactive alarms, active alarms can show a symbol (which you can specify) at the left end of the row. If both the acknowledge symbol and active symbol are shown, the active symbol column is to the right of the acknowledge symbol column.
- If the message is too long to fit in the object, the last shown character is replaced with an asterisk. Similarly, if a column heading is too long to fit in the column, the last shown character is replaced with an asterisk. You can specify how many lines to allow for each alarm message, from 1 to 10.

- The number of alarm messages that fits on the display depends on the height of the alarm list, the list's font size, how many lines you have specified for each alarm message, and whether the column headings are shown.
- If alarms are currently being sorted by time, alarms are listed with the newest alarms on top.
- If alarms are currently being sorted by trigger value, alarms from the first trigger in the **Alarm Setup** editor are listed first, with alarms from subsequent triggers listed after. Alarms in the list from the same trigger are grouped and then sorted by trigger value, from lowest to highest.
- If there are multiple instances of the same alarm, these alarms are sorted by time with the newest on top.
- If you assign a tag or expression to any of the ActiveAcknowledged, ActiveUnacknowledged, InactiveAcknowledged, or InactiveUnacknowledged connections, when a connection's value is a nonzero value, the related type of alarm is shown in the alarm list. The connection value overrides the setting for the list in the **Alarm List Properties** dialog box.
- If you set up alarm messages in multiple languages, the messages are shown in the language that they were originally logged in. All alarm and acknowledgment times are shown in the current application language.

How the list scrolls

- When a display containing an alarm list is opened, the highlight bar or cursor is at the top of the list: on the most recent alarm if sorted by time; or on the first alarm for the first trigger, if sorted by trigger value.
- If alarms are sorted by time, and the top alarm is selected, as new alarms occur, the old alarms scroll down, but the highlight bar or cursor remains at the top.
- If alarms are sorted by time, when the operator selects an alarm other than the top one in the list (by using the move down button), and new alarms occur, they are added above the visible area if the list is full. The currently selected alarm continues to be selected and maintains its position in the shown portion of the list.
- If alarms are sorted by trigger value, as new alarms occur the currently selected alarm continues to be selected and maintains its position in the list.
- If the sort order changes, the selected alarm continues to be selected, but it might be displayed in a different position in the list.

How the alarm banner graphic object works

The alarm banner graphic object shows a single, unacknowledged alarm.

At runtime, when a trigger connection at the data source matches a message's trigger value, an alarm appears in the alarm banner. The alarm banner can be in the [ALARM] display or [ALARM BANNER] display, in an alarm display you have created, or can be placed on any display in your application.

You can use multiple alarm banners, in the same display or in different displays. You can set up different banners so that one displays the most recent alarm, and the other queues alarms until the displayed alarm is acknowledged.

For information about setting up alarm banners, see Help.

What is shown?

- If you set up the banner to show the alarms for specific alarm triggers, only alarms for those triggers are shown.
- You can set up the banner to queue new alarms until the operator clears the current alarm, or to always show the most recent alarm.

- When a display containing an alarm banner is first opened, the alarm banner is blank, unless the display is the alarm display specified in the **Alarm Setup** editor and it was opened in response to an alarm.
- When the shown alarm is acknowledged, it is removed from the alarm banner. If there are no newer alarms, the banner is blank.
- If the operator presses a clear alarm banner button, the banner is blank.
- If the operator presses a clear alarm history button or if all alarms are acknowledged, the banner is blank.
- If the banner is set up to show both active and inactive alarms, when an alarm is active, an asterisk (*) appears beside the message.
- If you set up alarm messages in multiple languages, messages and trigger labels are shown in the language that they were originally logged in. All alarm times are shown in the current application language.

How the alarm status list graphic object works

The alarm status list graphic object shows the status of alarms, including whether an alarm has been triggered, how many times an alarm has been triggered, and for how long.

The alarm status list can be on the [STATUS] display, on an alarm display you have created, or can be placed on any display in your application.

You can use multiple alarm status lists, on the same display or on different displays. You can set up different lists to show different information.

For information about setting up alarm status lists, see Help.

What is shown?

- If you set up the list to show the alarms for specific alarm triggers, only alarms for those triggers are shown.
- If an alarm is active, an asterisk (*) appears in the alarm state column, if shown.
- If the message is too long to fit in the object, the last character that would fit is replaced with an asterisk. Similarly, if a column heading is too long to fit in the column, the last character that would fit is replaced with an asterisk. You can specify how many lines to allow for each alarm message, from 1 to 10.
- The number of alarm messages that fits on the display depends on the height of the alarm status list, the list's font size, how many lines you have specified for each alarm message, and whether the column headings are shown.
- For value-triggered alarms, the accumulated time column shows how long the alarm has been set to the trigger value.
- For bit-triggered alarms, the accumulated time column shows how long the bit has been set to 1.
- For LSBit-triggered alarms, the accumulated time column shows how long the corresponding alarm's least significant bit has been set to 1.
- If you set up alarm messages in multiple languages, all messages and trigger labels are shown in the current application language, regardless of what language they were originally logged in.

What happens when the display is opened

- The first time a display containing the alarm status list is opened, the first page of alarms is shown.
- If the list is set up to show active alarms only, and alarms that were visible when the display was closed are no longer active, the alarms are removed from the list.
- If alarms have been reset since the display was last opened, the first page of alarms is shown.

Use buttons with the alarm history and alarm objects

Use Alarm buttons on page 122 and Key buttons on page 123 with the alarm history and alarm objects.

You can link the acknowledge alarm button to a specific alarm list or alarm banner, or set up the button to work with whichever alarm list or alarm banner is selected on the graphic display.

Similarly, you can link the clear alarm banner button to a specific alarm banner. This option is useful if you have multiple alarm banners on the same graphic display. And you can link the alarm status mode button to a specific alarm status list.

For more information about linking buttons to objects, see Link buttons to objects on page 332.

Alarm buttons

Use these alarm buttons to interact with the alarm list, alarm banner, or alarm status list, and to clear and sort the alarm history:

Alarm buttons	Descriptions
Acknowledge Alarm	This button acknowledges and silences the selected alarm in an alarm list or alarm banner.
Acknowledge All Alarms	This button acknowledges and silences all unacknowledged alarms or the unacknowledged alarms for a specific alarm trigger in the alarm history.
Alarm Status Mode	This button changes the mode of an alarm status list to its next setting. The order of the settings is: 1. All alarms 2. Active alarms 3. Past alarms The current setting is dependent on how the list is set up on the **Alarm** tab of the list's **Properties** dialog box.
Clear Alarm Banner	This button removes the current alarm from the banner without removing it from the alarm history and alarm lists.
Clear Alarm History	This button removes all alarms or the alarms for a specific alarm trigger from the alarm history and alarm lists, and resets the alarm status. If you choose to reset alarms, pressing this button resets the number of times an alarm has been triggered to 0, and the accumulated time in alarm to 0, for all alarms that are being cleared. If the alarm is still active, the number of times in alarm changes to 1 after the alarm is reset to 0.
Print Alarm History	This button prints a report of the alarm messages in the alarm log file. You can include all alarms, or the alarms for a specific alarm trigger. The report can include the time when alarms occurred and were acknowledged.
Print Alarm Status	This button prints a report of the status of all alarms or the alarms for a specific alarm trigger. The report can include how many times each alarm was triggered and the accumulated time in alarm.
Reset Alarm Status	This button resets the accumulated time and the number of times the alarm has been triggered to zero for all alarms in the application. If the alarm is still active, the number changes to 1.

Alarm buttons	Descriptions
Silence Alarms	This button silences all current alarms. If the hold time for any alarm is still in effect, the alarm cannot be silenced.
Sort Alarms	This button sorts the alarms in the alarm history and alarm lists by time or by trigger. When alarms are sorted by trigger, they are first sorted so that all alarms assigned to one trigger are together in the order that the triggers appear in the **Alarm Setup** editor. Within these groups, triggers are sorted by trigger value.

Key buttons

You can also use these key button graphic objects to interact with the alarm list, alarm status list, or alarm banner:

- **Move up**
- **Move down**
- **Page up**
- **Page down**
- **Home**
- **End**
- **Enter**

You can link key buttons to a specific alarm object, or set up the buttons to work with whichever object is selected on the graphic display. For more information, see Link buttons to objects on page 332.

For more information about key buttons, see Work with lists, trends, alarm banners, and numeric input objects on page 278.

For information about creating graphic objects, see Use graphic objects on page 275.

For information about setting up specific buttons, see Help.

Use alarm buttons to acknowledge, silence, clear, and delete alarms

Acknowledging messages does not remove them from the alarm log file. You can still show acknowledged alarms in an alarm list that is set up to show acknowledged alarms.

Acknowledge the selected alarm

When the operator presses the acknowledge alarm button, an enter button, or the **Enter** key on an external keyboard or keypad:

- The selected alarm in the alarm list is acknowledged and the audio indicator for the alarm, if any, is silenced.
- The shown alarm in the alarm banner is acknowledged and the audio indicator for the alarm, if any, is silenced.

This is what happens when the alarm is acknowledged:

- If assigned, the alarm trigger's **Ack connection** is set to the alarm trigger value at the data source. The value is held as long as the operator presses the button, or for the hold time, whichever is longer.
  If the hold time is still in effect and the operator acknowledges a new instance of the same alarm, the new acknowledgment is ignored.

- If the alarm list is set up to show unacknowledged alarms only, the alarm is removed from the list. The highlight bar or cursor moves up to the next unacknowledged alarm, unless it is already on the top unacknowledged alarm.
- If the highlight bar or cursor is on an acknowledged alarm when the operator presses the acknowledge alarm button, the button press is ignored.
- The alarm is removed from the alarm banner.

Acknowledge all alarms

When the operator presses an acknowledge all alarms button, all unacknowledged alarms in the system (or, optionally, for a specific alarm trigger), are acknowledged.

This is what happens when the alarms are acknowledged:

- For each alarm trigger, to which an **Acknowledge all** value is assigned, the specified **Acknowledge all** value is sent to the trigger's **Ack connection**. The value is held as long as the operator presses the button, or for the hold time, whichever is longer. Then the connection's value is reset to 0.
  If the **Acknowledge all** value is not assigned for a trigger, no value is sent to the trigger's **Ack connection**.
  If the hold time for any alarm is still in effect and the operator presses acknowledge all alarms, the button press is ignored.
- If the alarm list is set up to display unacknowledged alarms only, all acknowledged alarms are removed from the list.
- The alarm banner is cleared.

Silence alarms

When the operator presses the silence alarms button, any audible signal triggered by an alarm is silenced and the **Silence** connection is set to 1 for the hold time or for the duration of the button press, whichever is longer. Then the connection's value is reset to 0.

If the hold time for any alarm is still in effect and the operator presses silence alarms, the button press is ignored.

Clear and delete messages

You can clear and delete messages:

- From the alarm history
  When the operator presses the clear alarm history button, all alarms in the system (or, optionally, for a specific alarm trigger) are deleted from the alarm log file and from all alarm lists. For information about the alarm log file, see Alarm log file on page 102.
  If the **Ack** connection is assigned for any of the alarm triggers being cleared, the operator is prompted to acknowledge the unacknowledged messages before deleting them. If the operator chooses to acknowledge the alarms, the **Acknowledge all** value (if any) is sent to the **Ack** connection before deleting the messages. If the **Ack** connection is not assigned, or if the operator chooses not to acknowledge alarms, all messages are deleted immediately.
- From the alarm banner
  When the operator presses the clear alarm banner button, the shown alarm is cleared from the banner. The alarm remains in the alarm log file and in all alarm lists it appears in.

Use alarm buttons to sort alarms and reset alarm status

You can use the alarm buttons to sort alarms and reset alarm status.

Sort alarms

When the operator presses the sort alarms button, the sort order toggles from time to trigger value or from trigger value to time. The sort order applies to all alarms in all alarm lists and in the alarm log file.

Reset alarm status

When the operator presses the reset alarm status button, the alarm status of each alarm is reset as follows:

- The accumulated time an alarm has been in alarm is reset to 0. If the alarm is still active, the time begins accumulating again.
- The number of times the alarm has been triggered is reset to 0. If the alarm is still active, the number changes to 1.

If the Status Reset connection is assigned (in the **Alarm Setup** editor), it is set to 1 for the hold time or for the duration of the button press, whichever is longer. If a hold time from a previous reset alarm status button press is still in effect, the button press is ignored.

Pressing the clear alarm history button can also reset the status of alarms, if you select the **Reset alarm status** option for the button. You can set the button up to clear and reset the status of all alarms, or just the alarms for a specific alarm trigger.

Retain alarm status

The status of alarms is retained when the application is shut down. When the application is restarted:

- If an alarm is still active, the time begins accumulating again.
- If an alarm is still active, the number of times the alarm has been triggered increases by 1.

Change the alarm status shown in the alarm status list

When the operator presses the alarm status mode button, the type of alarms shown in the alarm status list changes as follows:

- If the current type shown is all alarms, the list changes to show active alarms only.
- If the current type shown is active alarms, the list changes to show past alarms only.
- If the current type shown is past alarms, the list changes to show all alarms.

The change affects the linked or selected alarm status list only, not all alarm status lists.

Set up FactoryTalk Diagnostics

This chapter describes:

- FactoryTalk Diagnostics
- How to show diagnostics messages during application development
- How to view FactoryTalk Diagnostics log files
- How to use the Diagnostics Setup tool
- How to show and print diagnostics messages at runtime
- The [DIAGNOSTICS] display
- How to create your own diagnostics display
- How to open and close the diagnostics display
- How the diagnostics list graphic object works
- FactoryTalk audit trail

About FactoryTalk Diagnostics

FactoryTalk Diagnostics records information about various types of system activity, including:

- Macro usage
- Operator comments
- System messages and errors
- Errors from the communication network
- Tag read and write activity

Browse diagnostics messages

You can view diagnostics messages about system activity during application development and at runtime. For example, while you develop your application, use diagnostics messages to track what the system is doing and to ensure you have set things up the way you intend. At runtime, use diagnostics messages to show the operator messages about system activity. You can also save information in a log file for future processing or analysis.

How to set up FactoryTalk Diagnostics

You must set up FactoryTalk Diagnostics on both the development computer and the runtime computer. The settings for FactoryTalk Diagnostics apply to all Rockwell Automation products installed on the computer.

Set up FactoryTalk Diagnostics involves:

- Set up destinations, where logged information is received for storage or display.
- Set up which destinations receive which categories of messages. This is called message routing.
- Set up how messages are shown and printed at runtime.

This chapter describes how to set up FactoryTalk Diagnostics on personal computers. For information about setting up FactoryTalk Diagnostics on the MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, and PanelView Plus CE terminals, see Help.

Destinations

FactoryTalk Diagnostics allows you to send diagnostics messages to multiple destinations.

During application development, you can send diagnostics messages to:

- The **Diagnostics List** at the bottom of the FactoryTalk View Studio window
- A local log, which is viewable using the FactoryTalk Diagnostics Viewer tool
- An ODBC database

At runtime, you can send diagnostics messages to:

- Diagnostics list graphic objects on the default [DIAGNOSTICS] display or on a display you create
- A printer
- CSV files exported when triggered
- A local log (available on personal computers only), which is viewable using the FactoryTalk Diagnostics Viewer
- A remote log (MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal only)
- An ODBC database

The FactoryTalk Diagnostics Viewer is available on personal computers only. However, if you send diagnostics messages from a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal to a personal computer, you can view the messages in the FactoryTalk Diagnostics Viewer.

Tip: The destinations available on your computer might vary, depending on which Rockwell Automation products you have installed.

At runtime, you can send audit messages to:

- Audit graphic objects on a display you create
- Audit log files
- CSV files exported when triggered

Message routing

You can decide which destinations receive which kinds of messages. This allows you to make the right information available to the right person, in the right place.

For example, you can:

- Send messages that contain information about what the system is doing to the local log file. You can also send warnings about things that might go wrong if left unattended to the local log file.
  This allows a control systems engineer to analyze system activity and performance, and make corrections during scheduled maintenance times.
- Send errors that require immediate action to the diagnostics list object on the [DIAGNOSTICS] display, as well as the log file.
- Set up the [DIAGNOSTICS] display to open automatically when system activity occurs, to allow the operator to respond to problems that might stop production if they are not solved immediately.
- Send audit messages to a log file and audit graphic objects.

Message categories

FactoryTalk Diagnostics categorizes messages by severity and by audience. For each destination, you can specify the message severity and audience.

If messages of a particular category are not routed to any destination, the messages for that category are not logged. For example, you might not want to log information messages, or you might not want to send any messages to the Operator.

Message severities

FactoryTalk View categorizes messages in four severities:

- Errors indicate that a process or action has failed. For example, a tag's value could not be written, or a user typed the wrong password while logging on to the system.
- Warnings indicate that a process or action might not function correctly, or might eventually fail if preventive action isn't taken. For example, if an ActiveX® control used on a graphic display is a different version than the one installed on the runtime computer, a warning is logged to indicate the mismatch. Mismatched ActiveX controls might not behave as expected at runtime.
- Information indicates that a process or action has completed successfully. For example, a user logged on to the system or a tag value has been written to the data source.
- Audit indicates that the system configuration has been changed. FactoryTalk View records the creation, modification, and deletion of components (such as graphic displays) as audit messages.
  Other Rockwell Automation products also use audit messages. For example, if you set up an audit log in FactoryTalk AssetCentre, FactoryTalk View's audit messages will be sent there, as well as to FactoryTalk Diagnostics.

In the FactoryTalk Diagnostics Viewer, the message severity is shown in the **Severity** column, indicated by the ! symbol.

The content of the messages is provided by the system. You do not need to do anything to set up the message content.

Audiences

FactoryTalk Diagnostics allows messages to be categorized differently for the various people for whom the messages are relevant. You can sort the messages in the FactoryTalk Diagnostics Viewer, so that the messages relevant to a particular person are grouped.

You can specify which types of messages are sent to the Operator, Engineer, and Developer audiences.

FactoryTalk View automatically sends audit messages to the Engineer and Secure audiences. Audit messages allow auditing tools, such as those required for US Government 21CFR Part 11 compliance, to track system activity.

Use the Diagnostics Setup tool on a PC

Use the Diagnostics Setup tool on a PC to set up:

- Destinations and message routing on the development computer, including logging to Diagnostics List, audit trail list, local log, or an ODBC database.
- Destinations and message routing on personal computers, including runtime logging to Diagnostics List, audit trail list, local log, or an ODBC database.
- The MobileView or Windows CE runtime destination to receive logged messages from.

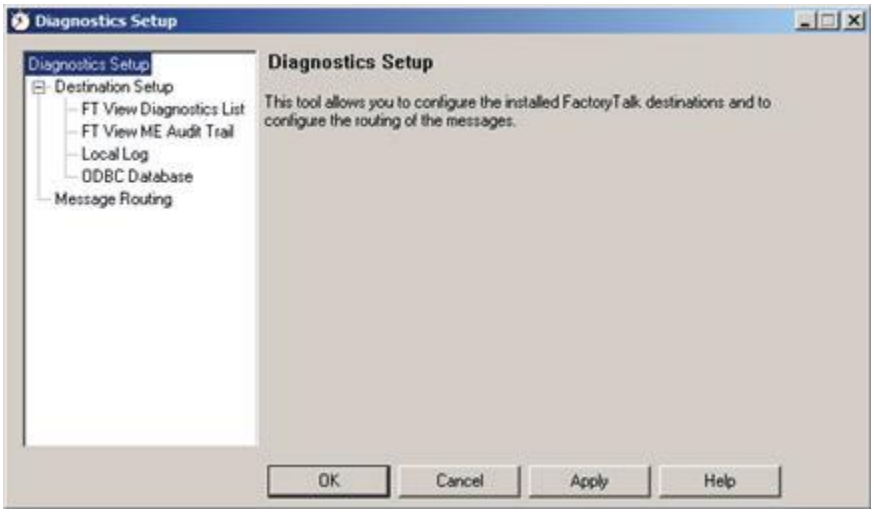
You can also use the Diagnostics Setup tool to clear the local log file.

Open the FactoryTalk Diagnostics Setup tool on a PC

Use the following steps to open the FactoryTalk Diagnostics Setup tool on a PC.

To open the FactoryTalk Diagnostics Setup tool, use one of these methods:

- In FactoryTalk View Studio, from the **Tools** menu, select **Diagnostics Setup**.
- In the **FactoryTalk View ME Station** dialog box, click **Terminal Settings** and then double-click **Diagnostics Setup**.



For information about opening the **FactoryTalk View ME Station** dialog box, see Start FactoryTalk View ME Station on the development computer on page 185.

For details about using the Diagnostics Setup tool, see Help.

Log to an ODBC database

The option of logging FactoryTalk Diagnostics messages to an ODBC database is available for personal computers only.

ODBC logging works by periodically exporting the contents of the local log file to an ODBC-compliant database. FactoryTalk View supports these ODBC-compliant databases:

- Microsoft Access®
- Sybase SQL Server

- Oracle®
- Microsoft SQL Server®

If you have set up FactoryTalk Diagnostics to overwrite events in the local log file, make sure that messages are logged to the ODBC-compliant database before the oldest events are deleted.

For information about the contents of the FactoryTalk Diagnostics ODBC tables, see ODBC database schema on page 463.

Message buffering

FactoryTalk Diagnostics messages are stored in the computer's local log file, and are exported to the ODBC-compliant database at the interval you specify.

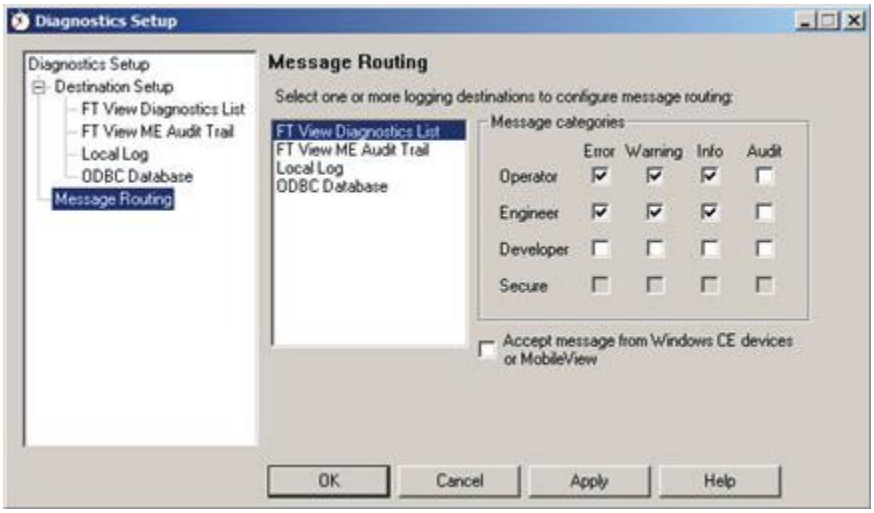
You can also specify that messages remain in the local log file for a period after they have been exported to the ODBC-compliant database. This is useful during a network failure, or any other reason that causes the database to be unavailable. In this case, the messages remain in the local log file until the buffer time expires. If the ODBC-compliant database becomes available during that time, the buffered messages are then exported to the database.

Route messages

For each destination you set up, you can specify how messages are routed.

To route messages, specify:

- The types of messages to log
- The audiences to send the messages to
- Whether to accept messages from MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal



Receive messages from a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal

You can set up FactoryTalk Diagnostics to receive messages that are generated on the MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal at runtime. The terminal must be on the same network as the computer running FactoryTalk Diagnostics.

To receive messages

- Enable the personal computer to receive the messages.
- Set up the MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal to send the messages.

For details about configuring the computer and the MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal for remote audits, see Help.

Diagnostics Setup on a MobileView or PanelView Plus family terminal

Use Diagnostics Setup on a MobileView or PanelView Plus family terminal to set up:

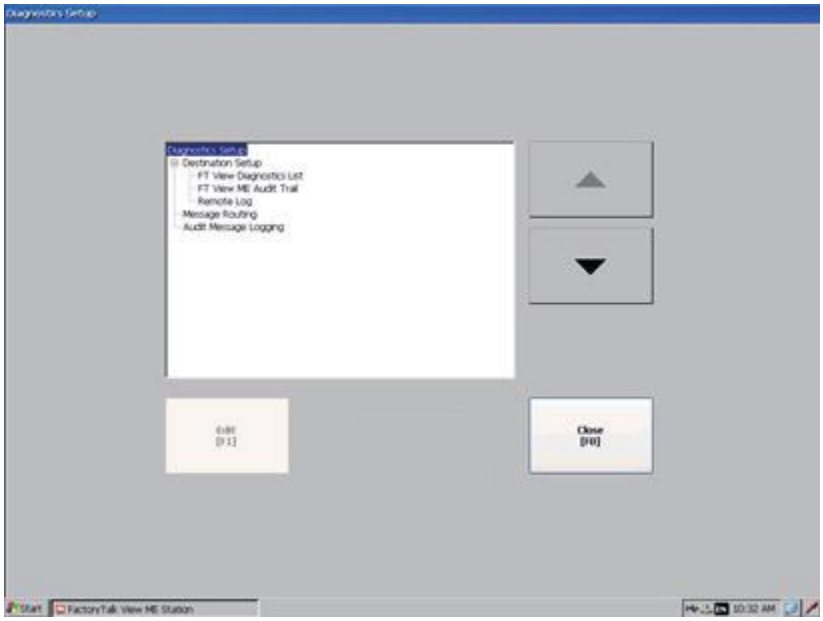
- Destinations for remote log
- Message routing, including logging to:
  - Diagnostic list
  - Audit trail
  - Remote log
- Audit message logging

Open Diagnostics Setup

Use the following steps to open **Diagnostics Setup**.

To open Diagnostics Setup

1. In the **FactoryTalk View ME Station** dialog box, click **Terminal Settings**
2. Select **Diagnostics Setup**.
3. Click ↵.



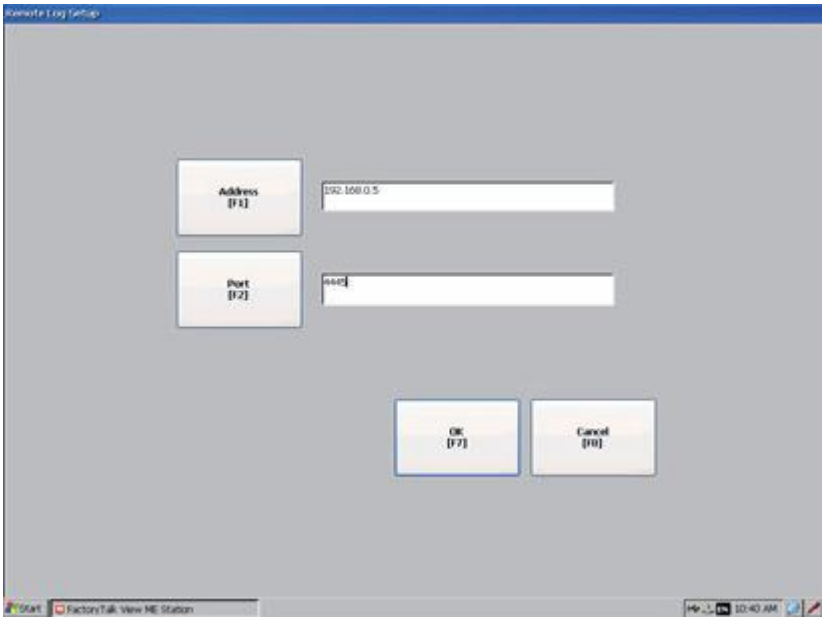
For information about opening the **FactoryTalk View ME Station** dialog box, see Start FactoryTalk View ME Station on a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal on page 201.

Set up remote log destination

You can specify the destination for remote log.

To set up remote log destination

1. In the **Diagnostics Setup** dialog box, select **Remote Log**.
2. Click **Edit**.



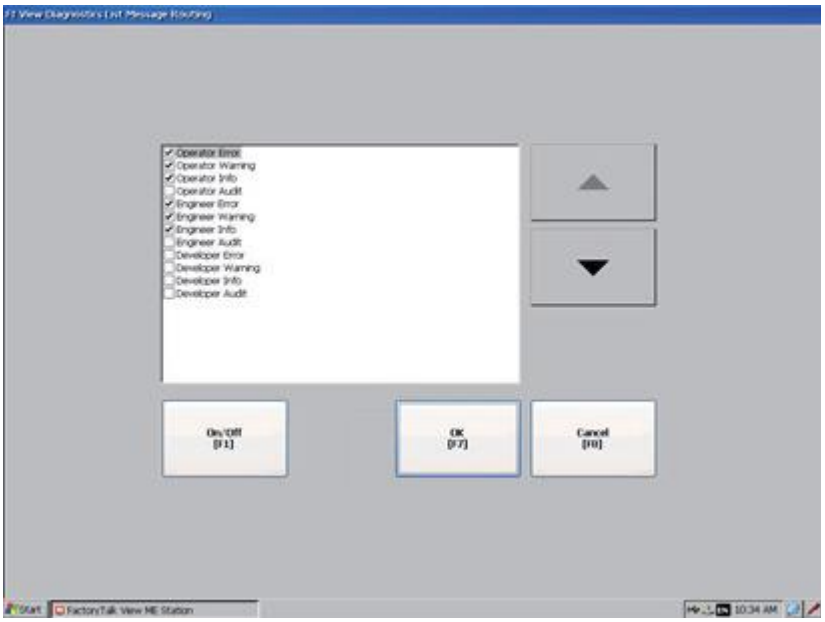
3. Enter the address and port of the remote computer.
4. Click **OK**.

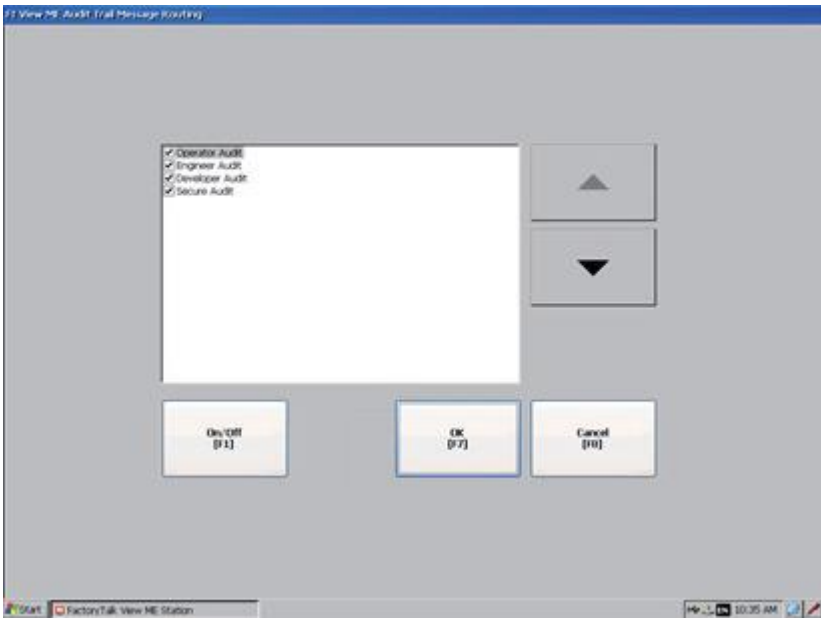
Route messages

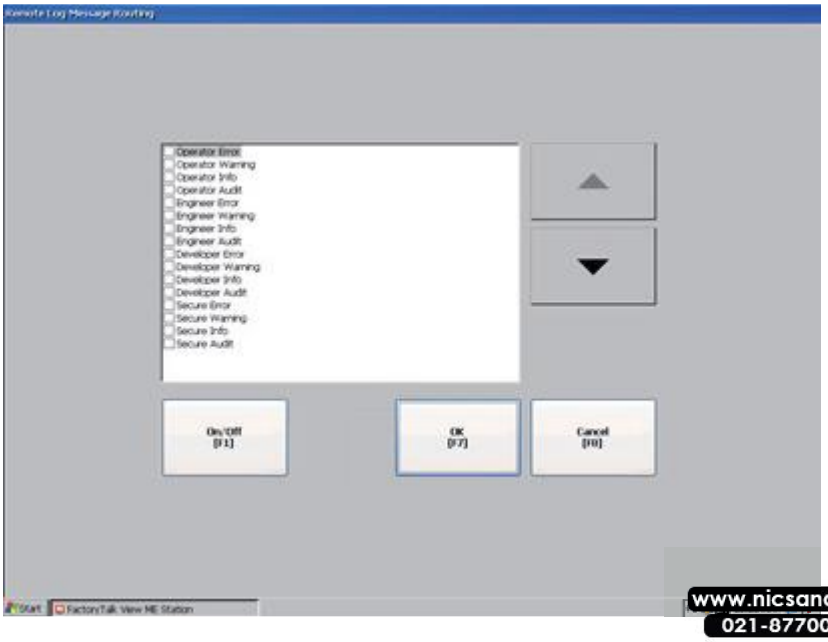
You can specify how messages are routed for each destination.

To route messages

1. In the **Diagnostics Setup** dialog box, select **Message Routing**.
2. Click **Edit**.
3. Select a destination:
   - **FT View Diagnostics List**
   - **FT View ME Audit Trail**
   - **Remote Log**
4. Click ↵.
5. Click **On/OFF** to select the checkboxes to specify:
   - The types of messages to log
   - The audiences to send the messages to







Specify audit message severity

You can specify whether to log audit messages as audit or information severity.

To specify audit message severity

1. In the **Diagnostics Setup** dialog box, select **Audit Message Logging**.
2. Click **Edit**.
3. Select:
   - **Audit only** to log audit messages as audit severity.
   - **Information only** to log audit messages as information severity.

For more information about message severities, see Message severities on page 128.

Show diagnostics messages during application development

During application development, diagnostics messages are shown as you create, modify, and delete components, and when you test your graphic displays. The messages are shown in the **Diagnostics List** at the bottom of the FactoryTalk View Studio window.

You can also view diagnostics messages in the FactoryTalk Diagnostics Viewer. See FactoryTalk Diagnostics log files on page 135.

You can also view audit messages in audit message log files and export audit messages in CSV format. See FactoryTalk audit log file on page 139.

By default, all warning and error messages are shown in the **Diagnostics List**. To change what is shown in the **Diagnostics List**, use the Diagnostics Setup tool. See Diagnostics Setup tool on page 129.

To show the Diagnostics List

- From the **View** menu, select **Diagnostics List**. A check mark beside the menu item indicates the **Diagnostics List** is shown.

If you don't want to show diagnostics messages

- From the **View** menu, select **Diagnostics List**. The option is turned off when no check mark appears.

For information about testing graphic displays, see Test your displays as you work on page 258.

FactoryTalk Diagnostics log files

Use the FactoryTalk Diagnostics Viewer tool to view the contents of FactoryTalk Diagnostics local log files. The FactoryTalk Diagnostics Viewer is available on personal computers.

You can also set up personal computers to accept messages from a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, and then show the messages in the FactoryTalk Diagnostics Viewer or in the **Diagnostics List**. For information about sending messages from a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal to personal computers, see Receive messages from a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal on page 130.

Open the FactoryTalk Diagnostics Viewer

Use the following steps to Open the **FactoryTalk Diagnostics Viewer**.

To open the FactoryTalk Diagnostics Viewer, use one of these methods:

- In FactoryTalk View Studio, from the **Tools** menu, select **Diagnostics Viewer**.
- From the Windows **Start** menu, select **All apps > Rockwell Software > Diagnostics Viewer**.

For information about setting up FactoryTalk Diagnostics to send messages to the local log, see Help for the Diagnostics Setup tool.

For additional information about setting up and using the FactoryTalk Diagnostics Viewer, see the FactoryTalk Diagnostics Viewer Help.

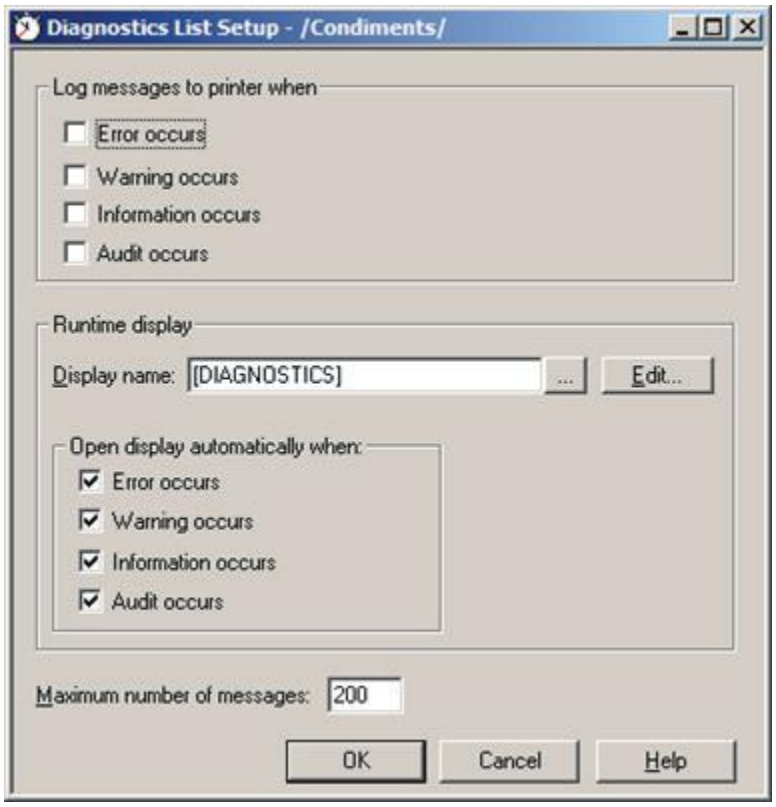
Show and print diagnostics messages at runtime

At runtime, you can show diagnostics messages on a graphic display. You can also print the messages.

Diagnostics List Setup editor

Use the **Diagnostics List Setup** editor to set up when to print and show messages at runtime in FactoryTalk View ME Station.

The **Diagnostics List Setup** editor is on the application tree under the **System** folder. Double-click **Diagnostics List Setup** to open the editor.



Tip: If you don't want to show diagnostics messages in FactoryTalk View ME Station, in the **Diagnostics List Setup** editor, clear the **Display name** box or clear the four **Open display automatically when** boxes.

For details about using the **Diagnostics List Setup** editor, see Help.

Printed messages list the date and time, the category of system activity, and the message text. By default, messages are not printed at runtime.

By default, your application is set up to show diagnostics messages automatically at runtime. If you want to use the default settings, you don't need to do anything further to set up diagnostics messages for print and display in FactoryTalk View ME Station.

The messages that are printed and shown depend on how you set up FactoryTalk Diagnostics on the runtime computer:

- For applications running on personal computers, use the Diagnostics Setup tool to set up diagnostics message logging. For information about using the Diagnostics Setup tool, see Help.
- For information about setting up FactoryTalk Diagnostics on the MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, see Help.

For applications running on personal computers, you can also view the runtime messages in the FactoryTalk Diagnostics Viewer as described on View FactoryTalk Diagnostics log files on page 135.

Set up how messages are shown and printed at runtime

Use the following steps to set up how messages are shown and printed at runtime.

To set up how diagnostics messages are shown at runtime

1. In FactoryTalk View Studio, in the **Diagnostics List Setup** editor, specify which messages to print, the graphic display to use, when to open the graphic display, and the maximum number of messages to store.
2. (optional) Modify the default [DIAGNOSTICS] display or create your own graphic display for diagnostics messages. For example, if you don't want the operator to clear all messages at once, edit the default display to remove the clear all button.
   For information about graphic displays, see Use graphic displays on page 252.

[DIAGNOSTICS] display

When you create an application, it comes with a graphic display called [DIAGNOSTICS]. The [DIAGNOSTICS] display is the default display for showing diagnostics messages at runtime.

You can use the [DIAGNOSTICS] display as is, or modify the display. For example, you can change the color of the objects, or add and remove buttons. Or, you can create your own graphic display to use for showing diagnostics messages. In the **Diagnostics List Setup** editor, specify the display to use.

Item	Description
1	Diagnostics list graphic object, which lists the diagnostics messages.
2	Diagnostics list buttons for clearing the messages, and key buttons for scrolling the list and closing the display.

You can create your own graphic display for showing diagnostics messages, containing a diagnostics list graphic object and the buttons the operator needs for using the display.

If you create your own graphic display, use an On Top display and select the **Cannot Be Replaced** option.

For more information about the diagnostics list, see How the diagnostics list graphic object works on page 138.

For information about the buttons you can use with the list, see Use buttons with the diagnostics list on page 138.

For information about creating graphic displays and graphic objects, see Use graphic displays on page 252 and Use graphic objects on page 274.

Open the diagnostics display

The diagnostics display you specify in the **Diagnostics List Setup** editor (either the default [DIAGNOSTICS] display or a display you create) is automatically opened whenever a diagnostics message of the specified severity occurs.

The display remains open until the operator closes the display.

If you decide not to open the diagnostics display automatically, you can create a Goto display button that the operator can press to open the diagnostics display. For information about setting up a Goto display button and specifying the display to open, see Help.

How the diagnostics list graphic object works

At runtime, when a severity of system activity occurs that you set up to log, a message about the activity is added to the diagnostics list. The diagnostics list can be on the [DIAGNOSTICS] display, on a diagnostics display you have created, or can be placed on any display in your application. You can use multiple diagnostics lists on the same display or on different displays. Each diagnostics list shows the same information.

If the message is too long to fit in the list, the last shown character is replaced with an asterisk (*).

The number of diagnostics messages that is visible at one time depends on the height of the diagnostics list and the list's font size.

Messages are listed with the newest diagnostics message on top.

For information about setting up diagnostics lists, see Help.

Use buttons with the diagnostics list

You can use these diagnostics buttons to interact with the diagnostics list:

This button	Does this
Diagnostics clear	Clears the selected message from all diagnostics lists in the application.
Diagnostics clear all	Clears all diagnostics messages from all diagnostics lists in the application.

You can assign any captions you choose to the labels on the diagnostics buttons.

You can link the diagnostics clear button to a specific diagnostics list. When the operator presses the button, the selected message is deleted from the linked diagnostics list, and from any other diagnostics lists in the application. For information about linking buttons to objects, see Link buttons to objects on page 332.

You can also use these key button graphic objects to interact with the diagnostics list:

- **Move up**
- **Move down**
- **Page up**
- **Page down**
- **Move left**
- **Move right**
- **Home**
- **End**

For more information about key buttons, see Work with lists, trends, alarm banners, and numeric input objects on page 278.

For more information about linking buttons to objects, see Link buttons to objects on page 332

For information about creating graphic objects, see Use graphic objects on page 275.

For information about setting up specific buttons, see Help.

FactoryTalk audit log file

FactoryTalk audit log file is also referred to as the audit history. If the audit trail function is turned on, when an audit occurs, FactoryTalk View adds the associated information to the audit log file. When the file is full, the oldest audit entries are deleted to make room for new entries. Specify the maximum number of audit entries to store in the **Audit Trail Setup** editor.

For each audit entry, the log file records the following:

- **Occurred time** - The UTC date and time at which the software product records the action as having occurred.
- **Source** - The name of the software product that logged the action in the audit log.
- **Location** - The name of the computer, Windows CE device, or MobileView terminal on which the action took place.
- **Resource** - *Reserved*
- **Username** - The login ID of the user who is using the software product. This can be a FactoryTalk Security user or a Microsoft Windows user, depending on how the user logged into the software product that generated the audit record.
- **Message** - Information about a specific operator action executed within the context of the running runtime application. For example, the message for a tag write will include the tag being modified, and the tag's previous and new values.

The log file is saved automatically.The log file is also saved when the application is shut down.

The log file is retained when you restart an application after a shutdown or power loss.

If you download a newer version of an application to the runtime computer, the log file for the older version is deleted automatically.

You can delete the log file from the runtime computer at application startup. For information about deleting the log file, see Delete log files on the runtime computer on page 196.

Tip:

The audit log file is automatically saved in:

- (PC and MobileView) **\Users\Public Documents\RSView Enterprise\ME\Logs\<ApplicationName>\M_AuditTrails**
- (PanelView Plus 7 and PanelView Plus 6) **\Windows\Logs\<ApplicationName>\M_AuditTrails**

Set up and turn on on-board audit trail during application development

Use the following steps to set up and turn on on-board audit trail during application development.

To set up and turn on on-board audit trail during application development

1. From the **Explore** tree, double-click **Startup** to open the **Startup** editor.
2. In the **Startup** editor, select the **Audit trail** checkbox and click **OK**.
3. From the **Explore** tree, double-click **Audit Trail Setup** to open the **Audit Trail Setup** editor.
4. Specify the **Maximum number of audit messages** for the log file.
5. (Optional) Specify these values and assign tags:
   - **Capacity high warning (%)**
   - **Capacity high-high warning (%)**
6. (Optional) Specify a tag to **Capacity overrun.**
7. (Optional) Specify a tag or expression to **Remote clear audit history**.
8. Click **OK** to save the options and close the editor.

Audit trail display

You can create a graphic display for showing audit messages, containing an audit trail list and an audit trail detail graphic objects. Use the **Clear Audit Trail** button to clear all audit messages.

For more information about the audit graphic objects, see How the audit graphic objects work on page 140.

For information about the buttons you can use with the audit graphic objects, see Use buttons with the audit trail list on page 141.

For information about creating graphic displays and graphic objects, see Use graphic displays on page 252 and Use graphic objects on page 275.

How the audit graphic objects work

Audit trail list shows the information about operator actions at runtime, including:

- **Occurred time** - The local date and time when the software product records the action as having occurred.
- **Source** - The name of the software product that logged the action in the audit log.
- **Location** - The name of the computer, Windows CE device, or MobileView terminal on which the action took place.
- **Resource** - *Reserved*
- **Username** - The login ID of the user who is using the software product. This can be a FactoryTalk Security user or a Microsoft Windows user, depending on how the user logged into the software product that generated the audit record.
- **Message** - Information about a specific operator action executed within the context of the running runtime application. For example, the message for a tag write will include the tag being modified, and the tag's previous and new values.

Audit trail detail object shows the information about the selected audit entry on an audit trail list.

When a display containing an audit trail list object is opened, the highlight bar is on the most recent audit message at the top of the list. As new audit messages come in, the old messages scroll down, but the indicator remains at the top. If the highlight bar on the object is not at the top of the list, for example, the operator selects an audit message other

than the top one in the list, the list will not scroll down when new messages come in. Therefore, the new audit will not automatically appear at the top of the list.

Use buttons with the audit trail list

You can also use these key button graphic objects to interact with the audit trail list:

- **Move up**
- **Move down**
- **Page up**
- **Page down**
- **Home**
- **End**

For more information about Key buttons, see Work with lists, trends, alarm banners, and numeric input objects on page 278.

For information about linking buttons to objects, see Link buttons to objects on page 332.

For information about creating graphic objects, see Use graphic objects on page 275.

For information about setting up specific buttons, see Help.

Set up security

This chapter describes:

- How to use FactoryTalk Security to add users and groups accounts
- How to use the **Runtime Security** editor
- How to configure displays to control runtime operator access
- How to provide a way for users to log in and log out
- How to set up FactoryTalk Security in FactoryTalk View Studio
- How to configure user accounts policies
- Examples of how to configure applications to use FactoryTalk View security features

FactoryTalk View and FactoryTalk Security

FactoryTalk View and FactoryTalk Security work together to provide secure access to applications during development and at runtime. FactoryTalk Security is installed as part of the FactoryTalk Services Platform, and uses FactoryTalk Directory to manage user accounts and policies.

Starting with FactoryTalk View ME 8.00.00, FactoryTalk security applies to the open application only. Users and groups are saved with the project and are not shared with other projects.

You can use a combination of FactoryTalk Security and FactoryTalk View security features at runtime to:

- Prevent unauthorized users from running the application.
- Prevent users from opening graphic displays.
- Prevent unauthorized users from stopping the application.
- Prevent users from seeing and using certain parts of a graphic display.
- Prevent users from switching to other applications. This feature uses the DeskLock tool, and is available for personal computers only. You can use this tool to set up a customized desktop and to prevent access to Windows functions. For more information about the DeskLock tool, see Run applications on a personal computer on page 183. The DeskLock tool does not use FactoryTalk Security.

To limit the users who have access to the application or parts of the application, you must set up user accounts and passwords.

If you set up the data source to open graphic displays remotely (using global connections), remote display changes occur whether or not the logged-in user has security access to a given display.

For details about controlling display changes remotely, see Control display changes remotely on page 91.

Tip: The Authenticated User group is not supported on Windows CE. On Windows CE terminals, Windows-linked users must be added to FactoryTalk Directory before they can be logged in.

Use a FactoryTalk View user account

If you do not create additional FactoryTalk View user accounts, FactoryTalk View creates a user named DEFAULT, and this user is automatically logged in when you start applications at runtime. You do not need to set up any other users, and users do not need to log in to use the applications you create.

If you use a FactoryTalk View user account, for example, to run login and logout macros, use the following steps.

To use a FactoryTalk View user account

1. Add users or user groups to FactoryTalk Security.
2. Add the FactoryTalk Security users or user groups to the **FactoryTalk View Runtime Security** editor. For details, see Add FactoryTalk Security users and groups to FactoryTalk View and assign security codes to them on page 146.
3. In the **FactoryTalk View Runtime Security** editor, if desired, assign login and logout macros and security codes to users or user groups. For details, see Help for the editor.
4. In the **Graphics** editor, assign security codes to graphic displays that will have limited access. An alternative is to use object animation to control visibility based on the logged in user's **FactoryTalk View Runtime Security** codes.
5. Provide users with a method for logging in and out.

These steps are described in detail in this chapter.

Set up security

Use the following steps to set up security.

To set up security using FactoryTalk Security and FactoryTalk View security features

1. In the **Explorer** window, in the **System** folder underneath the HMI server, add FactoryTalk Security user groups and users, and set up their passwords.
   You can also add users and groups in the **Runtime Security** editor, on the **ME Runtime 4.00 and later** tab, as described in Add FactoryTalk Security users and groups to FactoryTalk View and assign security codes to them on page 146.
2. In the **Runtime Security** editor, add FactoryTalk Security users or user groups to FactoryTalk View, and assign login and logout macros and security codes to them.
3. In the **Graphics** editor, assign security codes to graphic displays that will have limited access.
4. Provide users with a method for logging in and out.
5. If desired, prevent unauthorized users from stopping the application.
6. Set up policies for passwords and user accounts.

These steps are described in detail in the following sections.

You can also assign security to action groups, connections, and databases. These topics are not discussed in this chapter. For information about these topics, see the FactoryTalk Security Help.

Set up a FactoryTalk Security user

You can set up users in FactoryTalk Security before you add them to the **FactoryTalk View Runtime Security** editor.

FactoryTalk Security allows you to reference user accounts that have already been set up in Windows. These are called Windows-linked users. The link symbol in the **Explorer** window indicates that a user is a Windows-linked user.

To set up a FactoryTalk Security user

1. In the **System** folder, right-click **Users** and select **New > FactoryTalk User** or **New > Windows-Linked User.**
2. Specify the user name, password, and password options. For details, see Help.
3. To add users to a group, right-click the group and select **Properties**. For details, see Help.

Change a user password

After you set up a FactoryTalk Security user, you may need to change the user password.

To change a user password

1. In the **Users** folder, right-click the user name and select **Properties**.
2. Click **Reset Password**. For details, see Help.

Users can change their password at runtime using the **Password** button. For more information, see Change passwords on page 212.

Users can also change their properties at runtime using the **Change User Properties** button. For more information, see Change User Properties on page 224.

Set up a FactoryTalk Security user group

The use of user groups is optional but recommended. Groups allow you to set up security access once for the entire group, rather than individually for members of the group. You can still set up separate permissions for individual members of the group.

Tip: Setting up groups saves time. You can add all the users of an application to a single group in **FactoryTalk Security**, then add the group to the **FactoryTalk View Runtime Security** editor in one step, which is much quicker than adding the users one by one.

FactoryTalk Security allows you to use groups that have already been set up in Windows. These are called Windows-linked groups. The link symbol in the **Explorer** window indicates that a group is a Windows-linked group.

FactoryTalk Security comes with an Administrators group already set up. The local computers Windows Administrators group is automatically added to this FactoryTalk Security group. To start with, members of this group have full security access to **FactoryTalk Security**, and therefore have full access to your applications.

To ensure that you are never locked out of FactoryTalk Directory, we recommend that you assign more than one user to the Administrators group. If one user is locked out, you can use the other one to log in.

To set up a FactoryTalk Security user group

1. In the **System** folder, right-click **User Groups** and select **New User Group** or **New Windows-Linked User Group**.
2. Add user groups.
3. In the **Properties** dialog box, click **Add** to add members to the group.

Tip: To add members to the group later, right-click the group and select **Properties**, or double-click the group.

4. In the **Select User or Group** dialog box, select the user accounts you want to be members of the group and click **OK**.

Runtime Security editor

Use the **Runtime Security** editor to:

- Add FactoryTalk Security users and assign security codes and login and logout macros, for each user or group of users that will have access to the application at runtime.
- Migrate users from RSView version 3.20 and earlier to FactoryTalk View version 4.00 or later.
- Assign user passwords (version 3.20 and earlier applications only).
- Add Windows users or groups (version 3.20 and earlier applications only).

The method for adding users and assigning security codes to them depends on whether you are setting up users for FactoryTalk View 4.00 applications or for RSView 3.20 and earlier applications. There is a separate tab for 4.00 and later applications and for 3.20 and earlier applications.

The **Runtime Security** editor has special items on the **View** menu to control the appearance of the editor. It also has items on the **Setup** menu, and extra tools on the toolbar.

How user accounts and security codes work

When you assign a security code to a graphic display, only the users and groups who are assigned that security code have access to the display. There are 16 security codes, A through P.

The asterisk symbol (*) is used in displays and represents any code. Users must have at least one code assigned to open displays that use the * code.

You can assign combinations of security codes to users and groups, to allow each user or group access to a different set of displays or graphic objects.

To use security codes to restrict user access to graphic objects, assign visibility animation to the objects using expressions containing the security functions. For an example, see Application examples on page 152.

DEFAULT user

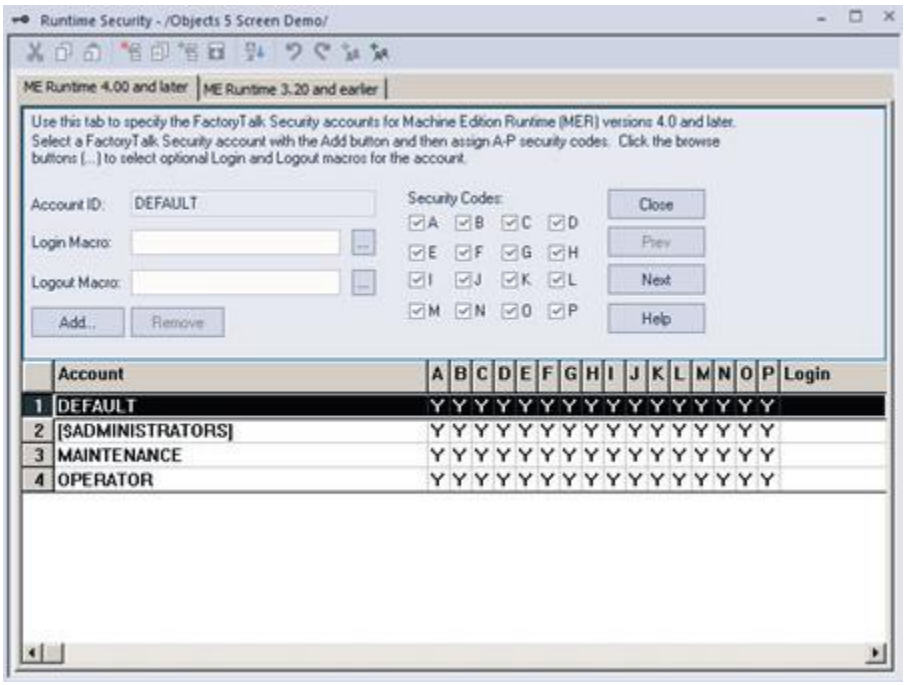
The first account in the editor is the DEFAULT user account. The DEFAULT account is used when no one is logged in. The DEFAULT user initially has access to each security code. Unless you want everyone to have access to all parts of the application at runtime without logging in, turn off the DEFAULT user's access to the security codes you plan to use.

You cannot delete the DEFAULT user account.

Assign the security code for the startup displays to the DEFAULT user, or else the startup displays won't open. If the startup display uses the * security code, you can assign any code from A to P to open the display. For more information, see Assign security to graphic displays on page 148.

Set up users for 4.00 and later applications

Use the **ME Runtime 4.00 and later** tab of the **Runtime Security** editor to set up users for 4.00 and later applications.



Add FactoryTalk Security users and groups to FactoryTalk View and assign security codes to them

Use the following steps to set up users.

To add FactoryTalk Security users and groups to FactoryTalk View and assign security codes to them

1. Open the **Runtime Security** editor and select the **ME Runtime 4.00 and later** tab.
2. Click **Add**.

3. In the **Select User or Group** dialog box, add users and groups, and select the ones to add to FactoryTalk View. You can only select one user or group at a time.
   For details on using the **Select User or Group** dialog box, see Help.
4. In the **Runtime Security** editor, assign **Login** and **Logout Macros** and **A-P Security Codes** to the users and groups.
   - **Login Macro** - *Optional*. To run a macro each time this user logs in, type the macro name in this box, or click the browse button (...) to select from a list of macros in the application.
   - **Logout Macro** - *Optional*. To run a macro each time this user logs out, type the macro name in this box or click the browse button (...) to select from a list of macros in the application. If a new user logs in before the previous user has logged out, the logout macro of the previous user runs after the new user logs in.
   - A-P Security Codes. For a new user, all the boxes are selected by default (Y is shown in all columns in the spreadsheet). To block access to a security code, clear the checkbox for the code in the form section.
     Be sure the DEFAULT user is allowed at least one security code, and assign the same security code (or the * code) to the initial graphic display. If the DEFAULT user has no security codes allowed, when the application is run, no display will open.

Remove a FactoryTalk Security user or group from FactoryTalk View

You might want to remove a user from FactoryTalk View.

To remove a FactoryTalk Security user or group from FactoryTalk View

1. Open the **Runtime Security** editor and select the **ME Runtime 4.00 and later** tab.
2. In the **Account** column, select a user or group, and click **Remove**.
3. Click **Yes**.

Assign security to graphic displays

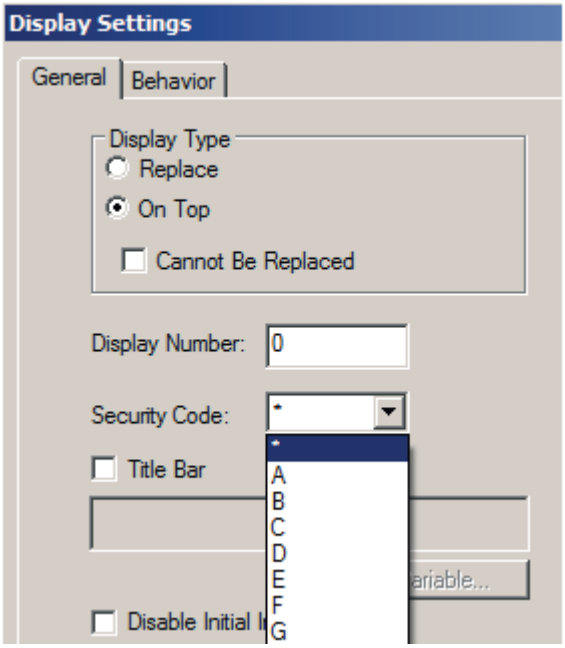
Assign security to graphic displays in the **Graphics** editor. You can:

- Assign security when you create the graphic display, or at any time after creating the display.
- Assign the same security code to more than one display.
- Assign whichever codes that you want, in any order. For example, you can choose to use only the codes D and P and you can assign P before you assign D.

For more information about graphic displays, see Use graphic displays on page 252.

To assign security to a graphic display

1. In the **Graphics** editor, right-click an empty area of the display, and then select **Display Settings**.



2. On the **General** tab, select a security code.
3. To give every security code access to the display, select the asterisk (*). The * is selected by default.

Provide a way for users to log in and log out

Use Login and Logout buttons to provide a way for users to log in and log out. Make sure you place these buttons in graphic displays that all users have access to.

Log in

When the application starts, the DEFAULT user is logged in. If a macro is assigned to the DEFAULT user, the macro runs.

For example, you could create a macro to set the remote display number to open a new login display after the DEFAULT user logs in. For information about controlling display changes remotely, see Control display changes remotely on page 91. Remote display changes are not checked for security.

To provide a way for users to log in

1. Specify the startup displays in the **Startup** editor. The startup displays are the graphic displays that open when your application starts running.
2. The startup display must have the * code or a security code that is assigned to the DEFAULT user, or it will not open.
   For information about the **Startup** editor, see Create runtime applications on page 177.
3. Create a **Login** button on a startup display.
   For information about creating graphic objects, see Use graphic objects on page 275. For information about setting up Login buttons, see Help.

For more information about logging in, see Use your application on page 205.

Log out

When the current user logs out, if a logout macro is assigned to the user, the macro runs. If the user belongs to a group, and a logout macro is set up for the group, the logout macro runs.

To provide a way for users to log out

- Create a **Logout** button on a graphic display that all logged-in users have access to.

After the current user is logged out, the DEFAULT user is logged in. If a login macro is assigned to the DEFAULT user, the macro runs.

You can also set up the application to automatically log out the current user after a specified period of inactivity. For more information, see Log out automatically on page 149.

Log out automatically

You can set up your runtime terminal to log out the current user automatically after the terminal has been inactive for a specified period.

At runtime, when the inactivity period expires:

- The current user is logged out
- Any logout macros assigned to the current user, or user group in Runtime Security execute
- The DEFAULT user is logged in
- The login macro assigned to the DEFAULT user executes
- The specified graphic display opens
- The display Startup and Shutdown macros execute

The automatic logout is independent of FactoryTalk Security settings or Windows security settings. The logout occurs regardless of what you set up in FactoryTalk Security, or Windows.

Set up automatic logout on the Runtime tab

Use the **Runtime** tab to set up automatic logout.

To set up automatic logout on the Runtime tab

1. In the **Explorer** window, double-click **Project Settings**, and then click the **Runtime** tab.

2. Select the **Enable auto logout** checkbox.
3. Specify how many minutes the terminal must be inactive before the automatic logout occurs. Enter a value (1-32767) or select a tag.
4. (Optional) Specify a graphic display to open after the automatic logout.

> **Tip:**
> When the advanced automatic logout is enabled, the automatic logout settings on the **Runtime** tab of the **Project Settings** dialog box are disabled, but the controls are preserved.

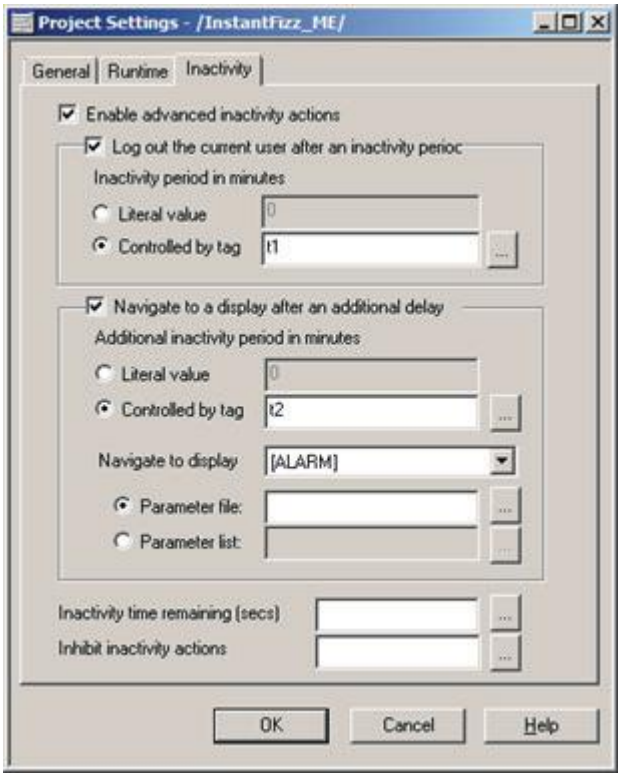
For more information, see "Specify project settings (Runtime tab)" in FactoryTalk View ME Help.

Set up advanced automatic logout on the Inactivity tab

Use the **Inactivity** tab to set up advanced automatic logout.

To set up advanced automatic logout on the Inactivity tab

1. In the **Explorer** window, double-click **Project Settings**, and then click the **Inactivity** tab.



2. Select the **Enable advanced inactivity actions** checkbox.

> **Tip:**
> When the advanced automatic logout is enabled, the automatic logout settings on the **Runtime** tab of the **Project Settings** dialog box are disabled, but the controls are preserved.

3. Select the **Log out the current user after an inactivity period** checkbox.
4. Specify how many minutes the terminal must be inactive before the automatic logout occurs. Enter a value (1-32767) or select a tag. Parameter and expression are not supported.
5. (Optional) Specify a graphic display to open after the automatic logout. Specify how many minutes the terminal must be inactive before the application navigates to the specified display. Enter a value (0-32767) or select a tag. Parameter and expression are not supported.
6. (Optional) Specify a numeric tag to read the remaining inactivity time in seconds. Parameter and expression are not supported.
7. (Optional) Specify a numeric tag to stop the inactivity actions from occurring indefinitely, and reset the inactivity timeout. Parameter and expression are not supported.

> **Tip:**
> If the tag value is zero, the inactivity actions are enabled.
> If the tag value is non-zero, the inactivity actions are disabled.

When setting up advanced automatic logout on the **Inactivity** tab:

If the Log out the current user after an inactivity period checkbox is	And the Navigate to a display after an additional delay checkbox is	Actions that occur after the specified inactivity period expires
Selected (Inactivity period value is 1-32767)	Selected (Additional inactivity period value is 0-32767)	The current user is logged out. If the terminal remains inactive, and the specified additional inactivity period expires, the application automatically navigates to the specified display.
Selected (Inactivity period value is 1-32767)	Clear	The current user is logged out.
Clear	Selected (Additional inactivity period value is 1-32767)	After the additional inactivity period expires, the application automatically navigates to the specified display.

When using a tag to specify the inactivity period, or the additional inactivity period, if the tag value is invalid, no inactivity action occurs.

An invalid tag value could be:

- an alphanumeric string, such as “Start”, or “123Go”
- a value <=0 for the automatic logout action
- a value <0 for the automatic display navigation action

When the tag contains a floating point value, rounding occurs. For example, when the tag value is 2.1, the system rounds down to a value of 2; when the tag value is 3.65, the system rounds up to a value of 4.

For more information, see "Specify project settings (Inactivity tab)" in FactoryTalk View ME Help.

Application examples

This section includes the following application examples:

- Example: Assign security codes to prevent access to graphic displays on page 152
- Example: Use security codes to control the visibility of the shutdown button on page 153
- Example: Assign visibility animation to the Goto display button on page 154
- Example: Assign visibility animation to the shutdown button on page 154
- Example: Prevent unauthorized users from stopping the application on page 154

Example: Assign security codes to prevent access to graphic displays

This example shows how to set up users with access to different graphic displays.

To assign security codes to prevent access to graphic displays

1. Assign security codes to graphic displays as follows:

To this display	Assign this security code
Alarm History	A
Boiler	B

To this display	Assign this security code
Furnace	C
Top Secret	D

2. In the **Runtime Security** editor, assign the security code P to the DEFAULT user.
3. Assign security codes for the displays they can use to the other users:

To this user	Assign these security codes
MAINTENANCE	A, B, C
OPERATOR	A, B
ADMIN	All

The users have access to these graphic displays:

- MAINTENANCE has access to the first three graphic displays.
- OPERATOR has access to the Alarm History and Boiler displays, but not to the Furnace or Top Secret displays.
- The ADMIN user has access to every graphic display.
- The DEFAULT user has access only to graphic displays to which * or P has been assigned. In this example, the DEFAULT user does not have access to any of the four graphic displays.

In this example, you could have assigned the same security code to the Alarm History and Boiler displays, since all users except the DEFAULT user have access to these displays.

Example: Use security codes to control the visibility of the shutdown button

This example shows how to set up users so that only authorized users can use the shutdown button to stop the application.

This example uses the CurrentUserHasCode security function to determine whether a user is authorized to view and use the shutdown button.

The security code E is assigned to users who are authorized to use the button.

To use security codes to control the visibility of the shutdown button

1. In the **Runtime Security** editor, clear all the security codes for the DEFAULT user.
2. Assign security codes to the other users:

To this user	Assign these security codes
OPERATOR	A, B
MAINTENANCE	A, B, C, E
ADMIN	All

3. Create a shutdown button.
4. Assign visibility animation to the button using this expression:

   ```
   CurrentUserHasCode( E )
   ```

   For the **Expression True State**, click **Visible**.

When the graphic display containing the shutdown button is open, the MAINTENANCE and ADMIN user can see the button. If the OPERATOR or DEFAULT user is logged in, the button is not visible.

Example: Assign visibility animation to the Goto display button

This example shows how to assign visibility animation to the Goto display button so that a user must log in before seeing the button.

This example uses the security function CurrentUserName( ). The function returns the string value of the Account ID (user name) for the user who is currently logged in.

The CurrentUserName( ) function is case-sensitive. All RSView 3.20 and earlier user names use uppercase letters.

When the application starts, the DEFAULT user is logged in. The DEFAULT user is also logged in when a user logs out.

To assign visibility animation to the Goto display button

1. Right-click the Goto display button, select **Animation > Visibility**.
2. In the **Animation** dialog box, in the **Expression** box, type this:

   ```
   CurrentUserName( ) <> "DEFAULT"
   ```
3. For the **Expression True** State, click **Visible**.
4. Click **Apply**.

Example: Assign visibility animation to the shutdown button

These examples show how to assign visibility animation to the shutdown button.

To assign visibility animation to the shutdown button so that only the ADMIN user can stop the application

1. Right-click the shutdown button, select **Animation > Visibility**.
2. In the **Animation** dialog box, in the **Expression** box, type this:

   ```
   CurrentUserName( ) == "ADMIN"
   ```
3. For the Expression True State, click **Visible**.
4. Click **Apply**.

To assign visibility animation to the shutdown button so that only users assigned code G or code H can stop the application

1. Right-click the shutdown button, select **Animation > Visibility**.
2. In the **Animation** dialog box, in the **Expression** box, type this:

   ```
   CurrentUserHasCode(GH)
   ```
3. For the **Expression True State**, click **Visible**.
4. Click **Apply**.

Example: Prevent unauthorized users from stopping the application

To prevent unauthorized users from stopping the application

Additional steps for personal computers

- Do not use a title bar on the application window, because the title bar contains a **Close** button. To turn off the title bar for the application window, in the **Project Settings** editor, on the **Runtime** tab, clear the **Title bar** checkbox.

  For more information about using the **Project Settings** editor, see Work with applications on page 49.
- Assign visibility animation to the shutdown button, so that the button is only visible when a user who is authorized to stop the application is logged in. For an example, see Example: Use security codes to control the visibility of the shutdown button on page 153.

  Or, set up security for the graphic display, in which the shutdown button is located, so only users who are authorized to stop the application can open the display.

Additional steps for PanelView Plus CE applications

- Use the DeskLock tool to prevent users from having access to the Windows desktop. This will prevent access to operating system functions such as restarting Windows or shutting down tasks. For more information, see Use the DeskLock tool on page 198.

- Hide the Windows Taskbar to make it inaccessible.

To hide the Windows Taskbar

1. From Windows **Start** menu, select **Settings > Taskbar**.
2. Clear the **Always on top** checkbox.
3. Select **Auto hide**.
4. When you are finished, click **OK**.
5. At the **Command** prompt or in the **Run** box, run **Regflush.exe** to save your changes to persistent internal flash memory. Otherwise, the changes will be lost when you reboot.

Set up FactoryTalk Security for your application

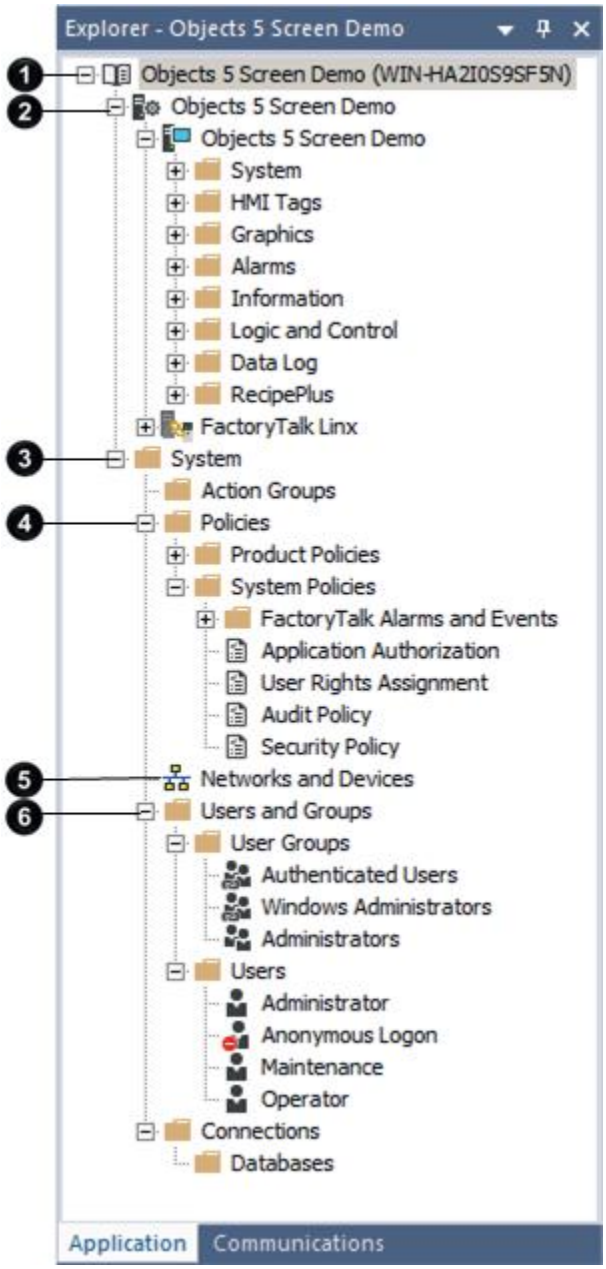
In the **Explorer** window of FactoryTalk View Studio, you can modify additional application behaviors related to security:

- Specify the types of activities to track for audit purposes.
- Specify policies for passwords and user accounts, such as how often passwords must be changed.

You can also change the following security settings in the **Explorer** window:

- Set up access to the application.
- Assign security access to action groups, policies, groups, and users.

When changing security access for these resources, it is important to understand how each change affects the application. For example, it is possible to prevent *any* user from accessing the project by changing these settings.



Item	Name	Description
1	FactoryTalk Directory	Set up access to the FactoryTalk Directory.
2	Application	Set up access to the application.
3	System	Set up access to all the folders.
4	Policies	Set up security policies.
5	Networks and Devices	Set up access to devices.
6	Users and Groups	Add users and groups, and assign passwords. The link symbol indicates Windows-linked groups and users.

Specify activities to track for audit purposes

You can keep an audit trail of activities such as changing the setup of the FactoryTalk Directory and whether security access is denied or granted to users.

To specify which activities to track for audit purposes

1. In the **System Policies** folder, double-click **Audit Policy**.
2. Set up audit policies. For details, see Help.

Specify policies for passwords, accounts, and FactoryTalk sign-on

You can use the FactoryTalk Directory Security Policy to specify how to manage passwords and user accounts, including how long passwords must be, how often they must be changed (if ever), and whether to keep a record of deleted user accounts.

Tip: Remember that any changes that you make to system security policies will be included as part of the runtime application. We recommend that you do not alter the default settings in the **Security Policy Properties** dialog box unless necessary, and unless you are aware of the full implications of the changes. Always test the runtime application to make sure the results are what you intended.

You can also use the Security Policy to turn off FactoryTalk single sign-on. If you turn off this feature, users must log into FactoryTalk each time FactoryTalk View Studio is restarted.

To specify how to manage passwords, user accounts, and FactoryTalk sign-on

1. In the **System Policies** folder, double-click **Security Policy**.
2. Set up options for accounts, passwords, and single sign-on. For details, see Help.

Uncommon security permissions

Although users can adjust the following security permissions, it is uncommon. Care must be taken when changing these security permissions from the default settings.

Set up security access to the FactoryTalk Directory

By default, applications and the System folder inherit the security permissions assigned to the FactoryTalk Directory. This means that you only need to assign permissions to user groups or users once. If desired, you can override the permissions that you set up at this level by assigning different permissions to applications and to the items in the System folder.

You can assign permissions for actions related to FactoryTalk View applications and tags here or at the application level. If you assign them here, they are inherited by all applications.

To set security permissions for the FactoryTalk Directory

1. At the top of the tree in the **Explorer** window, right-click **Local** and select **Security**.
2. Specify which users and groups have access to FactoryTalk Directory, and assign permissions to users and groups for the actions listed in the dialog box. For details, see Help.

Set up security access to the application

Assign FactoryTalk Security at the application level to specify which users or groups can open the application and add, edit, and delete components. You can also specify which users or groups can write to tags and set up, view, and respond to alarms.

To set security permissions for the application

1. Right-click the application name, and then select **Security**.
2. Specify which users and groups have access to the application, and assign permissions to users and groups for the actions listed in the dialog box. For details, see Help.

Set up security access to System policies, groups, and users

You can assign security access to the System folder and to all the folders in the System folder, including policies, groups, and users. You do this by specifying which groups and users have access to the folder, and what actions they can perform.

To set up security access to policies, groups, and users

1. Right-click the **System** folder, or a folder in the System folder, and then select **Security**.
2. Specify which groups and users have access to the folder, and what actions they can perform. For details, see Help.

Set up security access to networks and devices

You can assign security access to networks and devices in general, to the development computer, and to specific devices. Set up security for networks and devices if you are using RSLinx Classic devices.

To set up security access to networks and devices

1. In the **System** folder, right-click **Networks and Devices**, the development computer workstation, or a specific device, and then select **Security**.
2. Specify which groups and users have access to the item, and what actions they can perform. For details, see Help.

Set up language switching

This chapter describes:

- What language switching is
- Summary of steps for setting up language switching
- How to set up Windows to support language switching
- How to add languages to the application
- How to export application text strings for translation
- How to translate application text
- How to import translated text files
- How to switch application languages at design time
- How to set up multiple language support for graphic libraries

About language switching

The FactoryTalk View language switching feature provides the ability to set up multiple languages for an application and switch languages dynamically at runtime. You specify an initial language for the runtime application when you create it, and select the languages that will be available at runtime. You can use up to 40 languages per development application, and 20 languages per runtime application.

When the application runs, operators can change the language using a **Language Switch** button. Set up a different **Language Switch** button for each language.

With language switching you can:

- Develop an application in one language, export the user-defined text strings for the application, and then import translated strings for up to 40 languages into the same application.
- Export an application's text strings in multiple languages into a single Microsoft Excel spreadsheet.
- Import text strings in multiple languages from a Microsoft Excel spreadsheet into your application in a single operation.
- Use the same application in different countries, allowing operators in each location to view the application in their own language.
- Allow operators in multilingual countries to use the language of their choice.
- Import application components developed in different countries into a single application that supports multiple languages.

Default language

For applications that use multiple languages, you can specify one of the languages as the default language. Any undefined text strings in the other languages can be shown in the default language.

The default language is also used for RSView Machine Edition 4.00 applications. When you open or import the application into FactoryTalk View 5.00 or later, the application's alarm, information, and local messages are associated with the language that you select to be the default language. For example, if you open an RSView Machine Edition 4.00 application that contains local messages, and you select French for the default language, the local messages will be treated as if they are French when you export the application's strings for translation. That is, they will appear in the **French language** column. You can add other languages for these messages, as described in this chapter.

When you create an application, the language you specify for the application is the default language. You can change the default language at any time. The default language can be any of the languages in the application.

You can show undefined strings in the default language during application development. If you do not select this option, undefined strings are shown as a question mark "?" character. If you select this option, but do not include the default language with the runtime application, undefined strings are shown as a question mark "?" character at runtime.

Alarm trigger labels are shown in the Trigger Label Selector list using the default language. If you change the default language, you must re-specify the trigger labels for every graphic object that filters alarms. To do this, specify the trigger label names in the new default language. For more information about filtering alarms when your application uses multiple languages, see Filter alarm triggers in multiple languages on page 98.

Change the default language

Use the **Language Configuration** dialog box to change the default language.

To change the default language

1. From the **Tools** menu, select **Languages**.

2. Select the language to use for the default language.
3. Click **Set As Default**. A large check mark to the left of the selected language indicates that it is now the default language.
4. To use this language for all undefined strings in the application, select **Display undefined strings using the default language**.

Set up language switching

Use the following steps to set up language switching for an application.

To set up language switching

1. Set up Windows for language switching.
   - For applications that will run on a personal computer, install the Windows languages on page 161 that the application will use.
   - For applications that will run on a PanelView Plus or PanelView Plus CE terminal, set up the fonts on page 161 that the application will use.
2. Create, open, or import the application in the language of your choice. For details, see Work with applications on page 49.
3. Add languages to the application. For details, see Add languages to the application on page 162.
4. Create graphic objects and specify the text strings that they will use.
5. Create alarm, information, and local messages.
6. Create a **Language Switch** button for each language you plan to use at runtime. For details, see Help.
7. Export the application text strings for translation. For details, see Export application text strings for translation on page 162.
8. Translate the text strings. For details, see Translate application text in Unicode files on page 165.
9. Import translated text strings for each of the translation languages. For details, see Import text on page 168.
10. Open the application and switch languages to check the layout of the translated text. For details, see Switch application languages at design time on page 169.
11. Create the runtime application, specifying the startup language and the languages that the operator can switch to. For details, see Create runtime applications on page 177.

Install Windows languages

We recommend that you install all the languages that your application will use, on both the development and runtime computers. Installing languages turns on the Windows font support features, which allow applications to show characters for different languages using a single application font.

For details about installing languages, see Windows Help.

PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, and PanelView Plus CE terminals are shipped with languages already installed.

Set up Windows fonts

For applications that will run on a PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, install the fonts that the application will use. For applications that will run on a personal computer, the fonts you want are generally installed when you install the languages for the application. Using MS Arial Unicode as the default font provides better performance for language switching.

For details about choosing fonts that work well with language switching, see Choose fonts for language switching on page 283.

Make sure that the development and runtime computers are set up to use the fonts you want for the title bar and inactive title bar. Specify the font in the Windows Control Panel, in the **Display Properties** dialog box.

For information about setting up fonts on a PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, see the *PanelView Plus Terminals User Manual*.

Windows locale settings

Windows locale formatting determines how the application shows time, date, and floating point values at design time and runtime.

When the application language is switched, the locale settings for the new language are used even if that language has not been installed.

You do not need to edit the default locale settings.

Add languages to the application

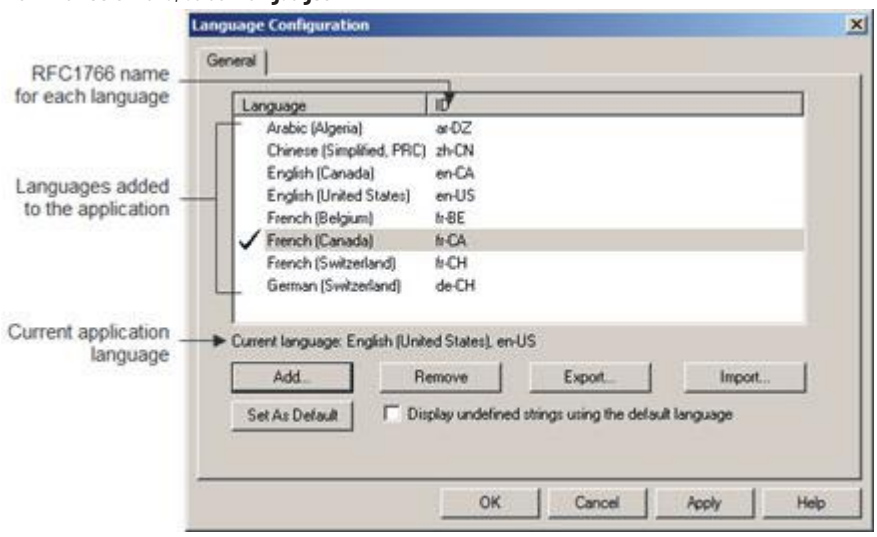
Use the **Language Configuration** dialog box in FactoryTalk View Studio to add up to 40 languages to the application.

Add languages before you:

- Create **Language Switch** buttons
- Create the runtime application

To add languages to an application

1. From the **Tools** menu, select **Languages**.



2. Add languages. For details, see Help.

Remove languages

You can also use the **Language Configuration** dialog box to remove languages. When you remove a language, all the strings for the language are deleted the next time you save the application. Save a component deletes strings for the removed language from the component.

Export application text strings for translation

There are two options for exporting text strings:

- One language at a time, saving the text in a tab-delimited text file in Unicode format. The file can be translated and then imported back into the application.
- All languages at once, exporting into a single Microsoft Excel spreadsheet. The strings in the spreadsheet can be translated and then imported back into the application.

These text strings allow language switching and are exported for translation:

- Text objects created in graphic displays and global object displays
- Graphic and global object display titles specified in the **Display Settings** dialog box
- Labels and captions on graphic displays, including alarm trigger labels
- Embedded time and date and numeric variables
- Alarm, local, and information messages
- Captions and shorter messages in the dialog boxes of **User Management** buttons and the **Electronic Signature** dialog box
- Text strings defined in the extended properties of controller tags

Tip: Make sure that you define the controller default language when multi-language is enabled.

- If the requested language is not available in the controller, the application shows the default language defined and enabled in the HMI.
- If the HMI default language is not defined in the controller, the application uses the controller default language or the single language that does not have an ID.

Undefined strings are exported as undefined strings, regardless of whether you select the option in the **Language Configuration** dialog box to show undefined strings using the default language. That option is used only for showing strings during application development and at runtime. It does not substitute the default language's strings for undefined strings in the exported file or spreadsheet.

Export text in Unicode format

When you export text in Unicode format, these text strings are also exported:

- Text descriptions and error messages for the Change Password window. These strings are exported if you use a password button in your application.

You can translate these strings, but you cannot change their language dynamically at runtime the way you can for graphic object strings.

For the **Change Password** window, the runtime application will use the strings that are in the application when the runtime application is created. For example, if the current application language is French, and you have imported French text strings for the Change Password window, then the French strings will be compiled in the runtime application. If the operator opens the window, the strings will be shown in French. If the operator changes to another language, the strings will still be shown in French.

Unicode file name format

The format for the exported Unicode text file name is <*ApplicationName*><*HMIServerName*><*RFC1766*>.txt, where:

- <*ApplicationName*> is the name of the application.
- <*HMIServerName*> is the name of the HMI server containing the text strings that you exported. This is always the same name as the application name.
- <*RFC1766*> is the RFC1766 name associated with the language selected for the export operation.

For details about the schema of exported Unicode text files, see File schema on page 166.

For a list of Windows languages and the corresponding RFC1766 names, see Mapping languages to RFC1766 names on page 465.

Export text to a Microsoft Excel spreadsheet

FactoryTalk View supports Microsoft Excel versions 2000, 2002, 2003, 2007, 2010, and 2013. The exported file is in Microsoft Excel 5.0/95 Workbook format, with the extension .xls.

The maximum number of characters viewable in an Excel cell depends on the version of Excel and the font size. If the maximum is exceeded, the cell will be blank.

Optimizing duplicate strings

When you export text to an Excel spreadsheet, you have the option of optimizing duplicate strings. FactoryTalk View analyzes the text and indicates repeated instances of the same string using a reference number. This way the translator only needs to translate the text once. When you import the translated strings from the spreadsheet into the application, FactoryTalk View copies the translated string into all the graphic objects that used the string originally.

Excel spreadsheet file name format

The format for the exported spreadsheet file name is <*ApplicationName*>_<*ExportVersion*>.xls, where:

- <*ApplicationName*> is the name of the application.
- <*ExportVersion*> is an incremental number assigned to each successful export operation.

For details about the schema of exported spreadsheet files, see Translate application text in Excel spreadsheet files on page 165.

For a list of Windows languages and the corresponding RFC1766 names, see Mapping languages to RFC1766 names on page 465.

Export application text

Use the **Language Configuration** dialog box to export application text.

To export application text

1. From the **Tools** menu, select **Languages**.
2. In the **Language Configuration** dialog box, select the application language for which to export text strings.
   If you are going to export strings for all languages defined for the application, it does not matter which language is selected in the **Language Configuration** dialog box.
3. Click **Export**.
4. Follow the instructions in the String Import Export Wizard.
   For details about options in the String Import Export wizard, click **Help**.

Exported language files are saved in the default folder **\Users\Public\Public Documents\RSView Enterprise \Strings**. You can also specify another location.

If you export to a Unicode text file multiple times to the same location, previously exported versions are overwritten.

Problems exporting

You do not have to check every text file created during an export to verify that text strings were exported correctly. If errors occur, or if you cancel the export while it is in progress, a message appears in the **Diagnostics List** and in the FactoryTalk Diagnostics log file.

In addition, errors are shown when they occur in a log file called *ExportErrors.txt* in **\Users\Public\Public Documents\RSView Enterprise\Strings**

Existing error log files are overwritten for each subsequent export operation that generates errors.

Translate application text in Excel spreadsheet files

This section contains information about the format and schema of the exported Microsoft Excel spreadsheet file.

The spreadsheet contains a header row followed by a row for each text string in the application. There is a column for each language defined in the application. The default language is first, followed by the remaining languages in alphabetical order by RFC1766 name. All cells use the text format.

Do not delete the header row in the spreadsheet. FactoryTalk View requires this information when you import the translated text into your application. Similarly, do not change the cell formats.

If you select the option to optimize duplicate strings, there is a reference column to the left of each language column. Do not change the information in the reference columns.

Translate application text in Unicode files

This section contains information about the format and schema of the exported Unicode text file. It also contains information about using Microsoft Excel or Windows Notepad to edit the file.

Because FactoryTalk View requires that parts of the text file remain the way they were exported, give the information in this section to the translator, to ensure that the file can be imported after it has been modified.

File name and format

You will likely want to rename the file before translating it, to avoid confusing it with the original file. You can use any file name ending with the file extension .txt.

To import text into a FactoryTalk View application, you must save the file as tab-delimited text, in Unicode text format.

Specify the text file in Microsoft Excel

When you open the text file in Microsoft Excel, the Text Import Wizard appears.

1. Specify the file format:
   a. Click **Delimited**.
   b. In the **Start import at row** box, type or select 1.
   c. From the **File origin** list, select **Windows (ANSI)**.
   d. Click **Next**.

2. Specify the field delimiter:
   a. Select the **Tab** checkbox. If any other checkboxes are selected, clear them.
   b. Make sure the **Treat consecutive delimiters as one** checkbox is cleared.
   c. Click **Next**.
3. Specify the column data format:
   a. If it is not selected already, under **Column data format**, click **General**.
   b. Click **Finish**.

Save the text file in Microsoft Excel

After you specify the text file in Excel, you must save the file.

To save the file

1. From the **File** menu, select **Save**.
   Excel warns that the file may contain features that are not compatible with Unicode text.
2. When prompted to keep the workbook in Unicode format that leaves out incompatible features, click **Yes**.
3. Close the file.
4. When prompted to save changes, click **Yes**.
5. When prompted again to keep the workbook in Unicode format that leaves out incompatible features, click **Yes**.

Differences in file format for 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.

You do not have to change the format of the file before you import it into FactoryTalk View.

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.

Save the 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 columns:

Column	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 example shown earlier, string reference 1 might refer to a Momentary push button in the graphic display called Pump station.

The string definition is enclosed in quotes in Notepad, but not in the spreadsheet column in Excel.

In the translated text file, the only text that can be modified is the text inside the quotation marks in the string definition column. For example, translated into German, the file would contain these changes:

Column	Component type	Component name	String reference	"String definition"
Example	Graphic Display	Pump station	1	"Motor abschalten"

Do not change the entries in the component type or component name columns, unless the component was renamed in the application after the text was exported.

Do not modify the string reference number. The string reference number is a unique number that is assigned to an object by FactoryTalk View. Modifying the string reference number prevents FactoryTalk View from identifying the object correctly when you import the text.

Work 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""

If the string definition contains an odd number of double quotes, the number of double quotes is rounded down to an even number, and then each pair is imported as one double quote. For example, the string:

"Call "Duty Manager"

appears in the application as:

Call Duty Manager

Work with backslashes and new line characters

To force text to begin on a new line, precede the text with the characters **\n**. For example:

Motor\nStopped

appears in the application as:

Motor

Stopped

To make the characters \n appear as part of the text, type \\n. To make a backslash appear in the application, type two 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

Import text

You can import text strings saved in Unicode text format or in Microsoft Excel spreadsheet format.

If your text strings are in Microsoft Excel spreadsheet format, only text strings that exist in the spreadsheet are imported. This means that existing strings in the application are not deleted during the import if they are not in the spreadsheet.

Back up the text in your application before importing. You can do this by exporting the text.

Save the backup file in a different location than the translated file that you are about to import.

To import text into your application from a text file

1. From the **Tools** menu, select **Languages**.
2. In the **Language Configuration** dialog box, select the application language for which to import text strings.
   If you are going to import strings for multiple languages defined in an Excel spreadsheet, it does not matter which language is selected in the **Language Configuration** dialog box.
3. Click **Import**.
4. Follow the instructions in the String Import Export Wizard.
   For details about options in the String Import Export wizard, click **Help**.

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 text that was originally in the application, import the text from the backup file you created.

Problems importing

You do not have to check every graphic display in your application to verify that text was imported correctly.

If errors occurred while importing text, they are shown automatically from a log file called *ImportErrors.txt* in **\Users\Public\Public Documents\RSView Enterprise\Strings**

Each time errors occur while importing text into an application, the ImportErrors.txt file is overwritten.

If errors occurred while importing text or if the import was canceled, a message appears in the **Diagnostics List** and in the FactoryTalk Diagnostics log file.

Cancel importing

If some, but not all, of the text in an application seems to have 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 text originally in the application, import the text from the backup text file you created before importing.

Switch application languages at design time

If an application is designed with multiple languages, you might want to switch application languages at design time to check the layout of translated text.

To switch application languages at design time

1. From the **Application** menu, select **Change Application Language**, or on the toolbar, click  .
2. In the **Change Application Language** dialog box, select the language you want to change to.

   Tip: The language list shows the languages added to the application, first with the language name and then with the language RFC1766 identification code.

3. Click **OK**.

The text in the following dialog boxes will be changed to the selected language:

- Alarm Setup
- Display
- Global objects
- Information Messages
- Local Messages

If any of the dialog boxes is opened and has unsaved changes, a message prompts you to save the changes.

Set up multiple language support for graphic libraries

When you create an application, the graphic libraries installed with FactoryTalk View are "language neutral" by default.

This means that the graphic objects in the libraries always show their text strings as shipped, regardless of the current application language. Since the current application language is ignored, text strings never appear as undefined.

You can change the default for any graphic library, so that it supports multiple languages instead of a single language. When you do this, when you open the library you can view the text strings that are defined for the current application language. If the library's text strings have not been defined for the current language, the undefined strings appear as single question marks.

If you turn off a graphic library's multiple language support, and then save the library, only strings for the current application language are saved. The current application language therefore becomes the "language neutral" language. Any strings for other languages are deleted.

To turn on support for multiple languages in a graphic library

1. Right-click an empty area in the graphic library display, and then select **Display Settings.**
2. On the **General** tab, select **Support Multiple Languages**.

After you turn on support for multiple languages, when you save the graphic library, all strings that support language switching are saved in the current application language. For information about saving libraries in multiple languages, see Use libraries to store displays with multiple languages on page 263.

Use graphic libraries that support multiple languages

To use the graphics libraries in an application, you can add a graphic library into the **Displays** folder or copy objects from a graphic library into a graphic display. If the graphic library supports multiple languages:

- When you add the graphic library into the Displays folder, all strings, including strings for languages that the application does not support, are included with the display.
- If you copy an object from the graphic library into a graphic display, only strings for languages supported by the application are copied.

For more information about using graphic libraries, see Use graphic displays on page 252.

Set up display navigation

This chapter describes:

- What display navigation is
- How to develop a hierarchy of displays
- How to test display navigation
- How to use graphic objects to navigate
- How to control display changes remotely

This chapter describes methods for navigating between graphic displays. For information about navigating between objects on a graphic display, see Use the keyboard to navigate to and select objects on page 331.

About display navigation

The term display navigation refers to the way that the operator moves between the graphic displays that make up an application.

Use these methods to set up display navigation for your application:

- Develop a hierarchy of graphic displays, to chart how users will navigate the application.
- Determine which users will have access to which parts of the application.
- Create graphic objects that the operator can use to navigate the application.
- Use the **Remote Display Number** connection to automatically control display changes. The use of this connection is optional. Assign it in the **Global Connections** editor.
- In the **Startup** editor, specify the graphic displays to open when the application starts.
- Set up security so that only authorized users have access to the application or parts of the application.

For information about setting up security, see Set up security on page 143.

Hierarchy of displays

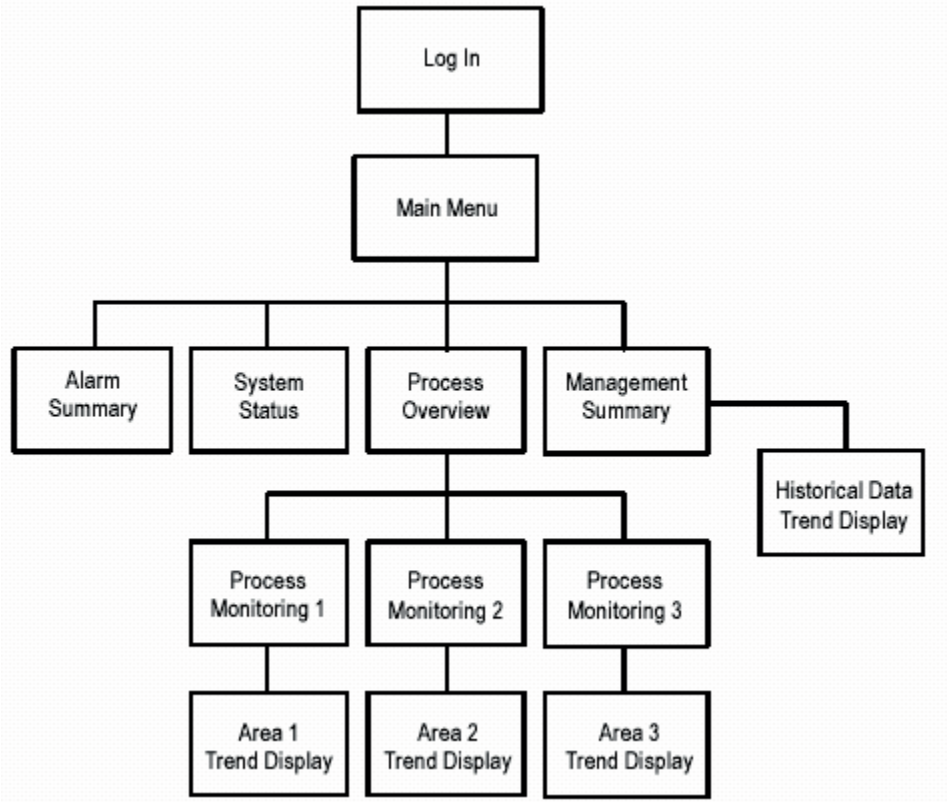
A hierarchy of displays is a series of graphic displays that provide progressively more detail as users move through them. Design your display hierarchy to meet the needs of the various users, including managers, supervisors, and operators.

If you plan to use security, determine which groups of users need access to which displays, and decide where in the hierarchy to locate login, logout, password, and shutdown buttons. For information about setting up security, see Set up security on page 143.

A display hierarchy could include:

- An initial graphic display for logging in
- A graphic display that serves as a menu
- An overview of the plant
- A comprehensive display of each process being monitored
- Process-specific displays that provide more detail
- Management summary displays
- Trend displays of historical and current data

The following illustration shows a sample display hierarchy.



Test display navigation

Once you set up display navigation for your application, test the application to make sure that navigation flows smoothly and that you have avoided problems like these:

- A graphic display contains no buttons for moving forward or back.
- When a graphic display closes, no other display is open and there is no way to continue using the application.

For information about testing your application, see Test your application on page 177.

Graphic objects for navigation

Use these graphic objects to navigate through the displays in the application:

Use this object	To do this
Goto display button	Open the specified graphic display.
Goto configure mode button	Stop the application and open the FactoryTalk View ME Station dialog box.
Return to display button	Close the current display and open the previous display.
Close display button	Close the current display. Can send a value to a tag when the display closes.
Display list selector	Provide a list of graphic displays so the operator can select which display to open.
Shutdown button	Stop the application and exit FactoryTalk View ME Station.

The behavior of these graphic objects depends on which types of graphic displays are used. See the following sections for details.

For detailed information about setting up graphic objects, see Help.

Language switching

If the application uses multiple languages and operators will be switching languages at runtime, place **Language Switch** buttons on a display that the operators have access to. For example, put the buttons on the display that opens when the operators log in. Create a **Language Switch** button for each language that the operators will be using.

For more information about setting up multiple languages, see Set up language switching on page 160.

Display type

The display type that you use gives you additional control over how the operator navigates between displays. For example, use the On Top type to keep a graphic display on top at all times, even when another display has focus. Or use the Replace type if you want a display to replace all other open displays when it opens.

For more information about display types, see Replace displays on page 259 and On Top displays on page 259.

Goto display buttons

When you set up a Goto display button, specify the graphic display to open when the operator presses the button at runtime.

The specified display does not open if the operator does not have security access for the display.

You can also assign a parameter file or parameter list that assigns tags to tag placeholders on the display when the display opens. For more information about parameter files and lists, see Use parameters and global objects on page 374.

For information about setting up **Goto display** buttons, see Help.

The Goto display button's behavior at runtime also depends on which types of graphic displays are already open and which type of display it is opening:

- If the graphic display assigned to the button is a Replace display, it closes any open On Top or Replace displays. It does not close any On Top displays that use the **Cannot Be Replaced** option.
- If the display assigned to the button is already open, but does not have focus, pressing the button gives the display focus.

The operator cannot use the Goto display button to open Replace displays if display change is currently controlled remotely (using global connections). But the operator can still open On Top displays.

Goto configure mode buttons

When the operator presses a goto configure mode button at runtime, the current application stops running and the **FactoryTalk View ME Station** dialog box opens.

While in configure mode, the operator can use FactoryTalk View ME Station to change applications, application settings, and terminal settings. The operator can also delete log files. For more information about the **FactoryTalk View ME Station** dialog box, see Run applications on a personal computer on page 183.

Return to display buttons

When the operator presses a return to display button at runtime, the graphic display that the button is on closes and the display that was previously open reopens.

The current display does not close if:

- The display change is currently controlled remotely (using global connections).
- There were no previously opened Replace displays.
- The operator does not have security access for the previous display. This can only occur if a new user logs in using a login button in the current display.

The return to display button only goes back to the most recent display. It does not go back through a series of displays. For information about setting up return to display buttons, see Help.

Both the current display and the previous display must be Replace displays. When the operator presses the return to display button:

- If the graphic display that is closing is a Replace display, the display closes and the previously opened Replace display opens. Any On Top displays that were previously open with the Replace display are not reopened.
- If the graphic display that is closing is an On Top display, the display closes but no display is reopened.

We therefore recommend that you use return to display buttons in Replace displays only.

Example: Navigating through displays

This example uses the display hierarchy illustrated in Hierarchy of displays on page 171, and shows what happens as the operator navigates through the hierarchy. The graphic displays are all Replace displays.

1. On the Main Menu display, the operator uses a display list selector to open the Process Overview display.
2. On the Process Overview display, the operator presses a Goto display button to open the Process Monitoring 1 display.
3. After viewing the state of the process, the operator presses a return to display button to close the current display and reopen the Process Overview display.
4. On the Process Overview display, the operator presses a return to display button. Which display opens? The Process Monitoring 1 display (because this was the previously opened display).

To return to the Main Menu display from the Process Overview display, the operator would have to press a Goto display button that is set up to open the Main Menu display.

Close display buttons

When the operator presses a close display button at runtime, the graphic display that the button is on closes. You can set up the button to write out a value when the display closes.

If the graphic display that is closing is a Replace display, and the display change is controlled remotely, the display does not close. If the display does not close, the close value, if any, is not written out.

If the display change is controlled by the operator and the graphic display that is closing is a Replace display, if there are no On Top displays open, an empty window is shown. The operator will not be able to use the application again (unless a remote display change occurs or an alarm, activity, or information message display opens).

We therefore recommend that you use close display buttons in On Top displays only.

For information about setting up close display buttons, see Help.

Display list selectors

Use the display list selector to show a list of graphic displays that the operator can choose from. The operator can scroll through the list and select the graphic display to open.

The specified display does not open if the operator does not have security access for the display.

You can also assign a parameter file or parameter list that assigns tags to tag placeholders on the display when the display opens. For more information about parameter files and lists, see Use parameters and global objects on page 374.

For information about setting up display list selectors, see Help.

The display list selector's behavior at runtime also depends on which types of graphic displays are already open and which type of display it is opening:

- If the selected graphic display is a Replace display, it closes any open On Top and Replace displays. It does not close On Top displays that use the Cannot Be Replaced option.
- If the selected display is an On Top display, it opens on top of the current display. The current display does not close.

The operator cannot use the display list selector to open Replace displays if the display change is currently controlled remotely. But the operator can still open On Top displays (with or without the Cannot Be Replaced option).

Select the display to open

The operator can scroll through the list and select displays using the key button graphic objects, or, if the list has the input focus, by using the arrow keys and Enter key on a keypad or external keyboard.

You can link key buttons to a specific display list selector, or set up the buttons to work with whichever object is selected in the graphic display.

For information about input focus, see What input focus looks like on page 331. For information about linking buttons to the display list selector, see Link buttons to objects on page 332.

Shutdown buttons

When the operator presses the shutdown button at runtime, the application stops and FactoryTalk View ME Station closes.

To prevent an unauthorized user from stopping the application, assign visibility animation to the shutdown button. For details, see Application examples on page 152. Or, place the button in a display that only authorized users have access to.

For information about setting up shutdown buttons, see Help.

Control display changes remotely

To control display changes remotely, you can set up the data source to open graphic displays using global connections.

For example, the **Remote Display Number** connection is a global connection that you can use to control display changes from the data source.

For more information, see About global connections on page 89.

Create runtime applications

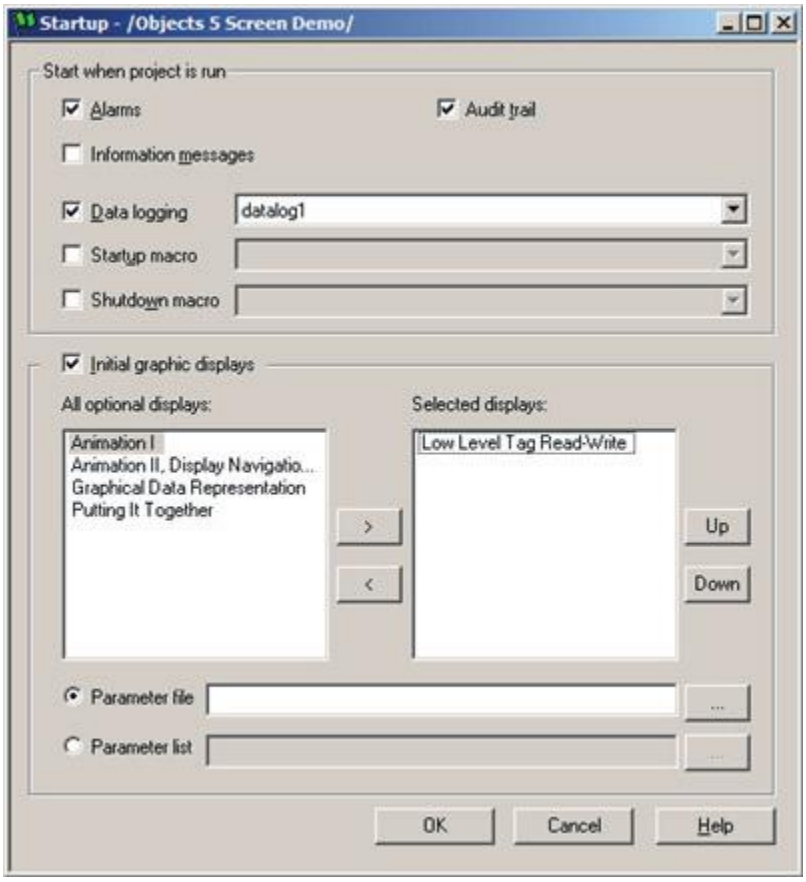
This chapter describes how to:

- Specify startup settings
- Test your application
- Create runtime application files
- Convert runtime application files to development applications

Specify startup settings

Use the **Startup** editor to specify which application processes and components to start when the application starts at runtime.

You can specify startup settings once you have set up all the parts of the application, or you can specify processes and select components in the **Startup** editor as you create them.



For detailed information about the options in the **Startup** editor, see Help.

Test your application

You can test your application in FactoryTalk View Studio at any time during the development process, to make sure that everything works the way you intend.

If the development computer is connected to the data source, you can test all functions of the application, including security settings, language switching, communications, and alarm monitoring.

A FactoryTalk View ME Station emulator opens on the development computer and runs the application. This runtime version of the application is a temporary version for testing use only. You cannot run it on another computer.

There is a two-hour time limit for test running the application in FactoryTalk View Studio.

The procedure in this section shows you how to test your entire application. For information about testing a single graphic display, see Test your displays as you work on page 258.

To test your application in FactoryTalk View Studio

1. From the **Application** menu, select **Test Application**, or click the **Test Application** tool  .
2. If your application uses multiple languages, specify the languages to include and the initial runtime language, and then press **Finish**. For details, see Help.
3. Test your application.
4. To stop your application, press a shutdown button, or type the character *x*.
   Make sure you provide the operator with a method for shutting down the application at runtime. For more information about methods for shutting down applications, see Shut down applications on page 186.

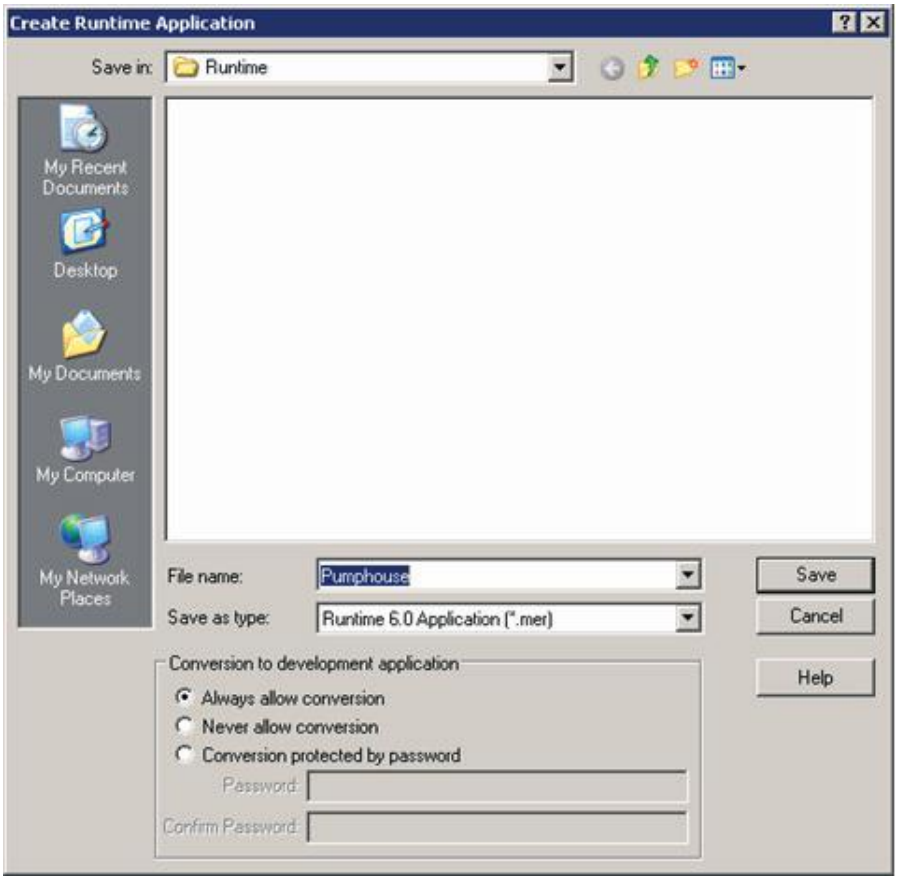
Once you have tested your application to make sure everything works the way you intend, create the runtime application file, and transfer the file to the runtime computer.

Create runtime application files

Before you can run an application, you must create a runtime version. When you create the runtime version, FactoryTalk View Studio compiles all of the necessary application information into a single file with the extension .mer.

To create a runtime application

1. In FactoryTalk View Studio, with the application open, do either of the following:
   - From the **Application** menu, select **Create Runtime Application**.
   - On the toolbar, click  .



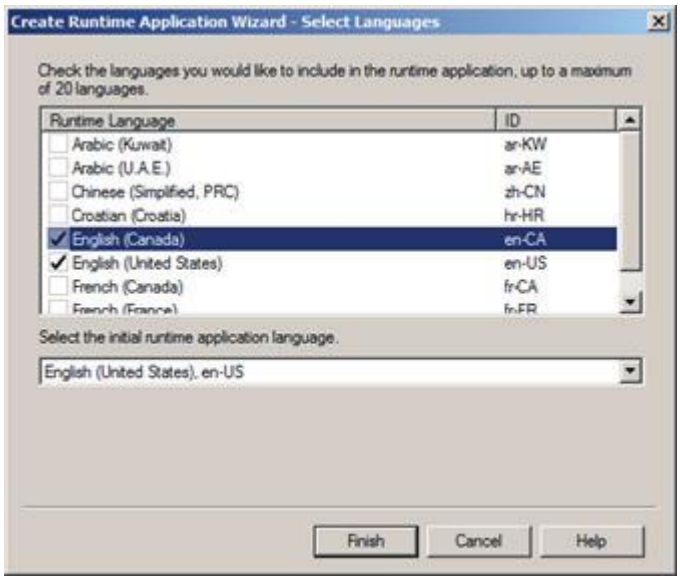
2. Specify the folder, in which to save the runtime application file.
3. In the **File name** box, enter the name for the application file.
   The default name is the same name as the development application.
4. In the **Save as type** box, specify the version of FactoryTalk View ME Station for which to create the .mer file.
5. In the **Conversion to development application** area, select one of the following options:
   - **Always allow conversion** - The .mer file can always be used to generate an .med file.
   - **Never allow conversion** - The .mer file can never be used to generate an .med file.
   - **Conversion protected by password** - The .mer file can only be used to generate an .med file by providing a password.
6. If you want to include FactoryTalk ViewPoint content in the .mer file, select the **Include ViewPoint content** checkbox and from the list, select a FactoryTalk ViewPoint version.

Tip: To enable the **FactoryTalk ViewPoint** checkbox and list, publish displays from FactoryTalk ViewPoint first. For details, see FactoryTalk ViewPoint Help.
To create a v8.20 or earlier version runtime application that includes FactoryTalk ViewPoint content, .NET Framework 3.5 is required. For information about how to install .NET Framework 3.5, see *FactoryTalk View ME Installation Guide*.

---

7. Click **Save**.
8. If there are multiple languages in the application, the language wizard opens. Select the language or languages to be included in the runtime application and click **Finish**.

Tip: To show undefined strings in the default language at runtime, include the default language in your selection. The default language is also required for filtering alarms.



FactoryTalk View Studio creates a runtime version of the application in the directory you specify. If any editors are open, you are prompted to save any unsaved changes.

For more information about the default language, see The default language on page 159.

For information about transferring the runtime application:

- To a personal computer, see Run applications on a personal computer on page 183.
- To a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, see Transfer applications to a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal on page 199.

Create .mer files for previous versions

You can specify the version of FactoryTalk View ME Station for which to create the .mer file. For example, if the application will run on a terminal that uses FactoryTalk View ME Station 5.10, you can specify that version for the .mer file.

Tip: Version 6.0 or earlier .mer files created on a 64-bit operating system cannot be converted to design-time projects (.med).

If the application contains features that the version you select does not support, FactoryTalk View shows a validation report that lists the unsupported features. The runtime application file is not created. You must remove or turn off the unsupported features before you can create the runtime application file.

For information about the features supported in different versions of FactoryTalk View, and how to remove or replace them, see Features supported in different versions of FactoryTalk View on page 424.

IMPORTANT: The PanelView Plus 7 and PanelView Plus 6 terminals cannot process a Gradient Fill that uses a Diagonal Up or Diagonal Down direction, nor can the terminal apply a Gradient Fill to a rotated object. When running an MER on a PanelView Plus 7 or PanelView Plus 6 terminal and the system encounters an object with these attributes, the system shall use the gradient fill starting color as the object's solid color background.

When running the MER on a computer, the system can process all gradient fills and this warning does not apply.

Convert .mer files to development applications

For FactoryTalk View version 5.00 and later .mer files, you can convert the runtime application (.mer file) back to a development application (.med file). This option is useful if:

- You lose or don't have access to the original .med file and you want to make a change to the application.
- You make changes to the .mer file at runtime, such as adding passwords, updating FactoryTalk Linx device shortcuts, or saving recipes, and you want to make a change to the application but don't want to lose this runtime information.
  Starting with FactoryTalk View 8.0, all security information modified at runtime is saved in the .mer files. These changes will be restored after converting FactoryTalk View 8.0 .mer files to .med files.
- You want to extract information that was generated at runtime, such as updated RecipePlus files.

Version 6.0 or earlier .mer files created on a 64-bit operating system cannot be converted to design-time projects (.med files). This is due to an incompatibility of the previous FactoryTalk View database engine with 64-bit operating systems. Thus, when you create a version 6.0 or earlier .mer file on a workstation running 64-bit operating system, the only option available in FactoryTalk View Studio is **Never allow conversion**. When creating a version 6.0 or earlier .mer file on a workstation running a 64-bit operating system, this is the default option (and only choice) for the **Conversion to development application setting**.

When you create the .mer file, you can specify whether to allow conversion of the runtime .mer file, and whether to require a password to convert the runtime application.

For more information about converting .mer files to development applications, see Convert runtime application files to development applications on page 181.

Convert runtime application files to development applications

For FactoryTalk View version 5.00 and later .mer files, you can convert the runtime application file back to a development application. This allows you to make changes to the application without losing runtime information such as passwords and RecipePlus data.

To convert the runtime application file, use the Restore runtime application operation in the FactoryTalk View ME Application Manager. Data logging and alarm data that occurred at runtime are not restored to the development application.

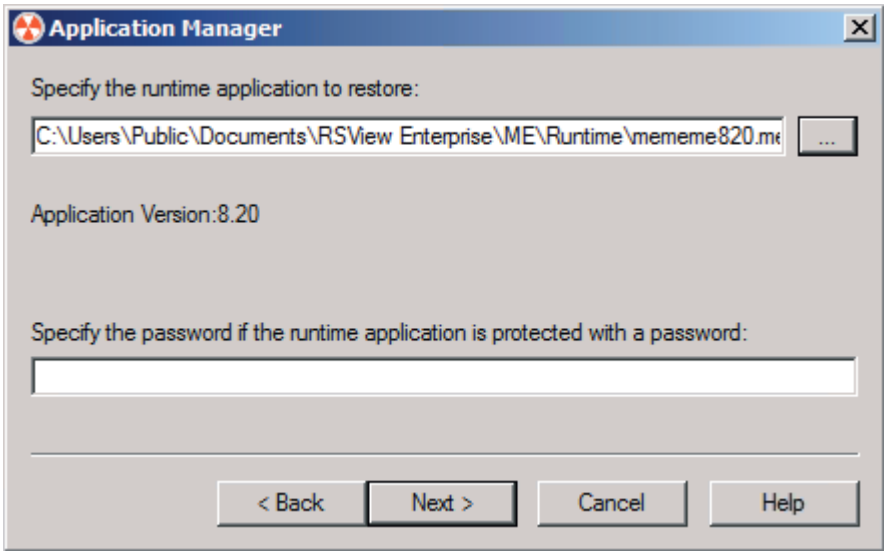
FactoryTalk Linx device shortcuts that were edited at runtime are restored when you convert the runtime application to a development application. However, the **Runtime** tab in the **FactoryTalk Linx Communication Setup** editor might

not show the device configuration for the runtime application. Make sure the **Runtime** tab shows the correct device configuration before making any changes to the shortcuts.

Tip: FactoryTalk Linx device shortcuts that were valid on the runtime terminal might not exist or might be invalid on the development computer. Review all device and tag file shortcut assignments in the **FactoryTalk Linx Communication Setup** editor before you edit, test, or transfer the application.

To convert a runtime application to a development application

1. In FactoryTalk View Studio, from the **Tools** menu, select **Application Manager.**
2. Select **Restore runtime application** and then click **Next**.



3. Specify the folder and file name of the runtime application to convert.
4. If you created a password for the file when you created the runtime application, type the password.
5. Click **Next**.
6. Type a name for the converted application, and then click **Finish**.

Run applications on a personal computer

This chapter describes how to:

- Install printers and software on the runtime computer
- Move applications to the runtime computer
- Start FactoryTalk View ME Station
- Load and run applications
- Shut down applications
- Edit device short cuts
- Look up contact information for technical support
- Set up FactoryTalk Diagnostics at runtime
- Set up serial ports for KEPServer Enterprise
- Set up FactoryTalk Linx communication drivers
- Specify the printers to use at runtime
- Specify startup options for FactoryTalk View ME Station
- Delete log files on the runtime computer
- Specify the time, date, and number formats to use at runtime
- Use the DeskLock tool

Run an application on a personal computer

Before you can run an application on a personal computer, complete the following steps.

- Install the necessary hardware and software on the runtime computer.
- Transfer your application to the runtime computer.
- Set up options in FactoryTalk View ME Station.

For information about installing FactoryTalk View ME Station, see *FactoryTalk View ME Installation Guide*.

Install printers and software on the runtime computer

Install printers and software on the runtime computer before you can run an application.

To install printers and software on the runtime computer

1. If you will be printing displays, alarms, or diagnostics messages, set up printer connections on the runtime computer.
   For more information, see Specify the printers to use at runtime on page 191.
2. If you are using FactoryTalk Linx, set up communications as described in Set up communications on page 65.
3. If you are using an OPC server other than FactoryTalk Linx, install the OPC server software on the runtime computer or on another computer on the network.
   For installation information, see the documentation supplied by your OPC server vendor. For more information about OPC, see Set up communications on page 65.
4. If your application uses third-party ActiveX objects, install and register the Windows computer version of the objects on the runtime computer.
   For information about ActiveX objects, see ActiveX Objects on page 290.
5. Install on the runtime computer all languages used by the runtime application.

6. If the runtime computer uses different time, date, or number formats than the development computer, specify the time, date, and number formats to use at runtime.
   For more information, see Specify time, date, and number formats on page 197.
7. If desired, use the DeskLock tool to prevent users from switching to another software application or using the Windows desktop at runtime.
   For more information, see Use the DeskLock tool on page 198.

Move applications to the runtime computer

Use either of the following methods to copy your runtime application (.mer) from the development computer to the runtime computer:

- Copy the application file from the development computer to removable media such as USB thumb drive or USB hard drive, and then from the removable media to the runtime computer.
- If the development and runtime computers are on the same network, use Windows Explorer or My Computer to move the file.

For information about creating the runtime application file, see Create runtime applications on page 177.

Set up options in FactoryTalk View ME Station

Use the following steps to set up options in FactoryTalk View ME Station.

To set up options in FactoryTalk View ME Station

1. On the runtime computer, start FactoryTalk View ME Station.
   For more information, see Start FactoryTalk View ME Station on the development computer on page 185.
2. Load the application.
   For more information, see Load and run applications on page 185.
3. Edit device shortcuts, if necessary.
   For more information, see Edit device shortcuts on page 187.
4. Set up FactoryTalk Diagnostics on the runtime computer (if you have not already done so).
   For more information, see Use the Diagnostics Setup tool on page 129.
5. (optional) Set up pop-up input colors.
   For more information, see Set up pop-up input colors on page 188.
6. If you will be using KEPServer Enterprise, specify serial port IDs.
   For more information, see Set up serial ports for use with KEPServer Enterprise on page 189.
7. If you will be using FactoryTalk Linx, set up communication drivers (if you have not already done so).
   For more information, see Set up FactoryTalk Linx communication drivers on page 190.
8. Specify the printers to use.
   For more information, see Specify the printers to use at runtime on page 191.
9. Specify startup options for FactoryTalk View ME Station.
   For more information, see Specify startup options for FactoryTalk View ME Station on page 192.

Once you have completed these steps, you're ready to run the application. For information about running your application, see Use your application on page 205.

Start FactoryTalk View ME Station on the development computer

Start FactoryTalk View ME Station on the development computer to run an application.

To start FactoryTalk View ME Station on a development computer, use one of these methods:

- From Windows **Start** menu, select **All apps > Rockwell Software > FactoryTalk View ME Station**.
- On the desktop, double-click the shortcut icon **FTView ME Station**.
- In the Windows Explorer, right-click **MERuntime.exe** and select **Run as administrator**. Default location for this file is **\Program Files (x86)\Rockwell Software\RSView Enterprise\MERuntime.exe**
- From Windows **Start** menu, select **Run**. In the **Open** box, type the path to the MERuntime.exe file.

It is recommended that you exit FactoryTalk View Studio before starting FactoryTalk View ME Station.

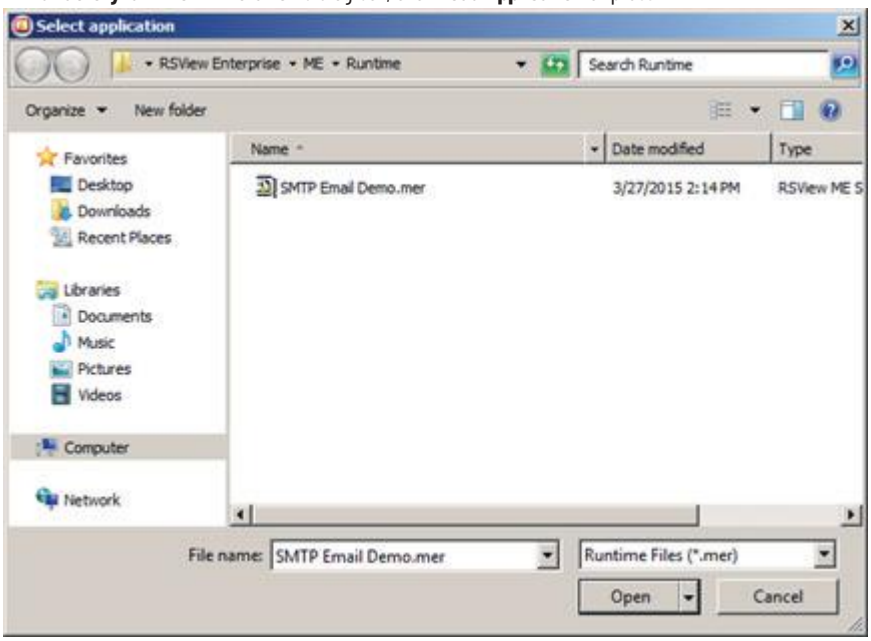
For information about specifying startup options for FactoryTalk View ME Station, see Specify startup options for FactoryTalk View ME Station on page 192.

Load and run applications

You can run any runtime application (.mer) that is on the runtime computer.

To load and run the application

1. In the **FactoryTalk View ME Station** dialog box, click **Load Application** or press **F1**.



2. Navigate to the folder containing the application's .mer file and then click the file name.
3. Click **Open**.
4. To replace the runtime computer's communication settings with the application's communication settings, click **Yes**. To keep the runtime computer's communication settings, click **No**.

Tip: To restore the previous communication settings, rename Backup*.xml to RSLinxNG.xml in C:\ProgramData\Rockwell\RSLinx Enterprise, and restart the computer.

You are notified that the application's FactoryTalk System Directory of users and security policies will be loaded on this computer. This is the set of users and policies that have been set up for the application and are contained in the .mer file.

The computer's existing FactoryTalk System Directory will be archived, and will be restored when you stop the application. To turn off this warning, see Turn off the FactoryTalk Directory Server warning on page 197.

5. To continue, click **Yes**. To stop loading the application, click **No**.

   If you continue, the application name is shown in the **Current application** box in the **FactoryTalk View ME Station** dialog box.

6. To run the application, click **Run Application**.

   The DEFAULT user is logged in. If a macro is assigned to the DEFAULT user, the macro runs.

Shut down an application

You might want to shut down an application.

To shut down an application, use one of these methods

- Press a shutdown button on a graphic display.
- If the application is set up to use a title bar with a Control box, click the **Close** button at the right end of the title bar.
- If the application is set up to use a title bar with a **Control** box, on the **Control** menu at the left end of the title bar, click **Close**.

Item	Name
1	Control box
2	Control menu
3	Close button

For information about using a title bar in graphic displays, see Title bar on page 61.

For information about preventing unauthorized users from shutting down applications, see Application examples on page 152.

When the application shuts down:

If you assigned a shutdown macro (in the **Startup** editor), the macro runs, assigning values to tags, and then the application stops. For information about the **Startup** editor, see Create runtime applications on page 177.

- FactoryTalk View ME Station closes.
- The computer's FactoryTalk System Directory is restored.

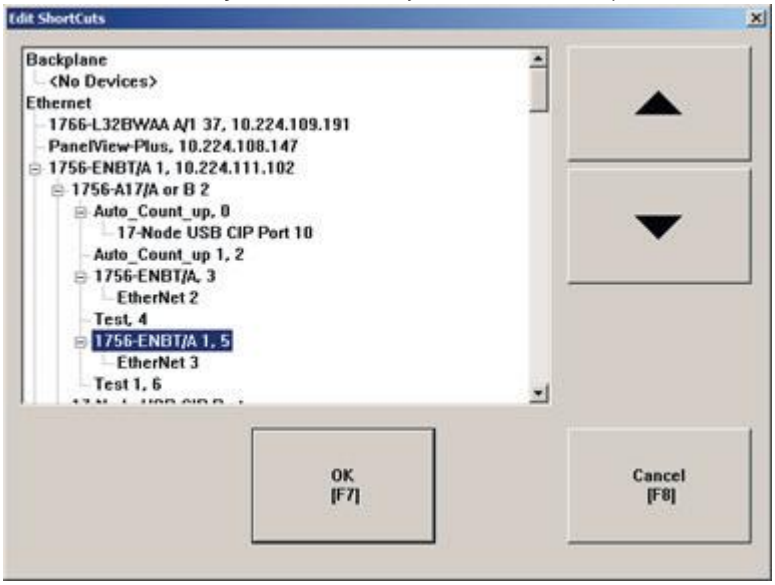
Edit device shortcuts

You can use FactoryTalk View ME Station to edit device shortcuts that have been set up in the application.

Before editing device shortcuts, load the application containing the device shortcut, as described on Load and run applications on page 185.

To edit device shortcuts

1. In the **FactoryTalk View ME Station** dialog box, click **Application Settings**.
2. Double-click **Device Shortcuts**.
3. Double-click the name of the shortcut to edit it.
4. In the **Edit ShortCuts** dialog box, click the device you want the shortcut to point to and click **OK**.



5. Click **Close**.

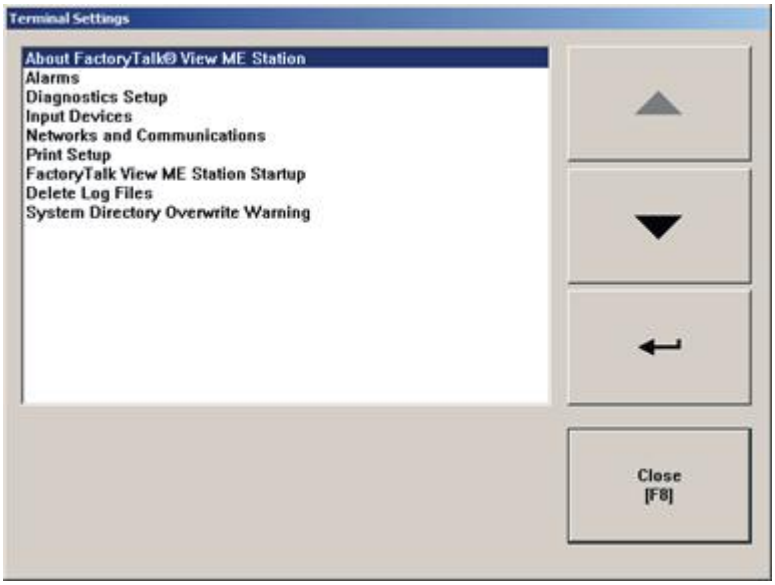
A message will prompt you to save the FactoryTalk Linx configuration to the runtime application file (.mer). Click **Yes** to save the change to the runtime application file. When you load the application next time, for example, after a power-cycle, the application will use the updated shortcut configuration.

Look up contact information for technical support

FactoryTalk View ME Station provides technical support contact information.

To look up technical support contact information

1. In FactoryTalk View ME Station, click **Terminal Settings**.



2. Double-click **About FactoryTalk View ME Station**.
3. Click **Technical Support**.

   The telephone number and URL for technical support are shown.

Set up FactoryTalk Diagnostics on the runtime computer

You can set up FactoryTalk Diagnostics on the runtime computer using the **FactoryTalk View ME Station** dialog box.

To set up FactoryTalk Diagnostics on the runtime computer

1. In FactoryTalk View ME Station, click **Terminal Settings**.
2. Double-click **Diagnostics Setup**.

   The **FactoryTalk Diagnostics Setup** dialog box opens.
3. Set up FactoryTalk Diagnostics, as described in Use the Diagnostics Setup tool on page 129.

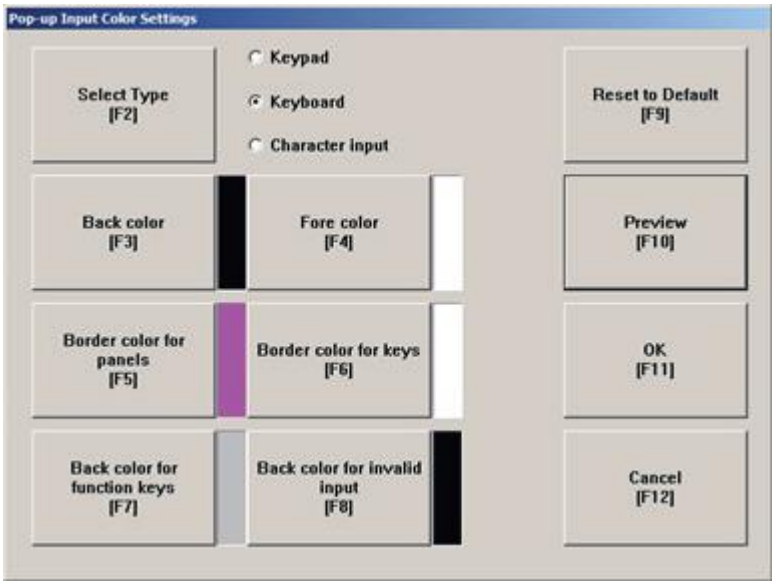
Set up pop-up input colors

You can set up pop-up input colors using the **FactoryTalk View ME Station** dialog box.

To set up pop-up input colors

1. In FactoryTalk View ME Station, click **Terminal Settings**.
2. Double-click **Input Devices**.

3. Double-click **Pop-up Input Color Settings**. The **Pop-up Input Color Settings** dialog box opens.



4. Click **Select Type** to select an input type:
   - **Keypad**
   - **Keyboard**
   - **Character input**

   Tip: The keyboard and keypad color settings also apply for scratchpad.

5. Click the following buttons to set up the colors:
   - **Back color**
   - **Fore color**
   - **Border color for panels**
   - **Border color for keys**
   - **Back color for function keys** (only for keyboard)
   - **Back color for invalid input**
6. (Optional) Click **Preview** to preview the color settings.
7. (Optional) Click **Default** to reset the colors to the default style.
8. Click **OK** to save the color settings.

Set up serial ports for use with KEPServer Enterprise

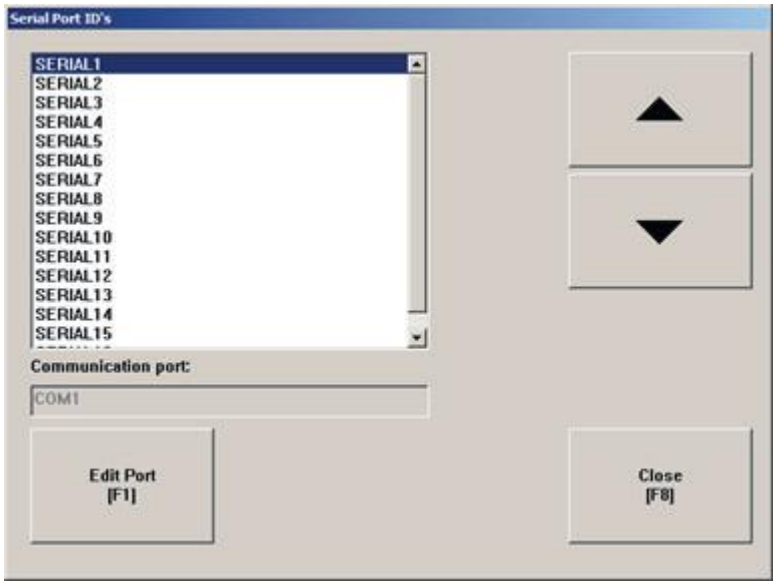
If you plan to use KEPServer Enterprise and serial communications, you must specify which COM port to use.

For information about setting up communications in KEPServer Enterprise, see KEPServer Enterprise Help.

To specify the COM port to use for serial communications

1. In the **FactoryTalk View ME Station** dialog box, click **Terminal Settings**.
2. Double-click **Networks and Communications**.

3. Double-click **KEPServer Serial Port ID's**.



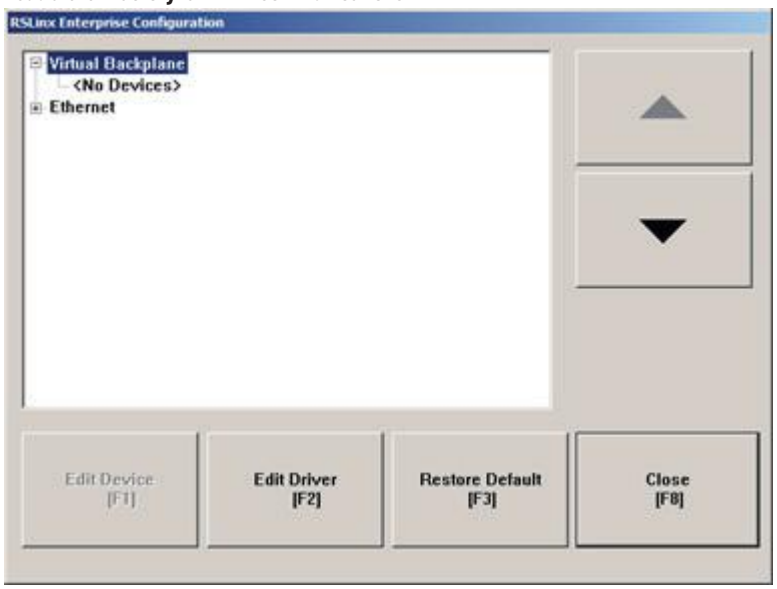
4. In the **KEPWare Serial Port ID's** dialog box, select the serial port ID you specified when you set up the KEPServer Enterprise channel.
5. Click **Edit Port.**
6. In the **Communication Ports** dialog box, select the COM port to use for KEPServer Enterprise communications.

Set up FactoryTalk Linx communication drivers

Use FactoryTalk Linx to set up communication drivers for your runtime application. You can set up the drivers directly in FactoryTalk Linx, or open FactoryTalk Linx by using the **FactoryTalk View ME Station** dialog box.

To set up the FactoryTalk Linx communication driver to use at runtime

1. In the **FactoryTalk View ME Station** dialog box, click **Terminal Settings**.
2. Double-click **Networks and Communications**.
3. Double-click **FactoryTalk Linx Communications**.



gialog box

4. To edit a driver, select it and then click **Edit Driver**.

5. To edit a device, select it and then click **Edit Device**.
6. When you are finished, click **Close**.

If you loaded an application, a message will prompt you to save the FactoryTalk Linx configuration to the runtime application file (.mer). Click **Yes** to save the change to the runtime application file. When you load the application next time, for example, after a power-cycle, the application will use the updated FactoryTalk Linx configuration.

For information about setting up FactoryTalk Linx drivers and devices, see the FactoryTalk Linx documentation.

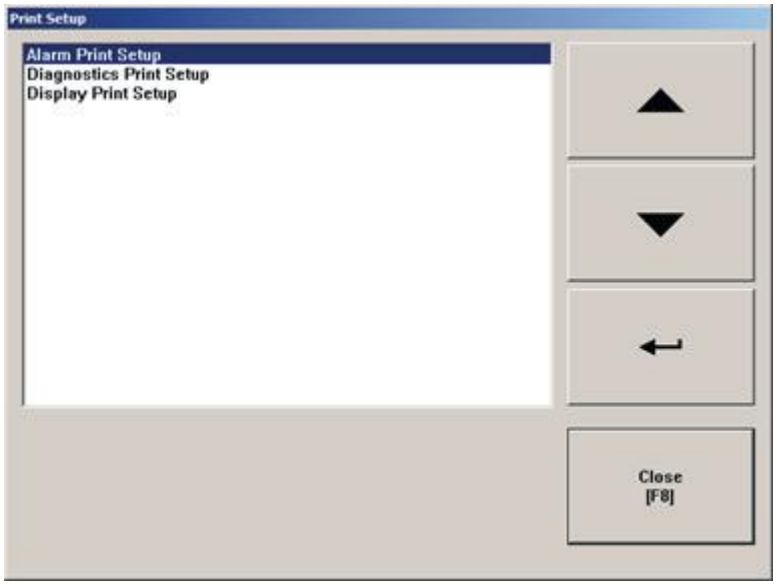
Once the driver is set up, FactoryTalk View ME Station automatically starts the driver software when you run the application.

Specify the printers to use at runtime

You can use local or network printers to print alarm messages, reports, diagnostics messages, and graphic displays at runtime. If desired, you can use a different printer for each type of printing.

To specify the printers to use at runtime

1. In the **FactoryTalk View ME Station** dialog box, click **Terminal Settings**.
2. Double-click **Print Setup**.



3. Double-click the type of printing to set up.

4. Modify the print options as required.

For detailed information about printer options, refer to your Windows documentation.

Specify startup options for FactoryTalk View ME Station

This section describes how to start FactoryTalk View ME Station automatically when Windows starts, and describes startup options for FactoryTalk View ME Station.

When you start FactoryTalk View ME Station, you can:

- Run an application
- Load an application
- Automatically delete an application's log files before running the application
- Replace FactoryTalk Linx communications on the runtime computer with the application's settings

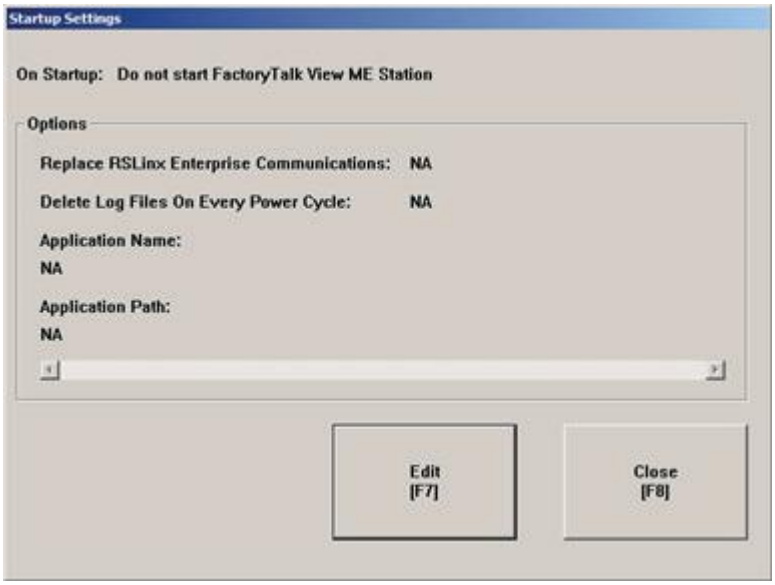
By default, the option to start FactoryTalk View ME Station when Windows starts is turned off. The settings in this section apply only if you want FactoryTalk View ME Station to start automatically when Windows starts.

Start FactoryTalk View ME Station and run an application when Windows starts

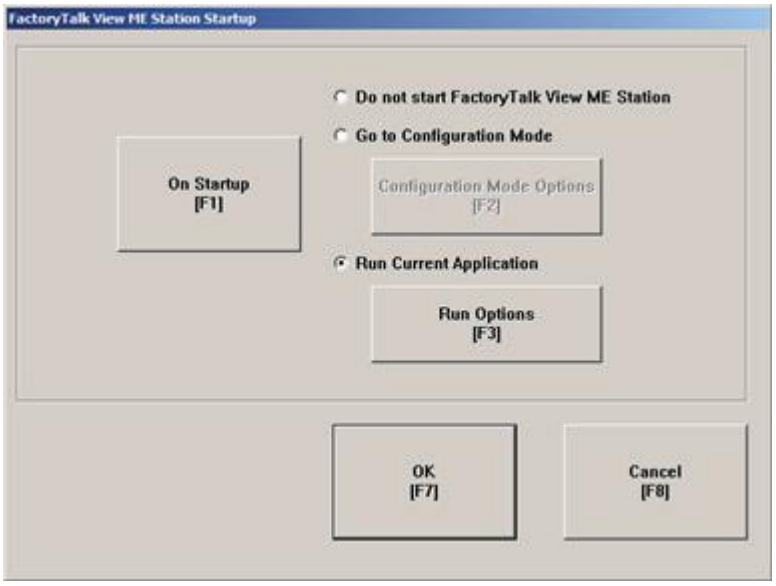
You can start FactoryTalk View ME Station and run an application when Windows starts.

To start FactoryTalk View ME Station and run an application when Windows starts

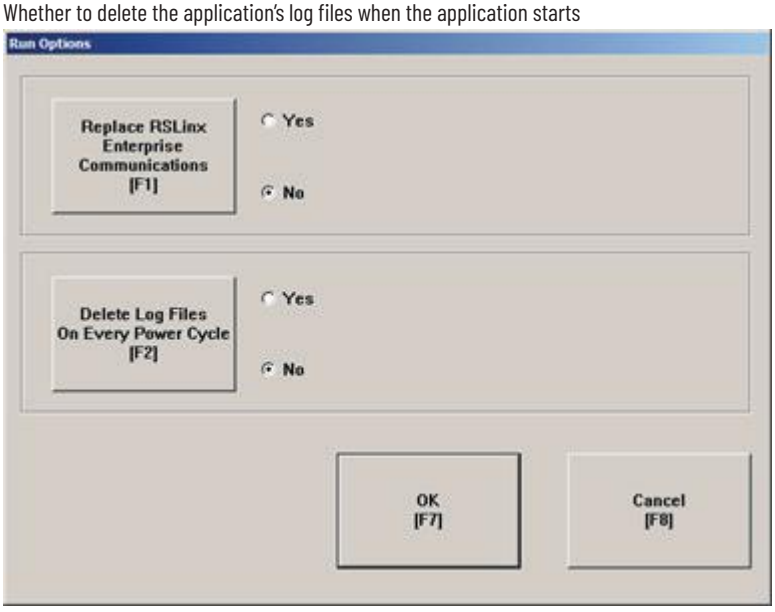
1. In the **FactoryTalk View ME Station** dialog box, load the application that you want to run.
   For details, see Load and run applications on page 185.
2. Click **Terminal Settings**.
3. In the **Terminal Settings** dialog box, double-click **FactoryTalk View ME Station Startup**.
4. In the **Startup Settings** dialog box, click **Edit**.



5. Select the **Run Current Application** option.

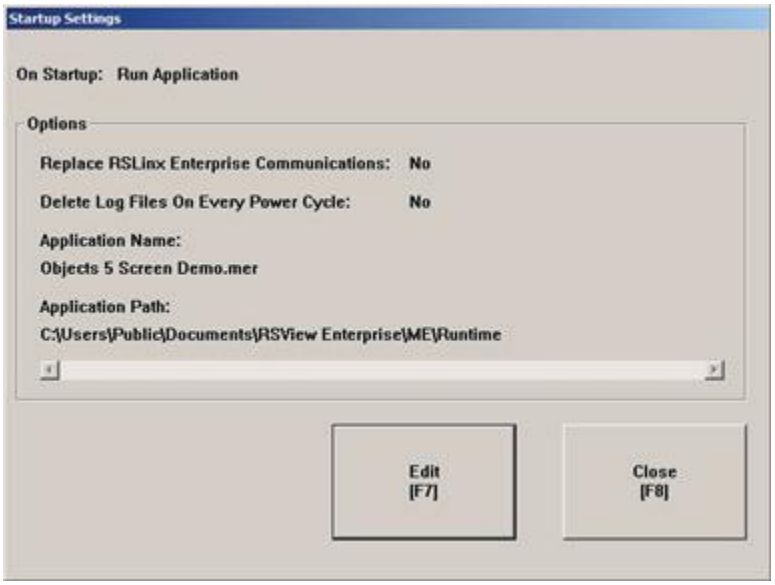


6. Click **Run Options.**
7. In the **Run Options** dialog box, specify the following:
   - Whether to replace the FactoryTalk Linx communication settings on the runtime computer with the application's settings when the application starts
   - Whether to delete the application's log files when the application starts



8. Click **OK** to save the changes.

The **Startup Settings** dialog box shows the options that you specified.

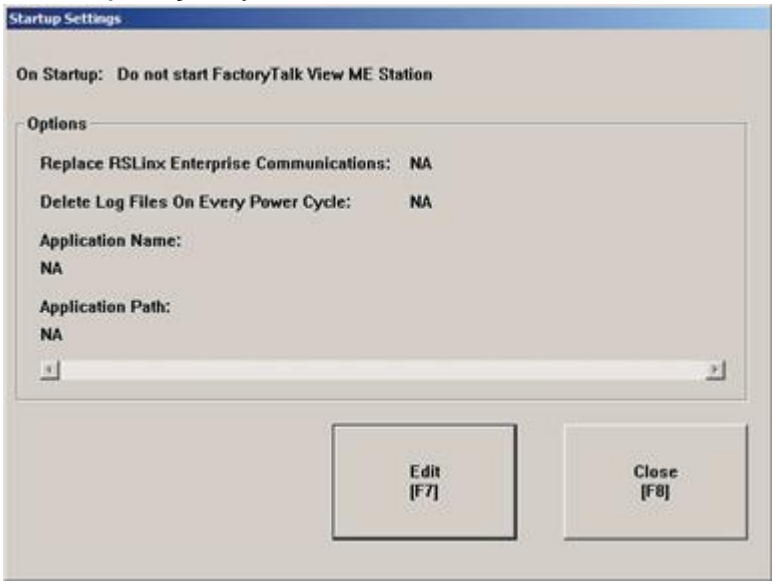


Start FactoryTalk View ME Station without running an application when Windows starts

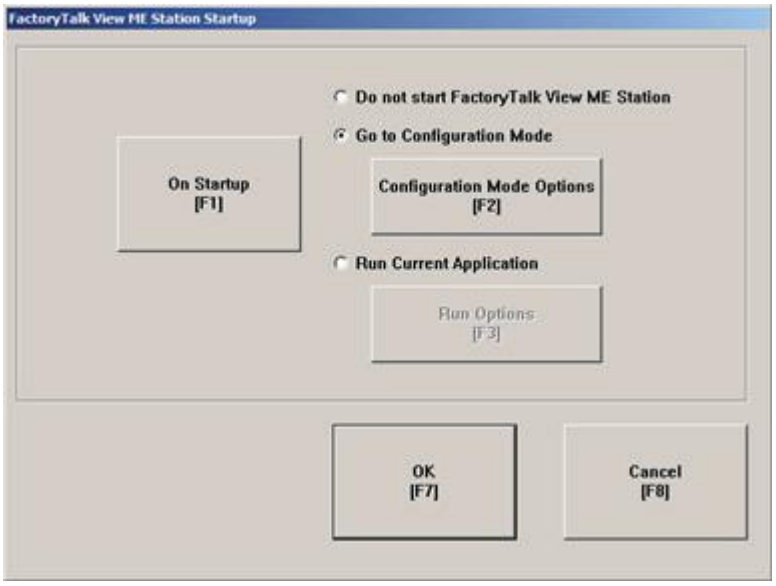
You can also start FactoryTalk View ME Station without running an application when Windows starts.

To start FactoryTalk View ME Station without running an application when Windows starts

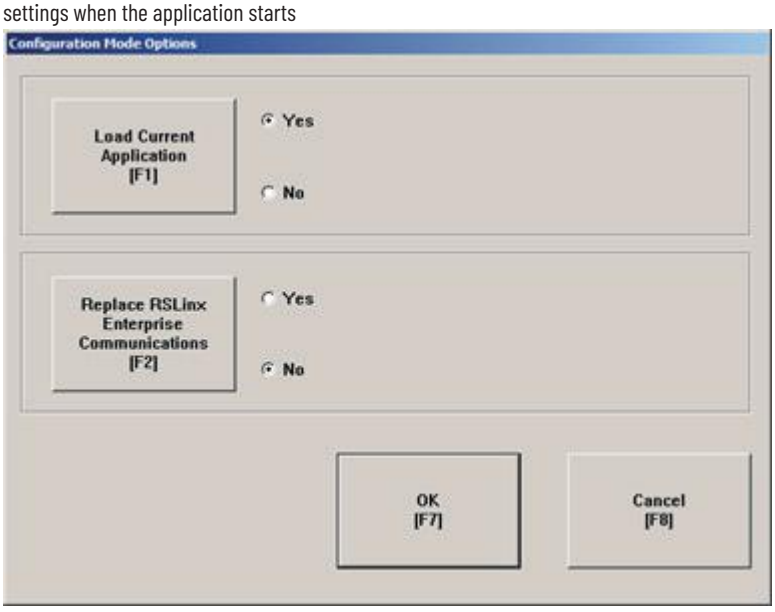
1. In the **FactoryTalk View ME Station** dialog box, click **Terminal Settings**.
2. In the **Terminal Settings** dialog box, double-click **FactoryTalk View ME Station Startup.**
3. In the **Startup Settings** dialog box, click **Edit**.



4. Select the **Go to Configuration Mode** option.

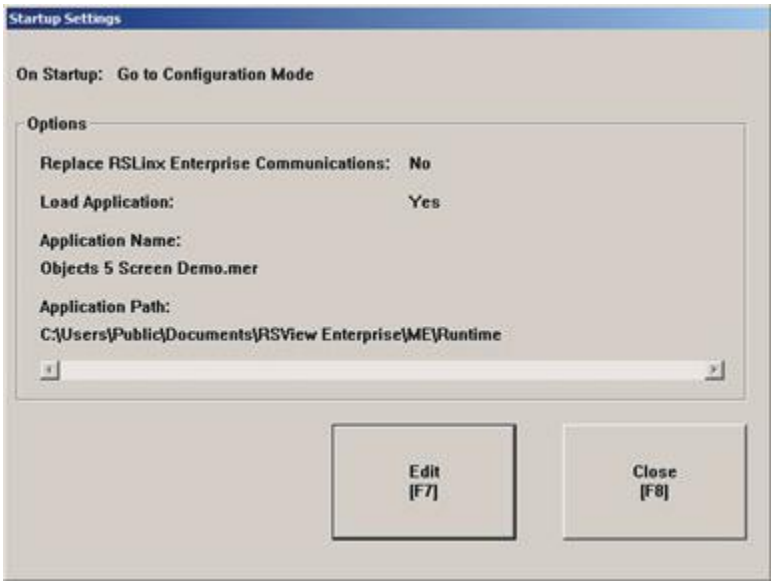


5. If you did not load an application, go to step 8.
6. If you loaded an application, click **Configuration Mode Options** and specify whether to:
   - Load the current application when FactoryTalk View ME Station starts
   - Replace the FactoryTalk Linx communication settings on the runtime computer with the application's settings when the application starts



7. Click **OK** to save the changes and close the **Configuration Mode Options** dialog box.
8. Click **OK** to save the changes.

The **Startup Settings** dialog box shows the options that you specified.



Use Windows operating systems with ME Station

To deploy an HMI system on a Windows computer, select one of the following options:

- The operator will need to log into Windows as a member of the Administrators group. The operator can then use ME Station for Windows.
- An administrator can disable the UAC. This allows a Windows user that is a member of the Users group to use ME Station for Windows.
- An operator that is a member of the Users group who knows an Administrators password and can provide authentication when UAC prompts, can use ME Station.

This allows the operator to load and run an MER as well as configure ME Station for Windows using the Graphical User Interface.

Delete log files on the runtime computer

When you run your application, FactoryTalk View ME Station stores log files for alarms, audits, and data logging (if you use these features). When you start FactoryTalk View ME Station, you can delete the alarm, audit, and data log files for the loaded application, or for all the applications on the runtime computer.

IMPORTANT:

When FactoryTalk View ME Station is configured to delete log files on startup, it will delete alarm log file, audit log file, data log model, and diagnostics.

CSV data log files produced using the CSVDataLog and the DataStorePlus ActiveX controls will not be deleted.

Run a newer version of the application

If you run a newer version of an application, the alarm log file and audit log file for the older version is deleted automatically. The data log file for the older version is retained, to allow the display of historical data in trends.

For more information about the alarm log file, see Alarm log file on page 102.

For more information about data log files, see Change the data log model used at runtime on page 390.

Delete log files manually

To delete log files for the loaded application

- In the **FactoryTalk View ME Station** dialog box, click **Yes** beside the **Delete Log Files Before Running** button.

All alarm and data log files for the loaded application are deleted.

To delete log files for all applications on the runtime computer

1. In the **FactoryTalk View ME Station** dialog box, click **Terminal Settings.**
2. Double-click **Delete Log Files.**
3. Click **Yes**.

Log files for data log models that use the default path are deleted. All alarm log files are deleted.

Turn off the FactoryTalk Directory Server warning

When you load an application, you are notified that the application's FactoryTalk System Directory of users and security policies will be loaded on the computer. The computer's existing FactoryTalk System Directory is archived while you run the application. It is restored when you stop the application. **You can turn off this notification warning.**

To turn off the overwrite warning

1. In the **FactoryTalk View ME Station** dialog box, click **Terminal Settings.**
2. Double-click **System Directory Overwrite Warning.**
3. Click **No**.

Specify time, date, and number formats

Time, date, and number formats are used by these graphic objects:

- Numeric display
- Gauge
- Time and date display
- Trend
- Alarm banner
- Alarm list
- Audit trail list
- Audit trail detail

The time and date formats are also used when printing the alarm history report (using the **Print Alarm History** button). The numeric and time and date embedded variables also use number, time, and date formats.

All objects use the time, date, and number formats of the current application language. For example, if the application language uses a comma for the decimal symbol, numeric variables use a comma for the decimal symbol.

For information about using multiple languages, see Set up language switching on page 160.

DeskLock tool

Use the DeskLock tool to prevent users from switching to another software application or having access to the desktop at runtime.

This tool can have far-reaching effects on your operating system. The DeskLock tool can replace the standard Windows non-configured computer with a customized desktop. The customized desktop is intended to prevent operators from having access to other applications and operating system functions such as restarting Windows or shutting down tasks. You can also use the tool to set up an automatic Windows logon and to specify options such as which software applications are allowed or not allowed to run on the computer.

Before using the tool, read the DeskLock Help.

Open the DeskLock tool

Use the following steps to open the DeskLock tool.

To open the DeskLock tool

- From Windows **Start** menu, Select **All apps** > **Rockwell Software** > **DeskLock**.

For details about using the tool, see the tool's Help.

Transfer applications to a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal

This chapter describes how to:

- Start FactoryTalk View ME Station
- Set up communication drivers to transfer applications to a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal
- Transfer applications and Windows True Type font files to a terminal
- Upload applications from a terminal to the development computer
- Compare applications on the development computer with applications on the terminal

For information about using your application at runtime, see Use your application on page 205

For information about using FactoryTalk View ME Station on the terminal, including setting up communications, printers, and input devices, see the terminal's *User Manual*.

Transfer applications to a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal

Before you can run an application on a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, transfer the application to the terminal, install printers and software on the terminal, and then transfer the application to the terminal.

To transfer an application to a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal

1. Install printers and software on a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal.
2. Transfer the application to a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal.

Install printers and software on a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal

Before running an application on a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, install printers and software on the terminal.

To install printers and software on a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal

1. If you will be printing displays, alarms, reports, or diagnostics messages, connect a printer to the USB or Network port.
2. If you are using KEPServer Enterprise, enable the driver you need on the terminal. For more information, see KEPServer Enterprise documentation.
   For information about OPC data servers, see Set up communications on page 63.
3. If your application uses third-party ActiveX objects, install and register the Windows CE version of the objects on the terminal.
   For information about ActiveX objects, see ActiveX objects on page 290.

Install printers on a MobileView terminal

The steps of installing printers on a MobileView terminal are similar with the steps on a computer. For more information, see Specify the printers to use at runtime on page 191.

Install printers on a PanelView Plus 7 or PanelView Plus 6 terminal

PanelView Plus 7 and PanelView Plus 6 terminals incorporate the JETCET PRINT solution for Microsoft Windows CE devices, supporting select Canon, Epson, Hewlett-Packard, and Brother printers. Plug-and-play installation is supported for these printers as well as manual installation if a suitable JETCET print driver is available.

Print support is centrally managed and independent of the application. You can add and configure a new printer from the desktop control panel. Printer configurations include the printer name, printer driver, port, and port-specific parameters. New configurations are retained through a power cycle.

PanelView Plus 7 and PanelView Plus 6 terminals support a wide range of printers. For a list of supported printers, see Knowledgebase Document ID: QA14804 - PanelView Plus 6 Printer Support and Configuration.

Install printers on a PanelView Plus terminal

PanelView Plus terminals support printing using the Microsoft Windows CE PCL3 printer driver, which is already installed on the terminal. Printing is supported for most laser and ink-jet printers. If you have problems printing, check that your printer is compatible with the PCL3 printer driver.

Transfer an application to a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal

Before running an application on a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, transfer the application to the terminal.

To transfer an application to a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal

1. On the MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, start FactoryTalk View ME Station.
   For more information, see Start FactoryTalk View ME Station on a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal on page 201.
2. In FactoryTalk View ME Station, if you are transferring via a serial connection, specify and set up the serial driver. If you are using Ethernet for the transfer, you can skip this step.
   For more information, see Specify the driver to use for the transfer on page 202.
3. On the development computer, if you are transferring via a serial connection, set up the FactoryTalk Linx driver to use for the transfer. If you are using Ethernet for the transfer, you can skip this step.
   For more information, see Specify the driver to use for the transfer on page 202.
4. Move the application to the MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal using ME Transfer Utility.
   For more information, see Download applications and Windows True Type fonts on page 203.

You can also transfer applications to the terminal using a flash card. For information about this method, see the terminal's *User Manual*.

Start FactoryTalk View ME Station on a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal

On a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, if FactoryTalk View ME Station does not start automatically when the terminal starts, you can start FactoryTalk View ME Station manually.

For information about specifying startup options for FactoryTalk View ME Station, see Specify startup options for FactoryTalk View ME Station on page 192.

Start FactoryTalk View ME Station on a PanelView Plus 7, PanelView Plus 6, PanelView Plus, or PanelView Plus CE terminal

You might want to manually start FactoryTalk View ME Station on a PanelView Plus 7, PanelView Plus 6, PanelView Plus, or PanelView Plus CE terminal.

To start FactoryTalk View ME Station on a PanelView Plus 7, PanelView Plus 6, PanelView Plus, or PanelView Plus CE terminal, do one of the following:

- On the desktop, double-click the shortcut icon **FTView ME Station**.
- In the Windows Explorer, double-click MERuntime.exe. Default location for this file is:
  (PanelView Plus 7 and PanelView Plus 6) **\Windows**
  (PanelView Plus and PanelView Plus CE) **\Storage Card\Rockwell Software\RSViewME**
- If Windows **Start** menu is available, from the **Start** menu, select **Run**. In the **Open** box, type the path to the MERuntime.exe file.
- If Windows **Start** menu is available, from the **Start** menu, select **Programs > Rockwell Software > FactoryTalk View ME Station**.

Start FactoryTalk View ME Station on a MobileView terminal

You might want to manually start FactoryTalk View ME Station on a MobileView terminal.

To start FactoryTalk View ME Station on a MobileView terminal, do one of the following

- On the desktop, press the **FTView ME Station** icon
- In **File Explorer**, right-click **MERuntime.exe** and select **Run as administrator**. The default location for the file is **C:\Program Files (x86)\Rockwell Software\RSView Enterprise\MERuntime.exe**

Specify the driver to use for the transfer on a terminal

Use one of these drivers for the transfer:

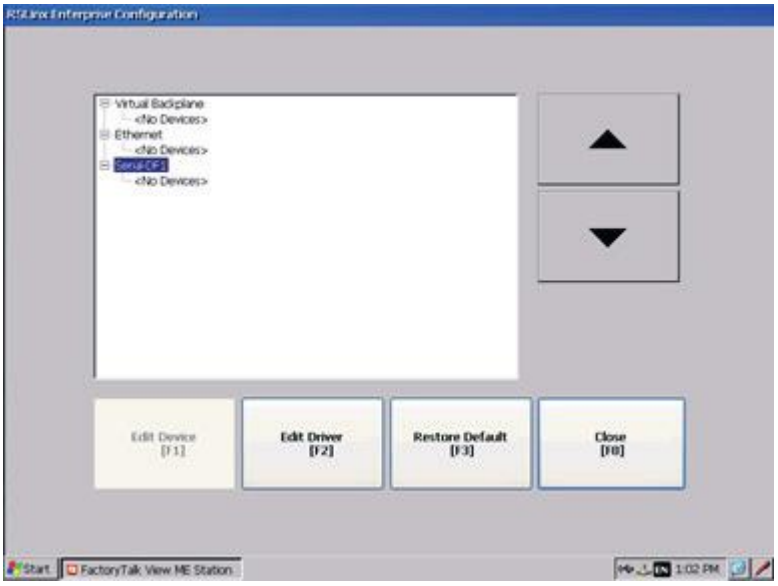
- Ethernet — An Ethernet connection between the development computer and the MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal destination. If you are using an Ethernet connection for the transfer, you do not need to set up a driver for the transfer since the Ethernet driver is selected and loaded by default.
- Serial-DF1 — A direct serial connection between the development computer and the PanelView Plus 6, PanelView Plus, PanelView Plus Compact, and PanelView Plus CE terminal destination. If you are using a serial connection for the transfer, specify and set up a driver.

For information about setting up drivers in FactoryTalk Linx, see FactoryTalk Linx Help.

You can also transfer applications to the terminal using a Flash Card. For information about this method, see the terminal's User Manual.

To specify and set up a serial driver for the transfer

1. In the **FactoryTalk View ME Station** dialog box, click **Terminal Settings**.
2. Double-click **Networks and Communications**.
3. In the **Networks and Communications** dialog box, double-click **FactoryTalk Linx Communications**.



4. Click the Serial-DF1 driver, and then click **Edit Driver**.
5. Select **Use Auto-Configuration**, and then click **Edit**.
6. In the dialog box that opens, select **Yes**, and then click **OK**.
   Auto configuration works with most devices. If auto configuration is not successful for your device, return to these steps, select **No**, and continue to step 7.
7. Set up the driver, and then click **OK**. If you are using auto configuration, skip this step.

Set up a driver for the transfer on the development computer

On the development computer, set up the FactoryTalk Linx driver that you will use to download your application to the MobileView or PanelView Plus family terminal.

Use one of these drivers for the transfer:

- **Serial-DF1** — Use for a serial transfer.
- **Ethernet** — Use with an Ethernet connection.

For information about setting up drivers in FactoryTalk Linx, see FactoryTalk Linx Help.

Start ME Transfer Utility on the development computer

On the development computer, do one of the following:

- From Windows **Start** menu, select **All apps > Rockwell Software > ME Transfer Utility**.
- In FactoryTalk View Studio, from the **Tools** menu, select **Transfer Utility**.
- In FactoryTalk View Studio, on the toolbar, click  .

For details about using the utility, see Help for ME Transfer Utility.

Download applications and Windows True Type fonts

Use ME Transfer Utility to:

- Download a runtime application file (.mer) or True Type font file (.ttf or .ttc) from the development computer (source) to a PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal (destination). A .ttf file contains a single font. A .ttc file contains a collection of fonts.
- Download a runtime application file (.mer) from the development computer (source) to a MobileView terminal (destination).

For information about creating the runtime application file, see Create runtime applications on page 177.

You can download:

- Multiple applications (one at a time) to the same terminal
- Applications while another application is running on the terminal
- A new copy of the application, which is running on the terminal, and use the new application next time when you run the application on the terminal

If the download process fails or is canceled, the application file will be deleted from the destination directories. Make a copy of the application file before you begin and make sure that there is enough storage space on the destination computer.

To perform a serial download to a PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, make sure that the terminal is connected to the development computer using the Allen-Bradley© serial cable 2711-NC13.

Upload applications from the MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal

Use ME Transfer Utility to upload a runtime application file (.mer) from the MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal to the development computer.

Any password changes that were made while the application was running are saved in the .mer file and will be uploaded.

If the upload process fails or is canceled, the application file will be deleted from the destination directory. Make sure that there is enough storage space on the destination computer.

You can upload while an application is running on the runtime computer. You can upload an existing copy of the application currently running on the runtime computer.

To perform a serial upload from a PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, make sure that the terminal is connected to the development computer using the Allen-Bradley serial cable 2711-NC13.

Compare applications

You can also use the ME Transfer Utility tool to compare an application on the development computer with an application on the terminal. The comparison tool can tell you whether the files are identical or different. The comparison tool detects differences using a binary checksum. It is not possible to list individual differences between the files.

Use your application

This chapter describes how to:

- Log in and log out
- Change passwords
- Add a user or group and delete a user or group
- Modify group membership
- Unlock users
- Enable and disable users
- Change user properties
- Enter numeric and string values
- Change tag values
- Use electronic signature
- View tag data
- View alarms and messages
- View information about runtime communication errors
- Change languages

For information about navigating between displays, see Set up display navigation on page 171.

For information about navigating between and using the graphic objects in a display, see Set up how objects are used at runtime on page 328.

Log in to the application

An application's security configuration may contain FactoryTalk Security users or Windows-linked users. Windows-linked user credentials are contained on either the local PC or in a domain. To support user authentication in a Windows domain, additional PanelView Plus terminal configuration is required.

Domain authentication

FactoryTalk View ME 8.0 and later supports PanelView Plus 7 and PanelView Plus 6 domain authentication against Windows Server 2019, Windows Server 2016, and Windows Server 2022 domains in the following mechanism:

- LDAP (Lightweight Directory Access Protocol)
- LDAP over SSL (Secure Sockets Layer)

For details, see Configure FactoryTalk View ME 8.0 and later terminals for Domain authentication on page 205.

FactoryTalk View ME releases prior to 8.0 support domain authentication against Windows Server 2000 and Windows Server 2003 domains. For details, see Configure FactoryTalk View ME prior to 8.0 terminals for Domain authentication on page 209.

Configure FactoryTalk View ME 8.0 and later terminals for Domain authentication

You might want to configure LDAP settings on a FactoryTalk View ME 8.0 or later terminal.

To configure LDAP settings on a FactoryTalk View ME 8.0 or later terminal

1. In the **FactoryTalk View ME Station** dialog box, click **Terminal Settings**
2. Double-click **Networks and Communications**.
3. Double-click **LDAP Configuration.**
4. On FactoryTalk View ME 8.1 or later terminals, select one of the following **Authentication Mode** options:
   - **Use Windows and LDAP only authentication** to use Windows authentication mode to log in first. If Windows authentication fails, LDAP without SSL authentication mode will be used to log in.
   - **Use LDAP only authentication** to use LDAP without SSL authentication mode.
   - **Use LDAP over SSL authentication** to use LDAP over SSL authentication mode.
5. On FactoryTalk View ME 8.0 terminals, select one of the following **Use SSL** options:
   - **Yes** to use LDAP over SSL authentication mode.
   - **No** use LDAP without SSL authentication mode.
6. Do one of the following:
   - In the **Server FQDN** box, enter FQDN (fully qualified domain name), for example, *FTV1TT45.domain08r2.domain03.com.*

     Tip: FQDN is a DNS name that uniquely identifies the computer on the network. By default, it is a concatenation of the host name, the primary DNS suffix, and a period.
     You can find the FQDN from the **Network Identification** tab of the properties page for **My Computer** on the Domain server.
     If SSL is enabled, you must have the root certificate on your local terminal. For more information about obtaining the root certificate, see To obtain a root certificate on page 207.

   - In the **DNS-Style Domain Name** box, enter the domain name, for example, *domain08r2.domain03.com*. The **Port** field is automatically filled.
7. On FactoryTalk View ME 8.1 or later terminals, select one of the following **Group Membership** options:
   - **Use user attribute (faster)** - Security subsystem will only evaluate domain group membership against the domain group(s) to which the user explicitly belongs. Nested domain group membership will not be evaluated. This provides much faster domain authentication than the **Use nested groups** option. When using this option, it is critical that the domain group(s) to which the target user community belongs is(are) added to FactoryTalk Security explicitly, or added to a default FactoryTalk group.
   - **Use nested groups (slower)** - Security subsystem will evaluate domain group membership by querying the domain. The domain query returns a list of all the groups to which the user belongs. User *memberof* attribute will not be evaluated. This provides slower but more complete evaluation of a user's domain group membership. When using this option, it is recommended that the domain group(s) to which the target user community belongs is(are) added to FactoryTalk Security explicitly, or added to a default FactoryTalk group.
     For example, User A is a member of domain group Line1, and User A *memberof* attribute explicitly lists Line1. User B is a member of domain group Shift3, and User B memberof attribute explicitly lists Shift3. Shift3 is a subgroup of Line1. The HMI designer adds the domain group Line1 to FactoryTalk Security,

and adds and configures the group in ME Runtime Security (A through P codes). On the PanelView Plus 7, or PanelView Plus 6 terminal:

- When the HMI designer selects **Use user attribute (faster)**, at runtime, User A can successfully authenticate; but User B cannot authenticate because domain group Shift3 is subgroup of Line1, and Shift3 was not explicitly added to the security configuration.
- When the HMI designer selects **Use nested groups (slower)**, at runtime, User A and User B can successfully authenticate. User B can successfully authenticate because domain group Shift3 is subgroup of Line1. The domain query will return Line1 and Shift3 when evaluating domain group membership for User B.

8. Click **OK** to save the changes and close the window.

Obtain a root certificate

In the Windows domain controller, if you use LDAP over SSL to do authentication, you need to have the root certificate on your local terminal. You can use Domain Controller Root Certificate utility to import the certificate from the remote domain server to your computer, and then transfer the certificate to the PanelView Plus 7 or PanelView Plus 6 terminal.

Tip: To correctly run Domain Controller Root Certificate utility, make sure .NET Framework version 3.5 or higher is already installed on your computer.

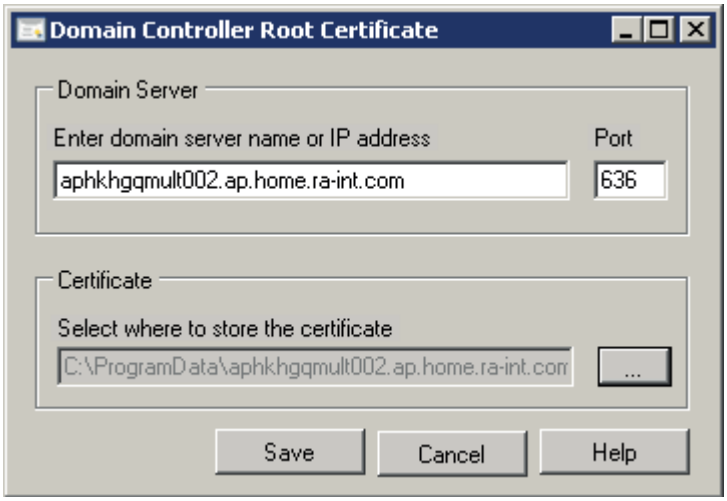
You can use the Domain Controller Root Certificate utility to obtain certificates only when the utility is installed on a computer that is in the domain.

Save a root certificate on a local computer

Use the Domain Controller Root Certificate tool to save a root certificate on a local computer.

To save a root certificate on a local computer

1. In FactoryTalk View Studio, from the **Tool** menu, select **Domain Certificate**. **Domain Controller Root Certificate** window opens.



2. In the **Domain Server** area, complete the following fields:
   - In the **Enter domain server name or IP address** field, enter the server name or IP address of the domain server. If the computer is already in a domain, the utility automatically detects and shows the domain server address in this field.
   - In the **Port** field, enter the port of the domain server for LDAP over SSL. The default value is 636.
3. In the **Certificate** area, click the **Browse** (...) button to specify where the certificate file will be stored. The default folder is **\ProgramData**.

   The default file name is *domainserver.cer*, where *domainserver* is the domain server name or IP address.
4. Click **Save** to complete and close the window.

Import a root certificate on a FactoryTalk View ME 8.0 or later terminal

You might want to import a root certificate on a FactoryTalk View ME 8.0 or later terminal.

To import a root certificate on a FactoryTalk View ME 8.0 or later terminal

1. Copy the certificate file to a removable device.
2. Plug the removable device, containing the certificate file, into the terminal.
3. On the terminal, select **Start > Settings > Control Panel > Certificates**.
4. In the **Certificates** dialog box, click **Import**.

5. In the **Import Certificate or Key** dialog box, select **From a File**, and click **OK**.

6. In the **Select a Certificate File** dialog box, select the certificate file, and click **OK**.

If the import is successful, the certificate file shows in the lists of the certificates in the **Certificates** window.

Configure FactoryTalk View ME prior to 8.0 terminals for Domain authentication

On the terminal

Make sure you have a **Primary DNS**, and **Primary WINS** defined in the PanelView Plus terminal

On the Windows 2003 domain controller

Navigate to **Start > All Programs > Administrative Tools > Domain Controller Security Policy > Local Policies > Security Options**, and set the following:

- Domain member: Digitally encrypt or sign secure data channel (always): DISABLED
- Domain member: Digitally encrypt secure data channel (when possible): ENABLED
- Domain member: Digitally sign secure data channel (when possible): ENABLED
- Microsoft network client: Digitally sign communications (always): DISABLED
- Microsoft network client: Digitally sign communications (if server agrees): ENABLED
- Microsoft network server: Digitally sign communications (always): DISABLED
- Microsoft network server: Digitally sign communications (if client agrees): ENABLED
- Network security: LAN Manager Authentication Level: Send LM & NTLM - use NTLM v2 session security if negotiated
- Network security: LDAP client signing requirements: Negotiate Signing

Access network resources from a terminal

To access network resources (printers, network shares), it is necessary to provide authentication credentials for the PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal. These credentials are independent of the currently logged in user in the HMI application.

To provide access to network resources when using a PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal

1. In the **FactoryTalk View ME Station** dialog box, click **Terminal Settings**.
2. Double-click **Networks and Communications**.
3. Double-click **Network Connections**.
4. Double-click **Network Identification**.
5. Type a user name, password, and domain name, and then click **OK**.

Log in to the application

Users log in using the **Login** button. The Login button opens a dialog box that allows the user to enter credentials.

To log in, both the **Runtime Security** and **FactoryTalk Security** configuration must contain the individual user account, or a group of which the user account is a member.

For information about setting up user accounts and passwords, see Set up security on page 143.

Tip:

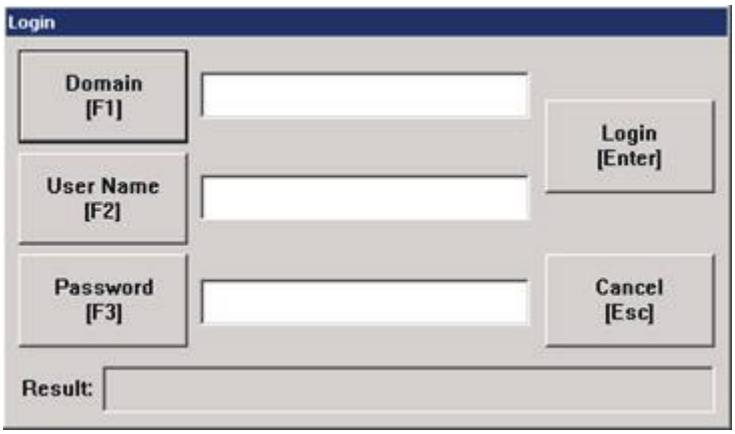
- When adding users and groups to FactoryTalk security, it is important to remember that a user cannot log in to the running application unless the user or group is configured in **Runtime Security**.
- To log in to a version 4.00 or later application, the **ME Runtime 4.00 and later** tab in **Runtime Security** configuration must contain either the individual user account, or a group of which the user is a member.
- To log in to a version 3.20 or earlier application, the **ME Runtime 3.20 and earlier** tab in **Runtime Security** configuration must contain either the individual user account, or a group of which the user is a member. If a user account is disabled, or locked, the user cannot successfully log in until the user account is enabled or unlocked

To log in to the application

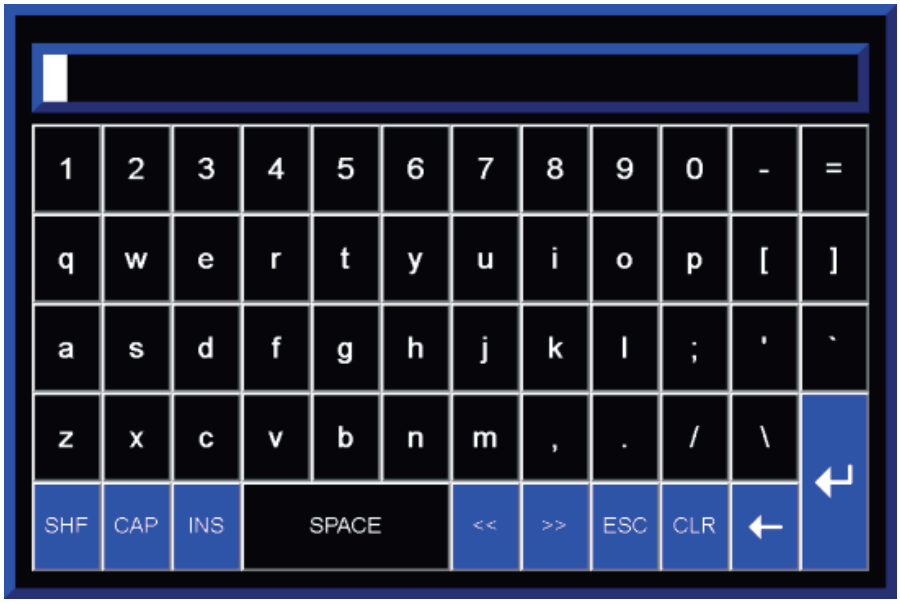
1. Press the **Login** button. The **Login** window opens.

Tip: At runtime:

- If the **Visible** checkbox is clear on the **General** tab of the **Login Button Properties** dialog box, the **Domain** box is hidden in the **Login** dialog box.
- If both the **Visible** checkbox and the **Disable** checkbox are selected on the **General** tab of the **Login Button Properties** dialog box, the **Domain** box is read-only in the **Login** dialog box.
- If the **Visible** checkbox is selected, and the **Disable** checkbox is clear on the **General** tab of the **Login Button Properties** dialog box, the **Domain** box can be edited in the **Login** dialog box.
- If there are only FactoryTalk users in the application, the **Domain** box is usually hidden.



2. Press the **Domain** button or press **F1** on an external keyboard. The string pop-up keyboard opens. For details about using the keyboard, see Use the string pop-up keyboard on page 229.



Type the domain name in the string pop-up keyboard, or on an external keyboard, and then press **Enter**.

3. Press the **User Name** button, or press **F2** on an external keyboard. Type your user name in the string pop-up keyboard, or on an external keyboard, and then press **Enter**.

If the **Hide User Name Entry** checkbox is selected on the **General** tab of the **Login Button Properties** window, the user name entered here is masked by #.

For 4.00 and later runtime applications, the name is not case sensitive. For RSView 3.20 and earlier applications, the name must use ALL CAPS.

4. To enter the password, press the **Password** button, or press **F3** on an external keyboard. Type the password in the string pop-up keyboard, or on an external keyboard, and then press **Enter**. For 4.00 and later runtime applications, the password is case sensitive. For RSView 3.20 and earlier applications, the password is not case sensitive.
5. To close the **Login** window and complete the login, press **Enter**.

What happens when a user logs in

When a user logs in:

- If the previous user did not log out, the previous user is logged out now. If a logout macro is set up for the previous user, the logout macro runs, assigning values to tags. If the previous user belongs to a group, and a logout macro is set up for the group, the logout macro runs.
- The new user is logged in.
- If a login macro is set up for the new user, the macro runs, assigning values to tags. If the new user belongs to a group, and a login macro is set up for the group, the login macro runs.

Problems with logging in

Login is unsuccessful under these circumstances:

- If the graphic display changes remotely before the user has finished logging in, the login is canceled.
- If the entered user name does not match a user account configured in **Runtime Security**
- If the entered user name is not a member of a group configured in **Runtime Security**.
- If the entered user password does not match the user account password configured in **Runtime Security**.
- If the user account is disabled or locked
- If a PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact or PanelView Plus CE terminal's time is not synchronized with the domain controller, domain users cannot log in on the terminal. To enable the users to log in, synchronize the terminal's time with the domain controller.

Change passwords

You can use the **Password** button to change the password at runtime. Any password changes made at runtime are saved in the .mer file.

The **Password** button supports two modes of runtime operations:

- Allows the user to modify their own password
- Allows the user to modify *any* user password

Tip: When designing the application in FactoryTalk View Studio, you must consider which users and groups can access this button, and how to control that access.

The Password button operational mode is configured in FactoryTalk View Studio, and cannot be changed at runtime.

Tip:

- For 5.00 and later applications, it is recommended that you can convert the runtime application mer file to a development application in order to preserve password changes the next time the runtime application file is created. For more information, see Convert runtime application files to development applications on page 181.
- For versions prior to 5.00, it is recommended that when passwords are changed at runtime, the same password changes are made on the development computer. Otherwise, the next time you create the .mer file, the password changes will be lost.
- For 3.20 and earlier applications, you cannot change passwords in FactoryTalk View ME Station. To change a password for a FactoryTalk View user, use the **Runtime Security** editor in FactoryTalk View Studio, and then recreate the runtime application and download the .mer file again. To change a password for a Windows user, use the Windows Control Panel.
- The passwords for Windows-linked users can only be changed in Windows.

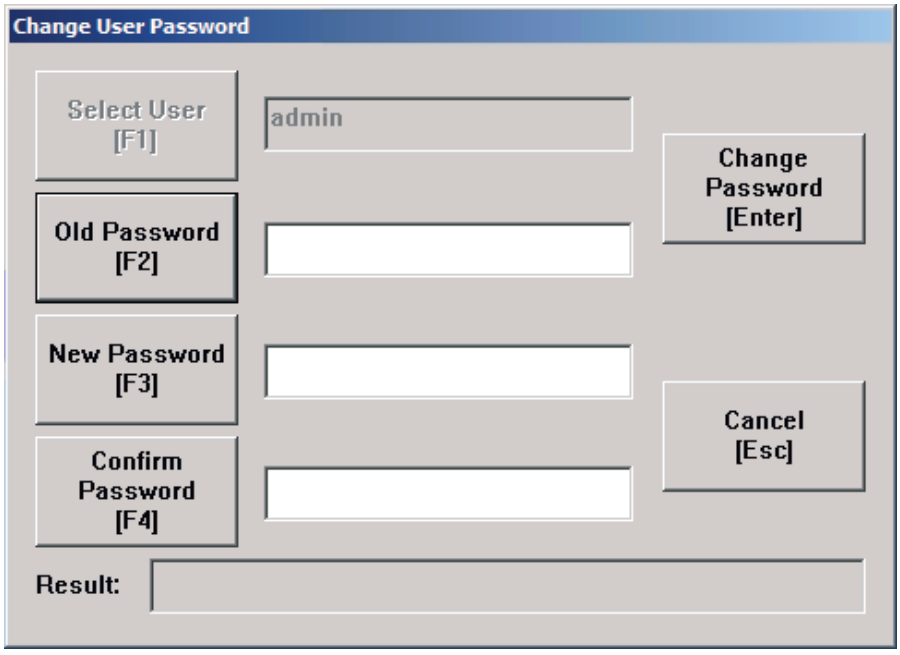
Change your current password

You might want to change your current password.

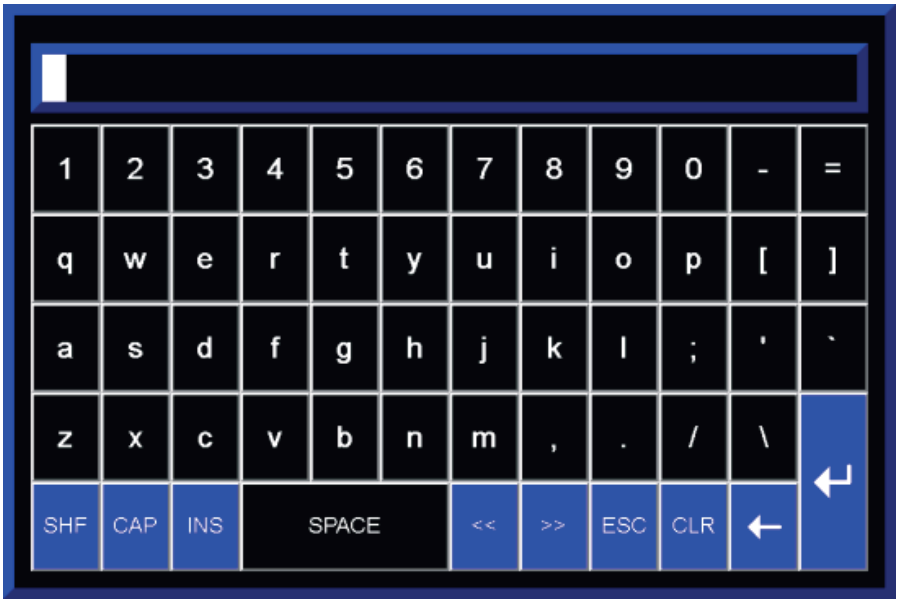
To change your current password

1. Press the **Password** button.

   The **Change User Password** window opens.



2. Press the **Old Password** button, or press **F2** on an external keyboard. The string pop-up keyboard opens. For details about using the keyboard, see Use the string pop-up keyboard on page 229.



   Type your old password in the string pop-up keyboard, or on an external keyboard, and then press **Enter**. The password is case-sensitive.
3. Press the **New Password** button, or press **F3** on an external keyboard.
4. Type the new password in the string pop-up keyboard, or on an external keyboard, and then press **Enter**.
5. Press the **Confirm Password** button, or press **F4** on an external keyboard.
6. Type the new password again, and then press **Enter**.
7. To close the **Change Password** window and complete the change, press **Cancel**.

If electronic signature is used, to change the password, the operator needs to accept the action, and the approver may need to approve the action using the **Electronic Signature** dialog box to execute the action. See Use electronic signature on page 234.

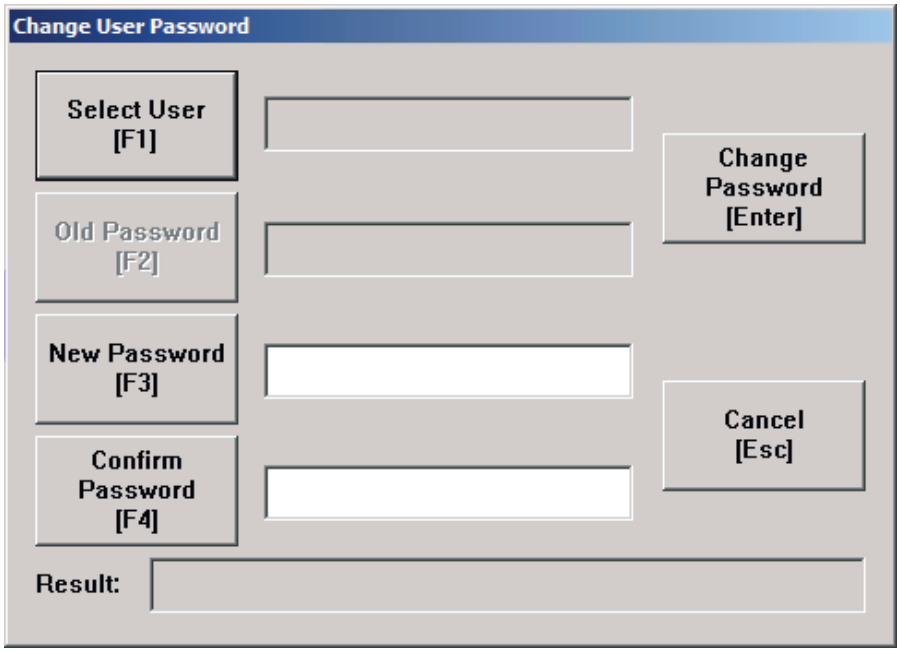
Change any user password

You might want to change your current password.

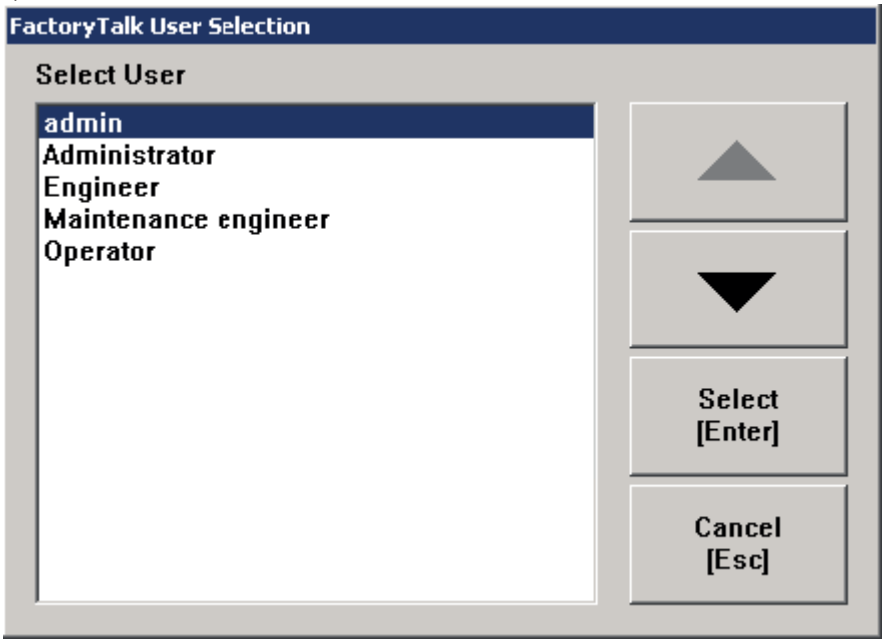
To change any user password

1. Press the **Password** button.

   The **Change User Password** window opens.



2. Press the **Select User** button, or press **F1** on an external keyboard. The **FactoryTalk User Selection** window opens.



3. Press the **Up Arrow** or **Down Arrow** to select a user, and press the **Select** button, or press **Enter** on an external keyboard.
4. Press the **New Password** button, or press **F3** on an external keyboard.
5. Type the new password in the string pop-up keyboard, or on an external keyboard, and then press **Enter**.
6. Press the **Confirm Password** button, or press **F4** on an external keyboard.
7. Type the new password again, and then press **Enter**.
8. To close the **Change Password** window and complete the change, press **Cancel**.

If electronic signature is used, to change the password, the operator needs to accept the action, and the approver may need to approve the action using the **Electronic Signature** dialog box to execute the action. See Use electronic signature on page 234.

Log out

When the current user logs out, if a logout macro is assigned to the user, the macro runs. If the user belongs to a group, and a logout macro is set up for the group, the logout macro runs.

To provide a way for users to log out

- Create a **Logout** button on a graphic display that all logged-in users have access to.

After the current user is logged out, the DEFAULT user is logged in. If a login macro is assigned to the DEFAULT user, the macro runs.

You can also set up the application to automatically log out the current user after a specified period of inactivity. For more information, see Log out automatically on page 149.

Add a user or group

Use the **Add User/Group** button to add a FactoryTalk security user, a Windows-linked user, or a Windows-linked group to the current running application. It is not possible to add a new FactoryTalk Security group to a running application.

Tip: When designing the application in FactoryTalk View Studio, you must consider which users and groups can access this button, and how to control that access.

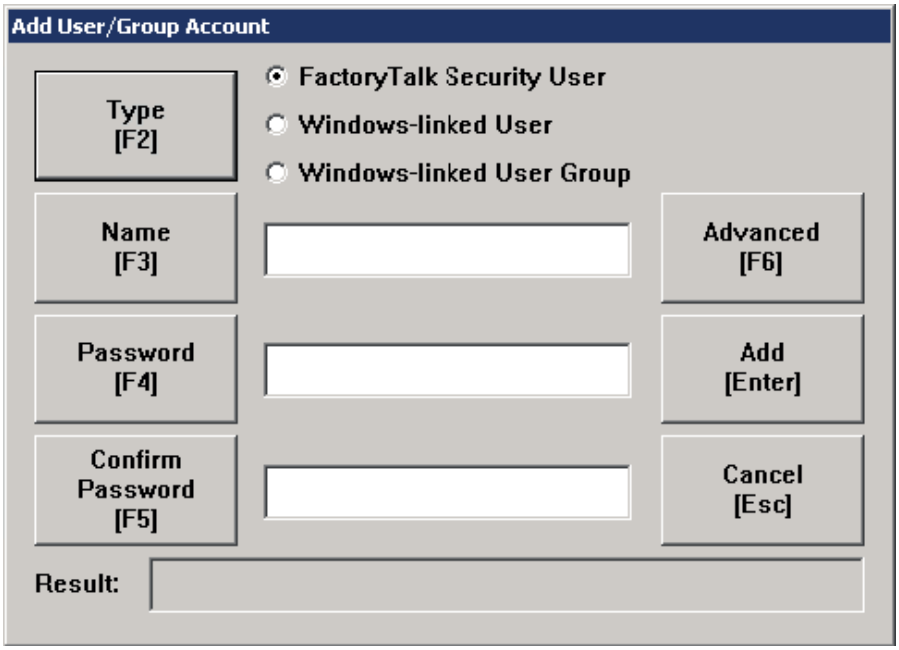
Newly added FactoryTalk Security users, Windows-linked users, and Windows-linked groups cannot log in to the system *until* the added user, or Windows-linked group is made a member of an existing FactoryTalk group, which is configured with runtime security code(s) in **Runtime Security**.

Add a FactoryTalk Security user

Use **Add User/Group** button to add a FactoryTalk Security user.

To add a FactoryTalk Security user

1. Press the **Add User/Group** button to open the **Add User/Group Account** dialog box.



   Press the **Type** button, or press **F2** on an external keyboard to select **FactoryTalk Security User.**

2. Press the **Name** button, or press **F3** on an external keyboard. The string pop-up keyboard opens.

   Type the user name in the string pop-up keyboard, or on an external keyboard, and then press **Enter**. The user name is not case-sensitive.

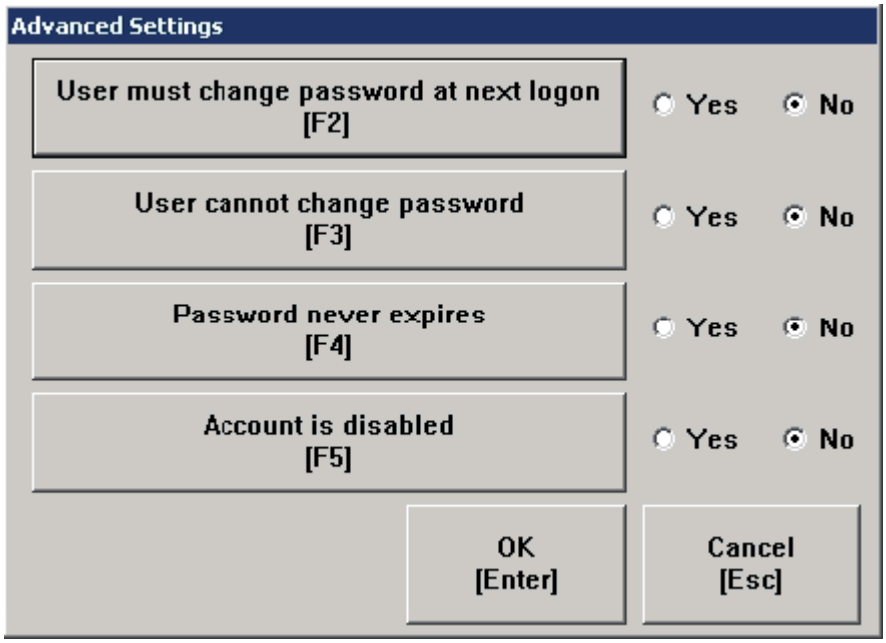
3. Press the **Password** button, or press **F4** on an external keyboard.
4. Type the password in the string pop-up keyboard, or on an external keyboard, and then press **Enter**.

   The password length must meet the Password Policy, which is defined in the FactoryTalk Security Policy. The password is case-sensitive.

5. Press the **Confirm Password** button, or press **F5** on an external keyboard.

   Type the password again, and then press **Enter**.

6. (Optional) Press the **Advanced** button, or press **F6** on an external keyboard. The **Advanced Settings** window opens.



   Press the following buttons to adjust the advanced settings:

   - **User must change password at next logon**
   - **User cannot change password**
   - **Password never expires**
   - **Account is disabled**

     All the settings are set to **No** by default.

7. To close the **Advanced Settings** window and complete the change, press **Enter**.
8. To close the **Add User/Group Account** window, press **Cancel**.

If electronic signature is used, to add a FactoryTalk Security user, the operator needs to accept the action, and the approver may need to approve the action using the **Electronic Signature** dialog box to execute the action. See Use electronic signature on page 234.

Add a Windows-linked user or group

Use the **Add User/Group** button to add a Windows-linked user or group.

To add a Windows-linked user or group

1. Press the **Add User/Group** button to open the **Add User/Group Account** dialog box.
2. Press the **Type** button, or press **F2** on an external keyboard to select **Windows-linked User** or **Windows-linked User Group**.

   When adding a Windows-linked user, or a Windows-linked user group, the system will verify whether the user or group exists in the Windows domain, Windows User Accounts. If the user or group does not exist in the Windows domain, Windows User Accounts, it cannot be added.
3. Press the **Name** button, or press **F3** on an external keyboard. The string pop-up keyboard opens.

   Type the user name in the string pop-up keyboard, or on an external keyboard, and then press **Enter** to save.

   The user name is not case-sensitive.

Tip: On the PanelView Plus 7 and PanelView Plus 6 terminals, there is a five-minute threshold when adding a Windows-linked user, or a Windows-linked user group.
When you add a Windows-linked user, or a Windows-linked user group for the first time, a window prompts you to enter an authorized user name and password.
Within five minutes after a Windows-linked user, or a Windows-linked user group is successfully added, you can add another Windows-linked user, or Windows-linked user group without entering an authorized user name and password.
Beyond five minutes after a Windows-linked user, or a Windows-linked user group is successfully added, a window will prompt you to enter an authorized user name and password again before you can add another Windows-linked user, or Windows-linked user group.

4. To close the **Add User/Group Account** window, press **Cancel**.

If electronic signature is used, to add a Windows-linked user or group, the operator needs to accept the action, and the approver may need to approve the action using the **Electronic Signature** dialog box to execute the action. See Use electronic signature on page 234.

Delete a user or group

The administrator can use the Delete User/Group button to delete a FactoryTalk security user, a Windows-linked user, or a Windows-linked group from the current running application. It is not possible to delete a FactoryTalk Security group from the running application.

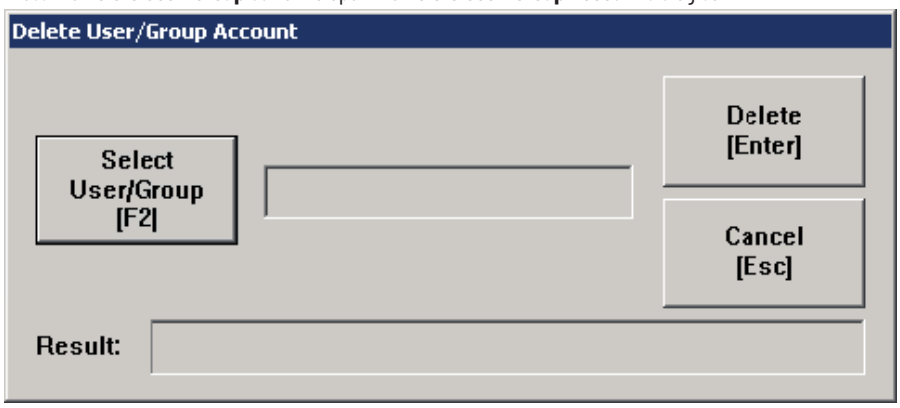
Tip: When designing the application in FactoryTalk View Studio, you must consider which users and groups can access this button, and how to control that access.

If a Windows-linked user group or a Windows-linked user is deleted, the group or user will be removed from the current running application, but will not be removed from Windows.

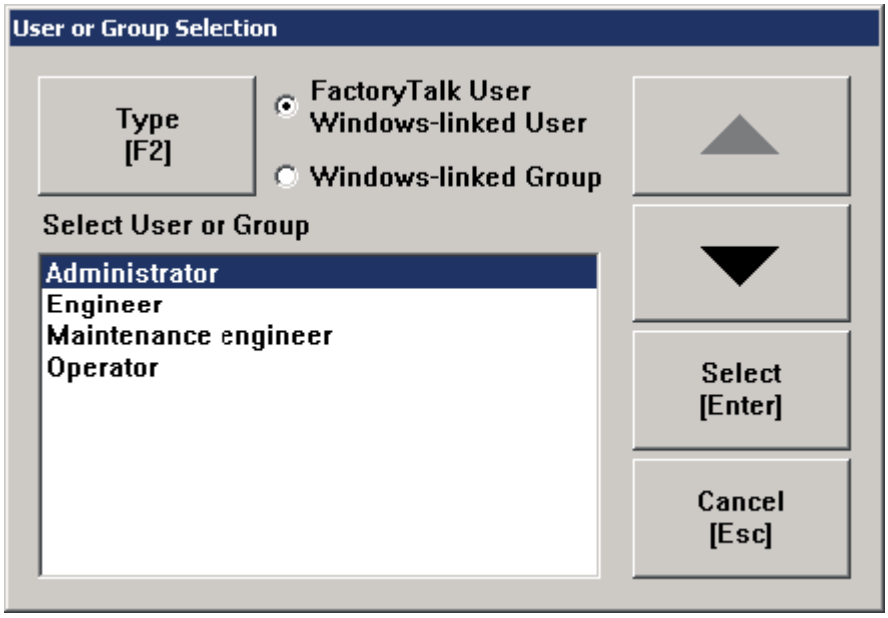
DEFAULT user is not allowed to delete a user or group.

To delete a user or group

1. Press the **Delete User/Group** button to open the **Delete User/Group Account** dialog box.



2. Press the **Select User/Group** button, or press **F2** on an external keyboard to open the **User or Group Selection** dialog box,



3. Press the **Type** button, or press **F2** on an external keyboard to select **FactoryTalk Security User, Windows-linked User**, or **Windows-linked Group**.
4. Press the **Up Arrow** or **Down Arrow** button to select a user, and press the **Select** button, or press **Enter** on an external keyboard.
5. To close the **Delete User/Group** window and complete the change, press **Cancel**.

If electronic signature is used, to delete a user or group, the operator needs to accept the action, and the approver may need to approve the action using the **Electronic Signature** dialog box to execute the action. See Use electronic signature on page 234.

Modify a user or group membership

The administrator can use the **Modify Group Membership** button to change a FactoryTalk user, a Windows-linked user, or a Windows-linked group membership in the current running application.

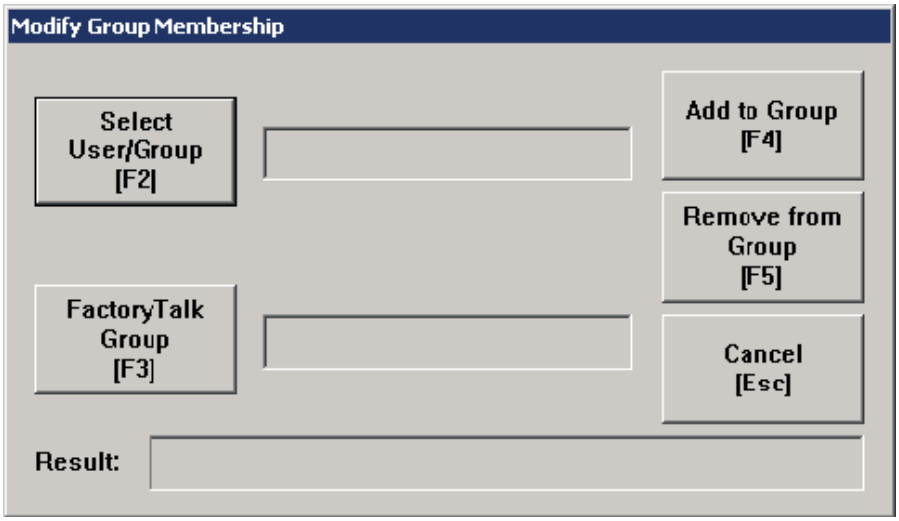
Tip: When designing the application in FactoryTalk View Studio, you must consider which users and groups can access this button, and how to control that access.

If a user or group is added to a group, the user or group inherits the A-P security codes of the group.

- If a user or group is removed from a group, the user or group no longer inherits the A-P security codes of that group.
- If a user or group is a member of multiple FactoryTalk Security groups:
  The user or group inherits the A-P security codes shared by the multiple FactoryTalk Security groups. For example, if a user or group belongs to both Group One and Group Two; Group One has the security codes of A, B, C, and Group Two has the security codes of B, C, D; and then the user or group only inherits the security codes of B, C.
  When a user or group is removed from one FactoryTalk Security group, the user or group is still a member of other FactoryTalk Security groups.
  When a user or group is no longer a member of *any* FactoryTalk Security group, it is not possible for the user or group users to log in to the running application.

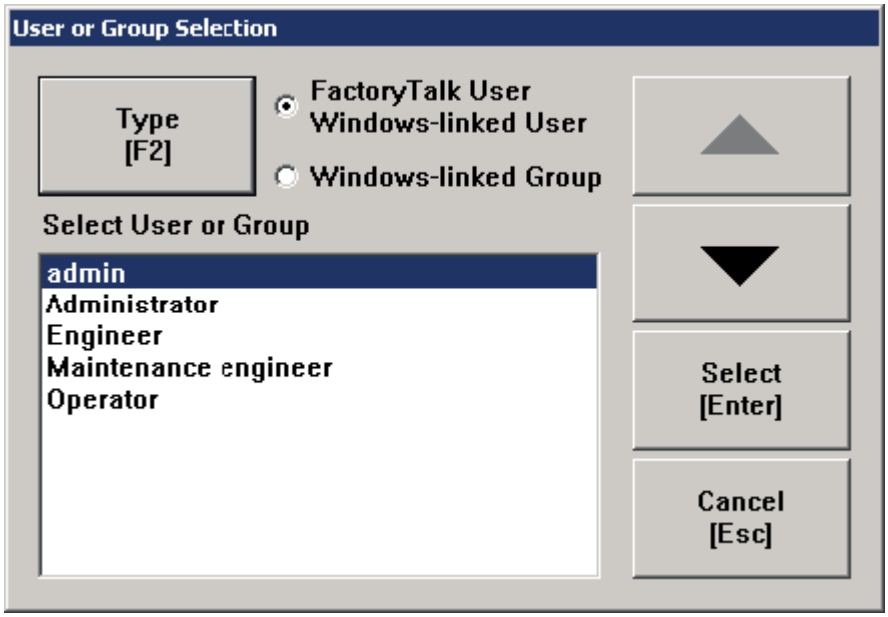
To modify group membership

1. Press the **Modify Group Membership** button to open the **Modify Group Membership** dialog box.



2. Press the **Select User/Group** button, or press **F2** on an external keyboard to open the **User or Group Selection** dialog box.
3. Press the **Type** button, or press **F2** on an external keyboard to select **FactoryTalk User, Windows-linked User**, or **Windows-linked Group**.

4. Press the **Up Arrow** or **Down Arrow** button to select a user, and press the **Select** button, or press **Enter** on an external keyboard.



5. Press the **FactoryTalk Group** button to open **FactoryTalk Group Selection** dialog box.
6. Press the **Up Arrow** or **Down Arrow** button to select a group, and press the **Select** button, or press **Enter** on an external keyboard.
7. To add the user or group to the selected FactoryTalk group, press **Add to Group**, or press **F4** on an external keyboard.

   To remove the user or group from the selected FactoryTalk group, press **Remove from Group**, or press **F5** on an external keyboard.

If electronic signature is used, to modify group membership, the operator needs to accept the action, and the approver may need to approve the action using the **Electronic Signature** dialog box to execute the action. See Use electronic signature on page 234.

Unlock a user

The administrator can use the Unlock User Account button to unlock a FactoryTalk user account, which has been locked.

Tip: When designing the application in FactoryTalk View Studio, you must consider which users and groups can access this button, and how to control that access.

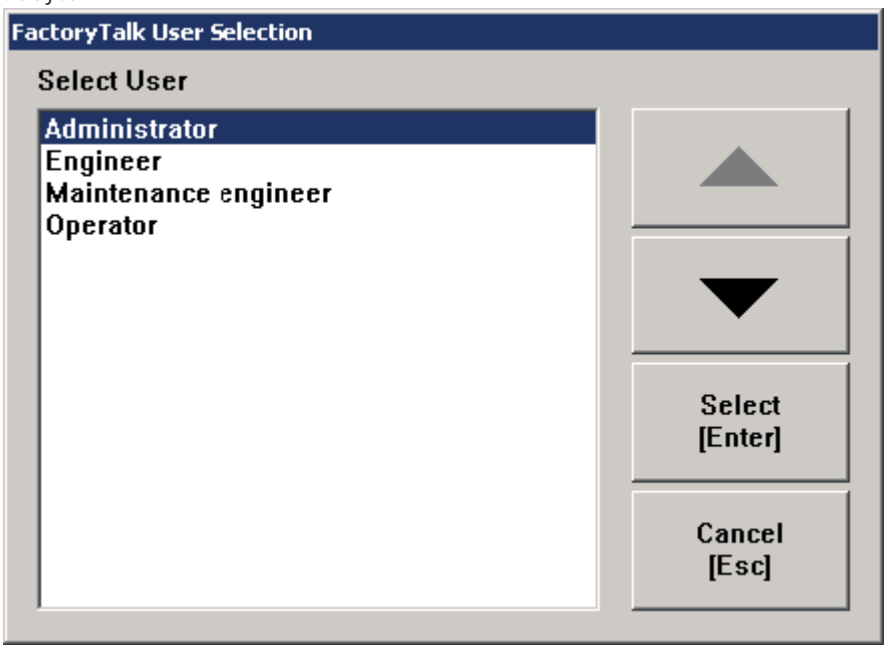
When a user account is locked, the user cannot log in the system.

If a FactoryTalk user exceeds the incorrect password login attempts, the user account will be locked. When the user account is locked, the user cannot log in to the system even with the correct password before the user account is unlocked.

You cannot use this button to unlock a Windows-linked user. Only a Windows Domain administrator can unlock a Windows-linked user from a domain computer.

To unlock a user

1. Press the **Unlock User Account** button to open the **Unlock User Account** dialog box.
2. Press the **Select User** button, or press **F2** on an external keyboard to open the **FactoryTalk User Selection** dialog box.



3. Press the **Up Arrow** or **Down Arrow** button to select a user, and press the **Select** button, or press **Enter** on an external keyboard.
4. To close the **Unlock User Account** window and complete the change, press the **Unlock** button, or press **Enter** on an external keyboard.

If electronic signature is used, to unlock a user, the operator needs to accept the action, and the approver may need to approve the action using the **Electronic Signature** dialog box to execute the action. See Use electronic signature on page 234.

Disable a user

The administrator can use the Disable User button to disable a FactoryTalk user account from the current running application.

Tip: When designing the application in FactoryTalk View Studio, you must consider which users and groups can access this button, and how to control that access.

When a user account is disabled, the user cannot log in the system.

You cannot use this button to disable a Windows-linked user. Only a Windows Domain administrator can disable a Windows-linked user from a domain computer.

To disable a user

1. Press the **Disable User** button to open the **Disable User Account** dialog box.
2. Press the **Select User** button, or press **F2** on an external keyboard to open the **FactoryTalk User Selection** dialog box.
3. Press the **Up Arrow** or **Down Arrow** button to select a user, and press the **Select** button, or press **Enter** on an external keyboard.
4. To close the **Disable User Account** window and complete the change, press the **Disable** button, or press **Enter** on an external keyboard.

If electronic signature is used, to disable a user, the operator needs to accept the action, and the approver may need to approve the action using the **Electronic Signature** dialog box to execute the action. See Use electronic signature on page 234.

Enable a user

The administrator can use the Enable User button to enable a disabled FactoryTalk user account from the current running application.

Tip: When designing the application in FactoryTalk View Studio, you must consider which users and groups can access this button, and how to control that access.

When a disabled user is enabled, the user can log in the system again.

You cannot use this button to enable a Windows-linked user. Only a Windows Domain administrator can enable a Windows-linked user from a domain computer.

To enable a user

1. Press the **Enable User** button to open the **Enable User Account** dialog box.
2. Press the **Select User** button, or press **F2** on an external keyboard to open the **FactoryTalk User Selection** dialog box.
3. Press the **Up Arrow** or **Down Arrow** button to select a user, and press the **Select** button, or press **Enter** on an external keyboard.
4. To close the **Enable User Account** window and complete the change, press the **Enable** button, or press **Enter** on an external keyboard.

If electronic signature is used, to enable a user, the operator needs to accept the action, and the approver may need to approve the action using the **Electronic Signature** dialog box to execute the action. See Use electronic signature on page 234.

Change User Properties

The administrator can use the **Change User Properties** button to modify a FactoryTalk user's properties in the current running application.

Tip: When designing the application in FactoryTalk View Studio, you must consider which users and groups can access this button, and how to control that access.

To change a user's properties

1. Press the **Change User Properties** button to open the **Change User Properties** dialog box.
2. Press the **Select User** button, or press **F2** on an external keyboard to open the **FactoryTalk User Selection** dialog box.
3. Press the **Up Arrow** or **Down Arrow** button to select a user, and press the **Select** button, or press **Enter** on an external keyboard.
   The **FactoryTalk User Selection** dialog box only shows the users already configured in FactoryTalk Directory and packed in the application.
4. Press the following buttons to adjust the advanced settings:
   - **User must change password at next logon**
   - **User cannot change password**
   - **Password never expires**
   - **Account is disabled**
5. Press **Change** to save the changes.

If electronic signature is used, to modify a user's properties, the operator needs to accept the action, and the approver may need to approve the action using the **Electronic Signature** dialog box to execute the action. See Use electronic signature on page 234.

Enter numeric values

The operator can enter or ramp numeric values at runtime using the Numeric Input Enable button or the numeric input cursor point.

When the operator presses the button or activates the cursor point, the numeric pop-up keypad or scratchpad opens. If you set up the button or cursor point to ramp, pressing the object gives it focus, but does not open a pop-up window.

To use the numeric pop-up scratchpad, a keyboard must be attached to the runtime computer, or the terminal must be a keypad terminal.

For information about setting up the Numeric Input Enable button and numeric input cursor point, see Help.

You can also use a Ramp button to ramp numeric values. For details, see Help for the button.

Activate the cursor point

When the numeric input cursor point has focus, the operator can activate the cursor point by pressing any of these keys on the keyboard or keypad:

- Numbers from 0 to 9
- The minus sign (-) or decimal point (.)
- The **Enter** key, or an enter button
- The Backspace key, or a backspace button

When the operator activates the numeric input cursor point, the numeric pop-up keypad or scratchpad opens.

Ramp numeric values

If you set up the **Numeric Input Enable** button or numeric input cursor point to ramp values, the operator presses the object to give it focus. When the object has focus, the operator can press a move up or move down button, or the Up Arrow or Down Arrow key on a keyboard or keypad, to ramp the value at the **Value** connection.

Use the numeric pop-up keypad

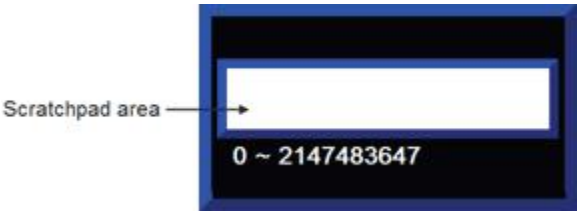
To enter numeric values, the operator must open the numeric pop-up keypad by pressing a numeric input enable button or numeric input cursor point object. The keypad can accept up to 17 digits, including the decimal point, leading zero, and minus sign.

When the keypad is open, no other objects in the graphic display can accept input.

Use the numeric pop-up scratchpad

To enter numeric values, the operator must open the numeric pop-up scratchpad by pressing a numeric input enable button or numeric input cursor point object. The scratchpad can accept up to 17 digits, including the decimal point, leading zero, and minus sign. The numeric pop-up scratchpad does not contain any buttons. The computer must have an external keyboard or keypad to use and close the numeric pop-up scratchpad.

When the scratchpad is open, no other objects in the graphic display can accept input.



Use buttons and keys with the numeric pop-up windows

You can use the following buttons and their keyboard or keypad equivalents with the numeric pop-up keypad. For the pop-up scratchpad, you can use the keys on an external keyboard or keypad only.

Use this button or key	To do this
Decimal (.)	Type a decimal point. This button is visible only if the decimal point type for the Numeric Input Enable button or numeric input cursor point is Implicit.
Minus (-)	Toggle the value between positive and negative.
ESC	Close the pop-up window without writing the value to the object's **Value connection**.
Backspace	Delete the right-most digit in the scratchpad. If no digits are left, the minus sign, if any, is removed.
Enter	Close the pop-up window and write the value to the object's **Value connection**. If the **Optional Expression** connection is assigned to the button or cursor point, FactoryTalk View calculates the value of the expression and writes the result to the **Value connection**. For details, see Use write expressions on page 365. If **Enter key handshaking** is set up for the button or cursor point, the object's **Enter connection** is set to 1, and the handshaking timer begins timing. For information about using **Enter key handshaking**, see Methods of Enter key handshaking on page 335.
Delete	Clear the scratchpad. There is no Delete button in the pop-up keypad, but you can use the Delete key on an external keyboard or keypad.

How values are ramped

If the button or cursor point is being used to ramp a value at the **Value** connection:

- When the tag assigned to the **Value** connection is an integer tag, but the ramp value is a floating point value, the ramp value is added to (or subtracted from) the **Value** connection value, and then the result is rounded and written to the **Value** connection.
  For information about how values are rounded, see How values are rounded on page 80.
- If the operator presses and holds down the move up or move down button, or the Up Arrow or Down Arrow key on the keyboard or keypad, the button or key goes into auto repeat mode. The ramp value is added to (or subtracted from) the last value sent to the **Value** connection.
  For more information about auto repeat mode, see To set up auto repeat for a button, use one of these methods on page 335.
- If ramping the button or cursor point would cause the value at the **Value** connection to exceed the maximum value, or be less than the minimum value, a message is written to the **Diagnostics List** and the value at the **Value** connection is not changed.

How values are calculated

When the operator presses the **Enter** button, the value that is sent to the Value connection depends on the type of decimal point and whether the Optional Expression is assigned.

If the decimal point is set to **Fixed Position**, the decimal point is stripped before sending the value to the Value connection. For example, if the shown value is 9.25, the value sent to the Value connection is 925.

Decimal point	Optional Expression not assigned	Optional Expression assigned
Implicit	Displayed value compared to the minimum and maximum range*. Floating point values are rounded if the Value connection is an integer connection.	Displayed value is compared to the minimum and maximum range. If OK, substituted for ? in the expression. Result is written to Value connection. Floating point values are rounded if the Value connection is an integer connection.
Fixed position, Digits after decimal = 0	Displayed value compared to the minimum and maximum range*.	Displayed value is compared to the minimum and maximum range*. If OK, substituted for ? in the expression. Result is written to Value connection.
Fixed position, Digits after decimal > 0 Fixed position output = Stripped	Decimal point is stripped from the displayed value. Result compared to the minimum and maximum range*.	Decimal point is stripped from displayed value. Result compared to the minimum and maximum range*. Result is substituted for ? in the expression. Expression result is written to Value connection.
Fixed position, Digits after decimal > 0 Fixed position output = Displayed Value	Displayed value compared to the minimum and maximum range*.	Displayed value is compared to the minimum and maximum range*. If OK, substituted for ? in the expression. Result is written to Value connection.

*The minimum and maximum range is specified on the **Numeric** tab for the numeric input enable button. If the value is within the range, it is sent to the Value connection.

Tip: Floating point values (the default values for numeric input objects) are rounded to six digits. Therefore, using a floating point data type and adding to it a value that exceeds the six significant digits will result in a loss of precision.

Problems with the numeric pop-up windows

Problems with the numeric pop-up keypad and scratchpad occur under these circumstances:

- If the graphic display changes remotely before the user has pressed the Enter button, the pop-up window closes without writing out a value.
- If the screen resolution is smaller than 124 pixels wide by 240 pixels high, the pop-up keypad cannot open.
- If the screen resolution is smaller than 124 pixels wide by 68 pixels high, the pop-up scratchpad cannot open.
- If the value is too large for the data type of the tag assigned to the **Value** connection, the pop-up window remains open and the value is not written to the connection. The scratchpad area changes to red to alert the operator of the error.

- If the value is outside the minimum and maximum range specified for the object, the pop-up window remains open and the value is not written to the connection. The scratchpad area changes to red to alert the operator of the error.
- If **Enter key handshaking** is still in effect, the pop-up window closes but the value is not written to the connection.

Enter string values

The operator can enter string values at runtime using the **String Input Enable** button.

When the operator presses the button, the string pop-up keyboard or scratchpad opens, depending on how you set up the button. To use the scratchpad, a keyboard must be attached to the runtime computer.

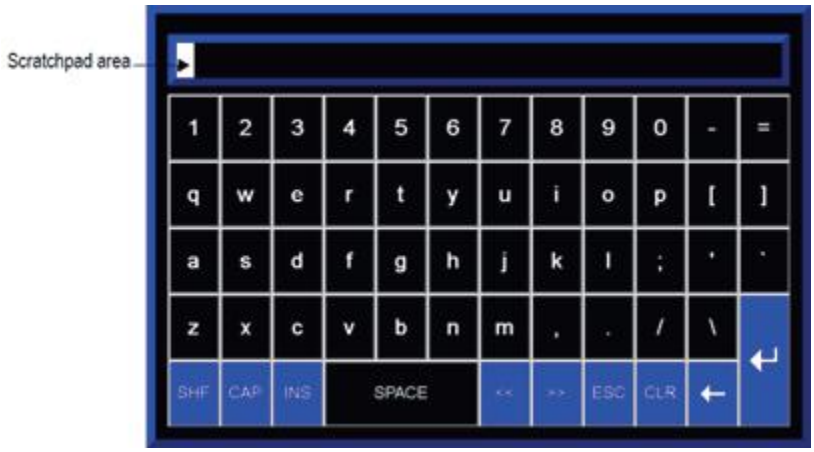
For information about setting up the **String Input Enable** button, see Help.

Use the string pop-up keyboard

To enter string values at runtime, the operator must open the string pop-up keyboard by pressing a string input enable button. The keyboard also opens when the operator presses the **User** and **Password** buttons in the **Login** dialog box or the **Old Password**, **New Password** and **Confirm Password** buttons in the **Change Password** dialog box at runtime.

If the operator opens the pop-up keyboard by pressing a **String Input Enable** button, the number of characters the keyboard accepts depends on how you set up the button.

When the keyboard is open, no other objects in the graphic display can accept input.



Use the string pop-up character input

You can use a string pop-up character input instead of the string pop-up keyboard. With the string pop-up character input, you use the arrow keys to select the characters to input.

The string pop-up character input opens when the operator presses a **String Input Enable** button, or the **User** or **Password** button in the **Login** dialog box. The pop-up is also used with the password button graphic object.

If the operator opens the string pop-up character input by pressing a **String Input Enable** button, the number of characters the input accepts depends on how you set up the button.

Use the string pop-up character input on a personal computer

You might want to use the string pop-up character input on a personal computer.

To use the string pop-up character input on a personal computer

1. In FactoryTalk View Studio, from the **Tools** menu, select **Options**.
2. Click the **String Pop-Up** tab.
3. Select **Use the string pop-up character input**.

Use the string pop-up character input on a PanelView Plus 7, PanelView Plus 6, PanelView Plus, or PanelView Plus CE terminal

You might want to use the string pop-up character input on a PanelView Plus 7, PanelView Plus 6, PanelView Plus, or PanelView Plus CE terminal.

To use the string pop-up character input on a PanelView Plus 7, PanelView Plus 6, PanelView Plus, or PanelView Plus CE terminal

1. In FactoryTalk View ME Station, click **Terminal Settings.**
2. Double-click **Input Devices.**
3. Select **String Pop-Up**, and then click **Enter**.
4. Specify whether to invoke the pop-up character input instead of the pop-up keyboard, and then click **OK**.

Use the string pop-up scratchpad

To enter string values, the operator must open the string pop-up scratchpad by pressing a string input enable button. The number of characters the scratchpad accepts depends on how you set up the **String Input Enable** button.

The scratchpad does not contain any buttons. The computer must have an external keyboard or keypad to use and close the pop-up scratchpad. If the runtime terminal is a keypad terminal, the operator can enter numbers in the scratchpad and close the scratchpad.

When the scratchpad is open, no other objects in the graphic display can accept input.

Use buttons and keys with the string pop-up windows

You can use the following buttons and their keyboard equivalents with the string pop-up keyboard. For the pop-up scratchpad, you can use the keys on an external keyboard only. The string pop-up character input does not have an SHF or CAP key, and it has additional arrow keys for selecting the characters to input.

Use this pop-up keyboard button	Or this keyboard equivalent	To do this
SHF	none	Capitalize a single letter, or type a shifted character such as #.
CAP	none	Capitalize multiple letters.
INS	Insert	Toggle between insert and overstrike modes.
SPACE	Spacebar	Insert a space.
<<	Left Arrow	Move the cursor to the left.
>>	Right Arrow	Move the cursor to the right.
ESC	Esc	Close the pop-up window without writing the string to the **Login** dialog box, the **Change Password** dialog box, or the **String Input Enable** button's **Value connection**.
CLR	Delete	Clear the scratchpad.
Backspace	Backspace	Delete the character in front of the cursor.
Enter	Enter	Close the pop-up window and write the string to the **Login** dialog box, the **Change Password** dialog box, or the **String Input Enable** button's **Value connection**. If **Enter key handshaking** is set up for the **String Input Enable** button, the button's **Enter connection** is set to 1, and the handshaking timer begins timing. For information about using **Enter key handshaking**, see Methods of Enter key handshaking on page 335.

What is written to the Value connection

When the operator presses the Enter button in the pop-up keyboard or scratchpad, the string that is sent to the **Value connection** depends on how the **String Input Enable** button is set up.

- If a fill character is set up for the button, and the operator enters fewer than the maximum number of input characters, the fill characters are placed after the string the operator enters.
  - Spaces have a hex value of 20.
  - Zeroes have a hex value of 30.
  - FF characters have a hex value of FF.
  - Null characters have a hex value of 0. The null character indicates the end of string input. It does not add to the actual string length.
- If the number of input characters is fewer than the number of characters in the length of the string tag assigned to the **Value connection**, the remaining spaces are padded with the null character.

When the string is written to the **Value connection**, the first character is placed in the high order byte of the first word at the tag address, the second character is placed in the low order byte of the first word, and so on.

Problems with the string pop-up windows

Problems with the string pop-up keyboard and scratchpad occur under these circumstances:

- If the graphic display changes remotely before the user has pressed the Enter button, the pop-up window closes without writing out a string.
- If the screen resolution is smaller than 236 pixels wide by 208 pixels high, the pop-up keyboard cannot open.
- If the screen resolution is smaller than 236 pixels wide by 44 pixels high, the pop-up scratchpad cannot open.
- If the string pop-up window is set up to accept more characters than the **Value** connection tag length, the pop-up window remains open, and the string is not written to the connection. The scratchpad area changes to red to alert the operator of the error.
- If **Enter key handshaking** is still in effect, the pop-up window closes but the value is not written to the connection.

Change tag values

This section gives an overview of the graphic objects that you can use to change tag values. For information about setting up the objects, see Set up graphic objects on page 319 and Help.

The operator uses these objects to start and stop plant operations, and to control machines and processes. Choose the objects that best suit your process. Set up the data source to carry out the desired actions in response to the changes in tag values.

The operator can use function keys with all of these objects except control list selectors, third-party ActiveX objects, and drawing objects with slider animation. The last three types of objects can be pressed using a mouse or touch screen.

For information about assigning function keys to graphic objects, see Assign function keys to buttons on page 329.

IMPORTANT: Do not use push buttons for emergency stops. Emergency stop buttons must be hard-wired.

You can also use macros to assign values to tags. For more information, see Use macros on page 421.

The operator can change tag values at runtime using these graphic objects:

Use this graphic object	To do this
Momentary push button	Start a process or action by sending one value to the tag when pressed, and another value when released.
Maintained push button	Toggle between two values by sending one value to the tag when pressed, and a second value the next time the button is pressed and released. This button is useful for changing a setting within a machine or process, but not for starting the machine or process.
Latched push button	Start a machine or process and remain set (latched) until the process is completed, by sending a value to the tag when pressed, and retaining this value until reset (unlatched) by the **Handshake** connection.
Multistate push button	Cycle through a series of values. Each time the operator presses the button, the value for the next state is sent to the tag. When the button is in its last state, pressing it causes the button to change to its first state and write out the first state value. This button is useful when you want the operator to see and select multiple options in sequence, using a single button. The button shows the current state of a process or operation by showing a different color, caption, or image to reflect the different states.
Interlocked push button	Use a group of buttons to send values to the same tag. When the operator presses a button in the group, the button's value is sent to the tag, and the button remains highlighted as long as the tag value is the same as the button's value. Pressing a new button in the group releases the other button and sends a new value to the tag. You can also use a single Interlocked push button to send a value to a tag.
Ramp button	Increase or decrease the value of a tag by a specified integer or floating-point value. Use two Ramp buttons together to create a raise/lower control.
Numeric input enable button	Enter a numeric value and write the value to a tag. You can also use this object to ramp values.
Numeric input cursor point	Enter a numeric value and write the value to a tag. You can also use this object to ramp values.
String Input Enable button	Enter a string value and write the value to a tag.
RecipePlus button	Write values for all the ingredients in the selected recipe to a set of tags. The button works with the RecipePlus table and RecipePlus selector graphic objects.
Drawing object with horizontal or vertical slider animation	Control the value of a tag by dragging the slider object with a mouse. The pixel position of the slider is translated into a value that is written to the tag. If the value of the tag is changed externally, the position of the slider changes to reflect this.
Control list selector	Select from a list of states for a process or operation. The list is highlighted to show the current state, and the operator can scroll through the list to select a different state. The value assigned to the selected state is written to the tag. If the value of the tag is changed externally, the position of the highlight changes to reflect this.
ActiveX object	A third-party object, connected to an analog, digital, or string tag, including both HMI and data server tags. When the object's property value changes, the new value is written to the associated tag.

You can attach visibility animation to these graphic objects, to show or hide the objects based on changes in tag or expression values. For information about visibility animation, see Set up the different types of animation on page 343.

For information about creating graphic objects, see Use graphic objects on page 275. For information about setting up graphic objects, see Set up graphic objects on page 319 and Help.

Use electronic signature

If electronic signature is used, the operator must accept the action, and the approver might need to approve the action using the **Electronic Signature** window to execute the action.

Tip: If electronic signature is used, DEFAULT user cannot propose the action.

The **Electronic Signature** window has three tabs:

- The **Operator** tab - Operator accepts the action on this tab.
- The **Counter Signatory** tab - Approver approves the action on this tab.
- The **Review Action** tab - This tab shows the action that the operator proposed.

Depending on the design time electronic signature configuration, there are three operation modes:

- **Comment only** - Operator accepts the action by entering a comment (if required) and pressing **Accept** on the **Operator** tab.
- **Single signatory** - Operator accepts the action by entering password and comment (if required), and pressing **Accept** on the **Operator** tab.
- **Counter signatory** - Operator accepts the action by entering password and comment (if required), and pressing **Accept** on the **Operator** tab; and approver approves the action by entering user name, password, and comment (if required) on the **Counter Signatory** tab.

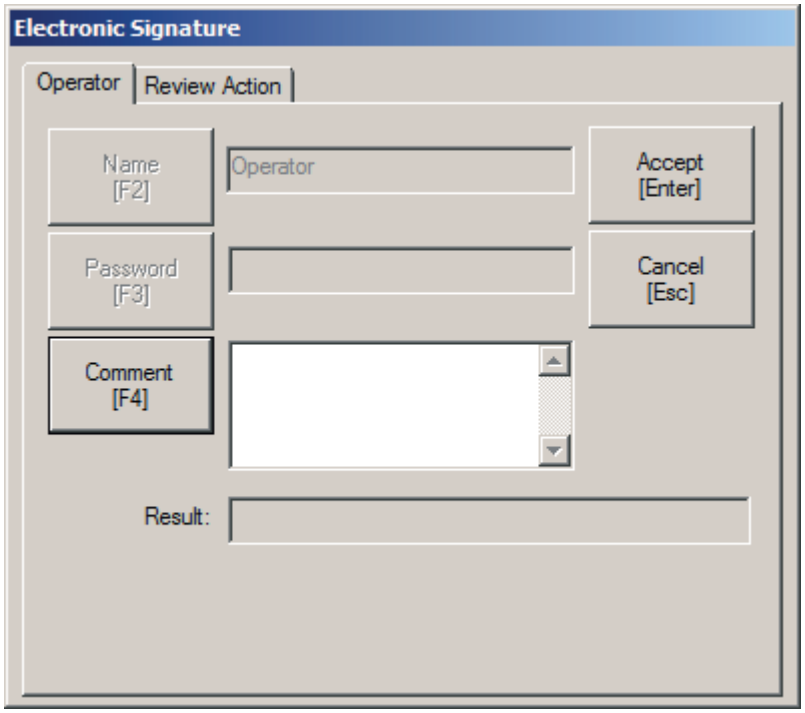
For more information, see "Set up electronic signature properties" in FactoryTalk View ME Help.

Accept an action in the comment only mode

On the **E-Signature** tab of the object **Properties** dialog box, if the **Require Electronic Signature** checkbox is selected, and the **Require re-authentication** checkbox is clear, the operation mode is comment only.

To accept an action in the comment only mode

- The logged-in operator enters a comment (if required) on the **Operator** tab, and presses **Accept**.



Tip: On the **Operator** tab, the **Name** box shows the logged-in operator name. To change the name, the operator need log out, and the expected operator logs in and proposes the action.

After the operator presses **Accept**, the **Electronic Signature** window closes, the action is executed, and the system generates an audit message. See Example: Operator accepts an action in the comment only mode, or the single signatory mode on page 239.

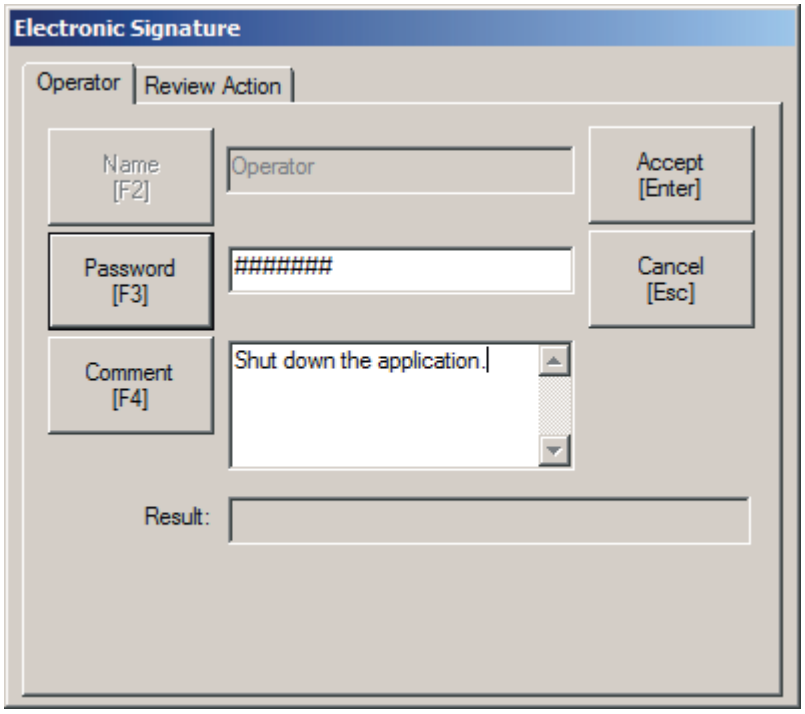
If the operator presses **Cancel**, the **Electronic Signature** window closes, the action is not executed, and the system does not generate audit messages.

Accept an action in the single signatory mode

On the **E-Signature** tab of the object **Properties** dialog box, if the **Require Electronic Signature** checkbox, and the **Require re-authentication** checkbox are selected, and the **Require counter signature** checkbox is clear, the operation mode is single signatory.

To accept an action in the single signatory mode

- The logged-in operator enters password, and comment (if required) on the **Operator** tab, and presses **Accept**.



Tip: On the **Operator** tab, the **Name** box shows the logged-in operator name. To change the name, the operator need log out, and the expected operator logs in and proposes the action.

After the operator presses **Accept**, the **Electronic Signature** window closes, the action is executed, and the system generates an audit message. See Example: Operator accepts an action in the comment only mode, or the single signatory mode on page 239.

If the operator presses **Cancel**, the **Electronic Signature** window closes, the action is not executed, and the system does not generate audit messages.

Approve an action in the counter signatory mode

On the **E-Signature** tab of the object **Properties** dialog box, if the **Require Electronic Signature** checkbox, the **Require re-authentication** checkbox, and the **Require counter signature** checkbox are all selected, the operation mode is counter signatory.

To approve an action in the counter signatory mode

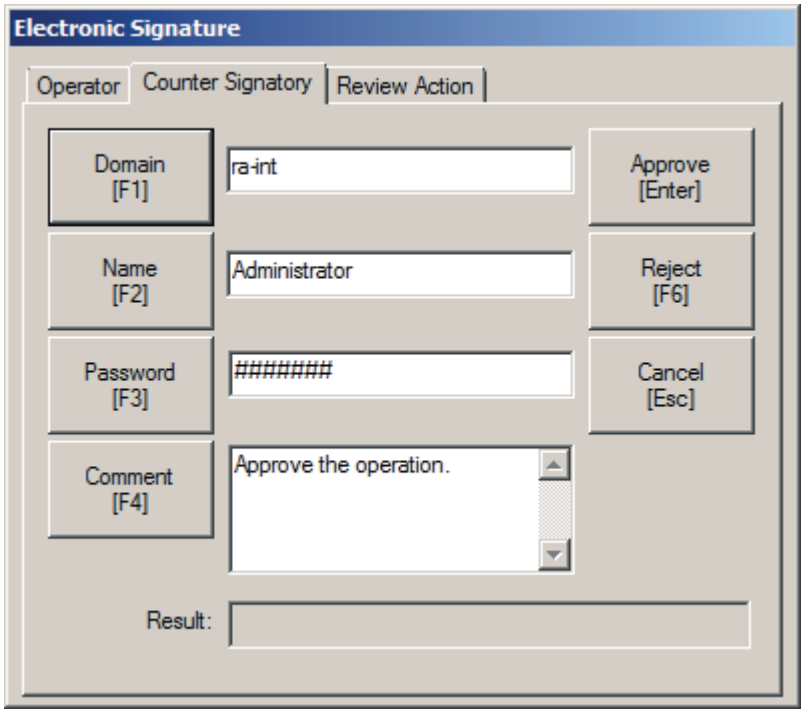
1. The logged-in operator accepts the action on the **Operator** tab. See Re-authenticate an action in the single signatory mode on page 235.

   Tip: After the operator presses **Accept**, to cancel the action, press **Cancel** on the **Counter Signatory** tab. The **Electronic Signature** window closes, the action is not executed, and the system generates audit messages. See Example: Operator accepts an action and then cancels the action on page 239.

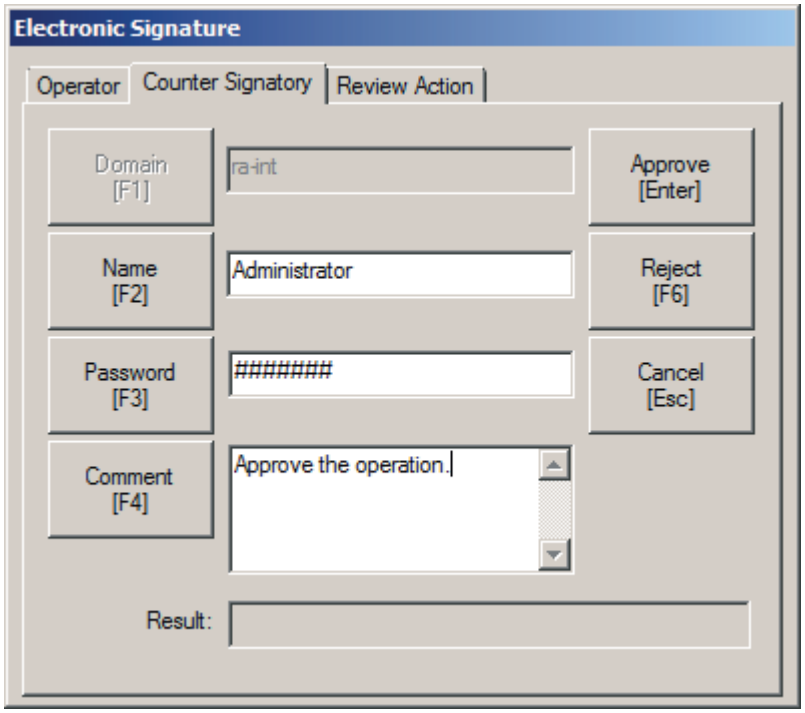
2. On the **Counter Signatory** tab, the approver enters the domain name (if required), user name, password, and comment (if required), and presses **Accept**.

   Tip: The approver must be different from the operator, and must be a member of the authorized group configured in FactoryTalk Security.

When the **Domain** box is enabled:



When the **Domain** box is disabled:



If the approver's security credentials are correct, the **Electronic Signature** window closes, the action is executed, and the system generates audit messages. See Example: Operator accepts an action and approver approves the action on page 239.

If the approver presses **Reject**, the **Electronic Signature** window closes, the action is not executed, and the system generates audit messages. See Example: Operator accepts an action and approver rejects the action on page 239.

Audit message examples

If electronic signature is used, when an operator proposes an action, the system may generate audit messages.

You can view audit messages in:

- Audit graphic objects on a display you create
- CSV files exported when triggered
- FactoryTalk® AssetCentre

For more information about audit log file, see FactoryTalk audit log file on page 139.

Example: Operator accepts an action in the comment only mode, or the single signatory mode

In the comment only mode, or the single signatory mode, when the operator accepts an action, the system generates the following audit message:

> Electronic Signature Action: Write '1' to the tag 'System_auto_mode'. Previous value was '0'. Performed by 'Tony'. Comment: Starting production run 24.Aug.2015.

Example: Operator accepts an action and approver approves the action

In the counter signatory mode, when the operator accepts an action, and the approver approves the action, the system generates the following audit messages:

> Electronic Signature Action: Write '225' to the tag 'Curing_Delay'. Previous value was '200'. Proposed by 'Tony'. Comment: Humid day requires longer material cure.

> Electronic Signature Action: Proposed Write '225' to the tag 'A'. Previous value was '200'. Approved by 'Emil'. Comment: Agree with Tony's assessment of humidity.

Example: Operator accepts an action and approver rejects the action

In the counter signatory mode, when the operator accepts an action, and the approver rejects the action, the system generates the following audit messages:

> Electronic Signature Action: Write '225' to the tag 'Curing_Delay'. Previous value was '200'. Proposed by 'Tony'. Comment: Humid day requires longer material cure.

> Electronic Signature Action: Proposed Write '225' to the tag Curing_Delay'. Previous value was '200'. Rejected by 'Emil'. Comment: Per QA process guidelines, relative humidity is only 55% and will not affect material curing.

Example: Operator accepts an action and then cancels the action

In the counter signatory mode, when the operator accepts an action, and then cancels the action, the system generates the following audit messages:

> Electronic Signature Action: Write '225' to the tag 'Curing_Delay'. Previous value was '200'. Proposed by 'Tony'. Comment: Humid day requires longer material cure.

Electronic Signature Action: Proposed Write '225' to the tag 'Curing_Delay'. Previous value was '200'. Canceled

View tag data

This section gives an overview of the graphic objects that you can use to show tag data. For information about setting up the objects, see Set up graphic objects on page 319 and Help.

The operator can view tag data at runtime using these graphic objects:

Use this graphic object	To show this
Numeric display	Numeric tag values. For example, show the current temperature of an oven.
Numeric input cursor point	Numeric tag values. For example, show the current temperature of an oven.
String display	String tag values. For example, set up the data source to generate strings that report on the state of a process or operation, or that provide the operator with instructions about what to do next.
Bar graph	Numeric values in bar graph format. The bar graph increases or decreases in size to show the changing value.
Gauge	Numeric values in dial format. The gauge's needle moves around the dial to show the changing value.
Multistate indicator	The state of a process, on a panel that changes its color, image, or caption to indicate the current state. Each state is set up to correspond to a numeric tag value or least significant bit.
Symbol	The state of a process, using a monochrome image that changes color to indicate the current state. Each state is set up to correspond to a numeric tag value or least significant bit. This object is useful for showing the state of a process or operation at a glance.
List indicator	The state of a process, using a list of possible states with the current state highlighted. Each state is represented by a caption in the list, and is set up to correspond to a numeric tag value or least significant bit. This indicator is useful if you want to view the current state but also want to see the other possible states. For sequential processes, the list can alert the operator about what happens next in the process.
Trend	Historical or current numeric tag values, plotted against time or shown in an XY plot where one or more tags' values are plotted against another tag's values to show the relationship between them.
RecipePlus table	Current tag values and data set values of the ingredients in the selected recipe, and the number of ingredients in the recipe. The table works with the **RecipePlus** button and **RecipePlus** selector graphic objects.
Drawing object with rotation, width, height, fill, color, or horizontal or vertical position animation	Show the value of a tag using a pictorial representation that shows the current value in relation to a range of possible values. For example, use rotation animation to show the tag value as a needle's position on a dial. For color animation, assign different colors to represent different values.
ActiveX object	A third-party object, connected to an analog, digital, or string tag, including both HMI and data server tags. The data shown depends on the object.

Many of these objects can be set up to manipulate tag values using expressions, and show the expression result rather than the original tag value. For information about expressions, see Use expressions on page 352.

You can also attach visibility animation to these graphic objects, to show or hide the objects based on changes in tag or expression values. For information about visibility animation, see Set up the different types of animation on page 343.

For information about creating graphic objects, see Use graphic objects on page 275. For information about setting up graphic objects, see Set up graphic objects on page 319 or Help.

Show the date and time

To show the current date and time, create a time and date display. This object uses the operating system's date and time, in the format of the application's current language, and therefore does not require tags or expressions.

View alarms and messages

The operator can view alarms and other messages at runtime using these graphic objects and graphic displays:

This information	Appears in this object	In this default graphic display	For details, see
Alarm messages	Alarm banner	[ALARM].	The [ALARM BANNER] display on page 115
Alarm messages	Alarm list	No default, although this object appears in the [ALARM MULTI-LINE] and [HISTORY] graphic libraries.	The [ALARM] display on page 114, the [ALARM MULTI-LINE] display on page 115, and the [HISTORY] display on page 117
Alarm messages	Alarm status list	No default, although this object appears in the [STATUS] graphic library.	The [STATUS] display on page 116
System activity	Diagnostics list	[DIAGNOSTICS]	How the diagnostics list graphic object works on page 138
Information messages	Information message display	[INFORMATION]	How the information message display graphic object works on page 396
Local messages	Local message display	No default.	How the local message display graphic object works on page 272

The default alarm and information displays open automatically when the assigned tags match the messages' trigger values. The default diagnostics display opens automatically when system activity occurs. If desired, you can set up your own graphic displays to open automatically, instead of the default displays. You can also set up any of the displays to open when an operator presses a Goto display button or selects a display in the display list selector.

The operator can acknowledge alarm and information messages. The operator can clear alarm and diagnostics messages. The operator can sort alarms and reset their status.

View information about runtime communication errors

To show communication errors in the diagnostics list object, set up message routing so that messages are sent to the FactoryTalk View Diagnostics List.

For information about setting up diagnostics message routing, see Set up FactoryTalk Diagnostics on page 126.

Change languages

You can change languages at runtime. The languages available depend on what has been set up for the runtime application. There is a separate **Language Switch** button for each language that you can change to.

For information about setting up language switching, see Set up language switching on page 160.

To change languages

- Press a **Language Switch** button.

Text strings in the application change to the language specified by the button.

Work with components

This chapter describes:

- Which editors have components
- How to work with components
- How to print information in components

Editors that have components

With some editors, you enter information in a single window or a tabbed dialog box. Other editors allow you to create multiple components, such as graphic displays or message files. Each component is stored in a separate file, with its own name.

You can create components in these editors:

- **Graphics** (components include graphic displays, global object displays, and graphic libraries, each in their own folder)
- **Parameters**
- **Local Messages**
- **Information Messages**
- **Data Log Models**
- **Macros**
- **RecipePlus**

The **Explorer** window lists the components that you create under the icon for the editor you used to create the component.

You can use the **Images** editor to copy bitmap images into your application (but not to create images). Each image you copy is listed as a component under the editor.

To view a list of components for an editor

- Click the + symbol to the left of the editor icon, or double-click the editor name.

Basic operations

This section describes how to perform basic operations that are common to all components.

For information about selecting a printer and printing components, see Print on page 40.

For information on working with components in particular editors, see the chapters later in this guide.

Create a component

You might want to create a component.

To create a component, use one of these methods:

- Drag the component icon into the workspace.
- Right-click the editor and select New.
- From the File menu, select New, and then select the type of component to create.

Open a component

You might want to open a component.

To open a component, use one of these methods:

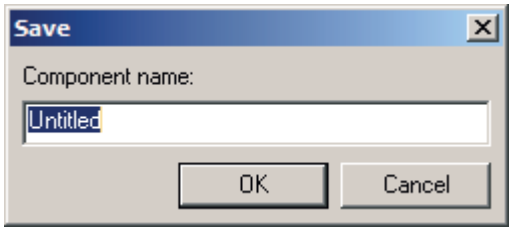
- Double-click the component.
- Right-click the component and select **Open**.
- Drag the component from the **Explorer** window to the workspace.

Save a component

The Save tool is available when the active component contains unsaved changes.

To save a component

1. From the **File** menu, select **Save**, or click the **Save** tool.



2. If this is the first time you're saving the component, type a name in the **Component name** box, and then click **OK**.

Close a component

You might want to close a component.

To close a component

- From the **File** menu, select **Close**, or click the Close button on the component's title bar.

Some components have a **Close** button 

that you can click to close the component.

Before the component closes you are prompted to save unsaved changes, if there are any.

Add a component to an application

You can use the same components in more than one application by adding components into an application. For example, if you want to use the same graphic display in Application A and Application B, create the display in Application A and then add the graphic display component from Application A to Application B. When you add the component into Application B, the changes you make to the component in Application B are not reflected in the component in Application A.

All the language strings associated with the component in Application A are copied into Application B. For example, if you add an information message file for which three sets of language strings have been defined, all three sets of strings are copied into Application B, regardless of which languages have been set up for Application B. For more information about using multiple languages, see Set up language switching on page 160.

To add a component from Application A to Application B

1. In Application B, right-click the type of editor that was used to create the desired component.
   For example, to add a graphic display component, right-click the **Displays** icon in the **Graphics** folder.
2. Select **Add Component into Project**.
3. In the dialog box, navigate to the component to add (in Application A's folder), and then click the component's file name.
   Click and Shift-click to select a group of components, or Ctrl-click to select multiple individual components.
   For information about application folders and files, see Work with applications on page 49.
4. Click **Open**. The components are listed under the editor in the **Explorer** window in Convert PanelBuilder and PanelBuilder32 applications on page 451.

If you later modify the component in Convert PanelBuilder 1400e applications on page 438, you can add the component into Convert PanelBuilder and PanelBuilder32 applications on page 451 again using the same steps as described above.

Use Add Component into Project with graphic displays, graphic libraries, and global object displays

The **Add Component into Project** menu command is also useful for:

- Adding graphic displays to your application's **Libraries** folder
- Using libraries as graphic displays in your application
- Moving libraries into the Displays folder so you can export their strings for translation
- Creating global object displays

For information about graphic libraries, see Use graphic libraries on page 262.

For information about global object displays, see Create global object displays and base objects on page 380.

Add Component Tool

Add Component Tool programmatically adds components to an existing HMI project without having to use FactoryTalk View Studio. FactoryTalk View Studio must be installed on the computer where the commands are run.

For FactoryTalk View ME version 15.0 or later, use the tool located in C:\Program Files (x86)\Rockwell Software \RSView Enterprise\SEAddComponent.exe.

The tool has the following command line usage:

```
SEAddComponent applicationName applicationType areaName componentListFile [/
U username] [/P password] [overwrite]
```

User authentication

Add **/U** and **/P** parameters to a specific user for user authentication. The user must have the application level Read, List Children, Create Children, Write, and Delete security permissions to run the command and add components successfully. Azure AD or OpenID Connect user accounts are not supported.

/U username

This parameter is an optional parameter that specifies the user name to sign in.

/P password

Parameters are case-insensitive. If a specified value includes a space, enclose the value in quotation marks.

NOTE: If **/U** and **/P** are omitted and only single sign-on is enabled, it can do the user authentication. First, it uses the current user who signed in to FactoryTalk Directory to do the authentication. If no user is signed in, it will use the current Windows system signed-in user to sign in FactoryTalk Directory and do the authentication with that user. When single sign-on is disabled, use **/U** and **/P** to specify a valid user to run the command.

Command parameters

- **applicationName** is the name of the FactoryTalk View ME application that contains the HMI project where components will be added. Quotations will be needed if the name contains any spaces.
- **applicationType** is **ME**.
- **areaName** is the name of area where the HMI project is located. For ME, set the area name to /.
- **componentListFile** is the full path to an XML file containing a list of components to add. Quotations will be needed if the path contains any spaces.
- **overwrite** is an optional parameter that, if present, will allow existing component files to be overwritten. If this parameter is not present, the existing component files will not be overwritten.

Example

```
SEAddComponent MyApplication ME / "C:\MEFiles\Files.xml" overwrite
```

Remarks

- The tool must be run on the same computer where the FactoryTalk View Studio is installed.
- The component file must not have the **Hidden** file attribute set.
- The component file name must not contain any of the following characters:
  % . " : ? * < > # { } - [ ] ' \ / ; = & +
- A log file is generated describing the results of the tool execution. This log file is saved in the same folder as the component list file and it is named **$AddComponent.txt**.
- When importing a tag CSV file, a log file is saved in the same folder as the component list file. The log file is named ***HMIProject*.log** where ***HMIProject*** is the HMI project name. For example, for the HMI project TagImportServer, the corresponding log file is TagImportServer.log.
- When the tool has completed execution, it returns one of the following codes:
  - 0 = successful completion
  - 1 = any error occurred during execution
- When adding a language file, the HMI server must be loaded and in the **Active** state.

Component list file

The component list file supplied to the Add Component Tool is an XML file consisting of a list of component category headings and component file names.

The file has the following format. The **<category>** and **<component>** sections can be repeating sections.

```
<hmi>

<category type="TYPE" folder="FOLDER">

<component file="NAME"/>

</category>

</hmi>
```

- **TYPE** is one of the component category types.
- **FOLDER** is the name of the folder where the component files in this category is located. This is an optional attribute. If this attribute is not present, the component files are assumed to be located in the same folder as this component list file. The folder attribute can be specified either as an absolute folder (for example, *C:\MEFiles\gfx*) or as a relative folder (for example, *.\gfx*). If it is a relative folder specification, it is assumed to be relative to the folder where this component list file is located.
- **NAME** is the name of the component file without a folder specification (for example, *pump.gfx*).

Examples

Adding graphic displays, global object displays, and images to an existing application:

When adding graphic displays or global object displays, you can specify a folder to hold them. If the folder doesn't exist, it will create one for you.

```
<hmi>

<category type="Graphics;Displays" folder="C:\MEFiles\Displays">

<component file="overview.gfx" folder="overview"/>

<component file="pump house.gfx" folder="pumps"/>

</category>

<category type="Graphics;Global Objects" folder="C:\MEFiles\GOs">

<component file="button.ggfx" folder="buttons"/>

<component file="panel.ggfx" folder="panels"/>

</category>

<category type="Graphics;Images" folder="C:\MEFiles\Images">

<component file="logo.bmp"/>

<component file="pump.bmp"/>
```

```
</category>

</hmi>
```

Adding folders under Displays and Global Objects of an existing application:

```
<hmi>

<category type="Graphics;DisplayFolders">

<component folder="line1"/>

<component folder="line2"/>

</category>

<category type="Graphics;GlobalObjectsFolders">

<component folder="common1"/>

<component folder="common2"/>

</category>

</hmi>
```

Adding XML files to an existing application:

```
<hmi>

<category type="Graphics;Displays" folder="C:\MEFiles\Displays">

<component file="overview.xml"/>

</category>

<category type="Graphics;Global Objects" folder="C:\MEFiles\GOs">

<component file="common.xml"/>

</category>

</hmi>
```

Adding a language file to an existing application:

```
<hmi>

<category type="Language;Files" folder="C:\MEFiles">

<component file="Excelname.xls"/>

</category>

</hmi>
```

Adding alarm setup files to an existing application:

```
<hmi>
<category type="Alarm Setup;Files" folder="C:\MEFiles">
<component file="Alarmfiles.xml"/>
</category>
</hmi>
```

Adding information messages to an existing application:

```
<hmi>
<category type="InfoMessages;Tables" folder="C:\TagFile\InfoMessages">
<component file="me_aa.ifm"/>
</category>
</hmi>
```

Adding recipe files to an existing application:

```
<hmi>
<category type="RecipePlus;Editor" folder="C:\TagFile\RecipePlus">
<component file="ME_RecipePlus.rpp"/>
</category>
</hmi>
```

The component category types and their associated application tree folder names are defined in the following table.

Component Category Type	Application Tree Folder Name
Graphics;Displays	Displays
Graphics;DisplayFolders	Displays
Graphics;Images	Images
Graphics;Global Objects	Global Objects
Graphics;GlobalObjectsFolders	Global Objects
Parameter;Files	Parameters
MessageServer;Tables	Local Messages
TagDb;Files	Tags
Macro;Files	Macros
Data Logging;Models	Data Log
InfoMessages;Tables	Information Messages

Component Category Type	Application Tree Folder Name
RecipePlus;Editor	Recipe Plus Editor
Alarm Setup;Files	Alarm Setup
Language;Files	

Delete a component

Deleting components deletes them from the **Explorer** window and from the hard disk.

To delete a component

1. In FactoryTalk View Studio, in the **Explorer** window, right-click the editor icon and select **Delete**.
2. In the component selection dialog box, select the components to be deleted and click **OK**.
3. Click the **Yes** button to confirm the deletion.

Tip: To delete a component, you can also right-click the component in the **Explorer** window and select **Delete**.

Remove a component

Removing components removes them from the **Explorer** window but leaves them on the hard disk.

To remove a component

1. In FactoryTalk View Studio, in the **Explorer** window, right-click the editor icon and select **Remove**.
2. In the component selection dialog box, select the components to be removed and click **OK**.
3. Click the **Yes** button to confirm the removal.

Tip: To remove a component, you can also right-click the component in the **Explorer** window and select **Remove**.

Rename a component

The following components can be renamed:

- Displays
- Libraries
- Images
- Parameters
- Local Messages
- Information Messages
- Data log Models
- Macros

To rename a component

1. Select the component in the **Explorer** window.
2. Right-click the component and select **Rename**.

3. In the **Rename** dialog box, type the new name in the **To** text box.
4. Click **OK.**

Tip: The rename function changes the component name shown in the **Explorer** window and the file name listed in Windows Explorer.

Rename components within FactoryTalk View Studio. Do not rename files using Windows Explorer.

A component name and its path can be up to 256 characters.

Duplicate a component

The **Duplicate** option is useful for creating multiple similar components. For example, you could create a graphic display to use as a template, then duplicate the display each time you want to use the template.

To duplicate a component

1. Right-click the component, and then click **Duplicate**.
2. In the **Component** name box, type a name for the duplicate component.
3. Click **OK**.

Filter a component

Filter a component to show only the ones that meet the specific criteria in the Explorer window. You can filter:

- **Displays**
- **Global Objects**
- **Parameters**
- **Computers and Groups**
- **Networks and Devices**
- **Users and Groups**

The filter applies to the current application only. Reopening the application or opening a new one will clear the filter, and all the components show in the Explorer window.

To filter a component:

1. In the Explorer window in FactoryTalk View Studio, right-click a component (for example, **Displays**) that you want to filter, and then click **Filter**.
2. In the **Filter** dialog box, select a filter condition, and then enter a filter term.
3. Click **OK**.

   A filter icon appears beside the component. The filtered components remain in their original folders.

Tip: To clear the filter, right-click the component, and then click **Clear Filter**.

Use graphic displays

This chapter describes the **Graphics**, **Images**, and **Local Messages** editors in the **Graphics** folder and outlines how to:

- Use the **Graphics** editor
- Set up graphic displays
- Create a background for your displays
- Use graphic libraries
- Import graphic images
- Create local messages in your displays
- Print graphic displays at runtime

Before creating graphic displays, specify project settings. Project settings determine important aspects of your graphic display such as size and position. For more information, see About project settings on page 57.

For information about global object displays and the **Parameters** editor, see Use parameters and global objects on page 374.

About graphic displays and graphic objects

A graphic display represents the operator's view of plant activity. The display can show system or process data and provide operators with a way to write values to an external data source. The data source can be memory or a device such as a programmable controller or an OPC server.

Operators can print the display at runtime to create a visual record of tag values, object states, or current messages.

The elements that make up a graphic display are called graphic objects. You can create objects in the **Graphics** editor, or copy them from a global object display, from a graphic library, or from another application.

For information about creating and copying graphic objects, see Use graphic objects on page 275.

You can use up to 1,000 tags per graphic display. This limit includes the tags contained in embedded variables.

PanelView Plus 7 Standard applications have a limit of 100 displays and PanelView Plus 6 Compact applications have a limit of 25 displays.

Before you begin

Before you begin, plan your displays. Think about what information that the operator needs to see, and the best way to provide the information. For example:

- Does the operator need to know the exact speed of a conveyor belt, or just whether the belt is moving, jammed, or stopped?
- Do different users need to have access to different types of information?
- Do you need to limit access to certain types of information?

Also consider the runtime environment and how the operator will use the application:

- Does the runtime computer have a touch screen, mouse, keyboard, or some combination of these?
- How will the operator navigate through the displays of the application?
- Will the application be available in multiple languages?

Review the chapters on planning, security, and navigation before you begin creating displays. Browse through the sample applications for design ideas. Map out a display hierarchy. Then create a graphic display to use as a template.

The time you spend planning your displays will make your application easy to use and will save you time in the long run.

For information about	See
Plan your displays and creating a template	Plan applications on page 42
Set up application security to control access to displays	Set up security on page 143
Set up display navigation and creating a display hierarchy	Set up display navigation on page 171
Set up how objects are used at runtime	Set up graphic objects on page 319

Graphics editor

The **Graphics** editor opens when you create or open a graphic display, global object display, or graphic library. Each display is stored in the **Displays** folder. You can open and work on multiple graphic displays at the same time.

Create a graphic display

Use the following steps to create a graphic display.

To create a graphic display

1. In the **Graphics** folder, do one of the following:
   - Right-click **Displays** and select **New**.
   - Click the **New Display** tool .
   - Drag and drop the **Displays** icon into the workspace.
2. From the **Edit** menu, select **Display Settings** to open the **Display Settings** dialog box and specify settings for the display.
   For more information, see Set up graphic displays on page 258.
3. Create the objects that you want to put on the display.
   For information about creating graphic objects, see Use graphic objects on page 275.
4. From the **File** menu, select **Save**, or click the **Save** tool .
5. In the **Component name** box, type a name for the display, and then click **OK**.

The display is added to the list in the **Displays** folder. The display is created as a Replace display by default, but you can change it to an On Top display in the **Display Settings** dialog box. For more information about display types, see Replace displays on page 259 and On Top displays on page 259.

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

Open a graphic display

Use the following steps to open a graphic display.

To open a graphic display

- In the **Graphics** folder, open the **Displays** folder and then double-click the display name, or right-click the display name and select **Open**.
  You can also drag and drop the display from the **Explorer** window to an empty area in the FactoryTalk View Studio workspace.

The **Graphics** editor has special items on the **View** and **Edit** menus, and the toolbox.

For details about using the **Graphics** editor, see Help.

Import and export graphic displays

Each graphic display's information is contained in a file called *Displayname*.gfx.

The Graphics Import Export Wizard in FactoryTalk View Studio allows you to export this information to an XML file, or to import a graphic display XML file.

You can export the display information, import the graphic display XML file to another FactoryTalk View application, and then modify it to suit your requirements. Or, you can modify the file before importing it.

You can use the Graphics Import Export Wizard to import graphic display information that has been created using an external programming tool or editor, or you can import a FactoryTalk View XML file.

For more information about importing and exporting graphic display files, see Import and export graphics XML files on page 433.

Tools and tips for working in the Graphics editor

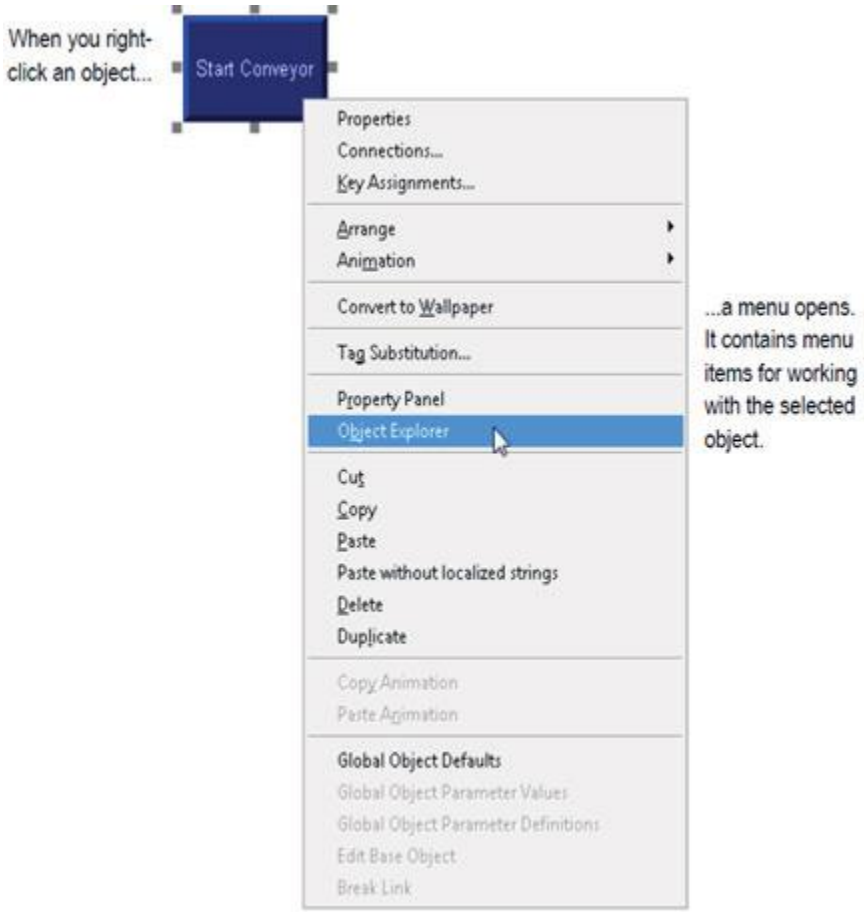
This section describes features of the **Graphics** editor that help you create your displays. It describes how to:

- Use context menus and toolbars to perform actions quickly.
- View displays in grayscale to emulate the appearance of a PanelView Plus terminal.
- Set up a grid that you can use to position and size objects precisely.
- Zoom in when you want to look at details closely.
- Correct mistakes.
- Test your displays as you work.

Use context menus

No matter where you are in the **Graphics** editor, you can open a menu by clicking the right mouse button. This is often quicker than moving the mouse up to the menus at the top of the screen.

The items on the menu depend on the cursor's location. For example, when you right-click an object, the menu contains the most common actions you can perform on that object.



Use the toolbars

The toolbars provide another convenient way to perform an action quickly. You can customize your workspace by dragging the toolbars to any location on the screen. You can also dock them to the inside edges of the FactoryTalk View Studio workspace.

To show a toolbar

- From the **View** menu, select **Toolbars**, and then select the toolbar to show.

The menu shows a check mark beside the toolbar that is open.

To close a toolbar

- From the **View** menu, select **Toolbars**, and then select the name of the toolbar to close, or click the toolbar's **Close** button.

Show displays in grayscale

Applications running on the PanelView Plus grayscale terminals do not show in color. If you are developing applications for these terminals, show displays in grayscale to see how the displays appear at runtime.

To show displays in grayscale

- Select **View > Show Displays in Grayscale**.

All open displays will change from full color to grayscale.

Tip: Show displays in Grayscale doesn't work:

- On third-party ActiveX objects.
- In the Test Application mode.

When the display's background color or the object's back style is gradient.

Use the grid

To size and position objects precisely, use the grid. You can change the grid settings at any time during the drawing process.

The grid can be active or passive. When the grid is active, all the objects you draw or position are pulled to the closest grid point. This makes it easy to align and size objects. When the grid is passive, it is visible but does not affect the position of your objects.

Make the grid passive to position an element between the grid lines. Make the grid active, and the next object you draw or place is automatically aligned with the grid. Making the grid active does not affect the placement of existing objects.

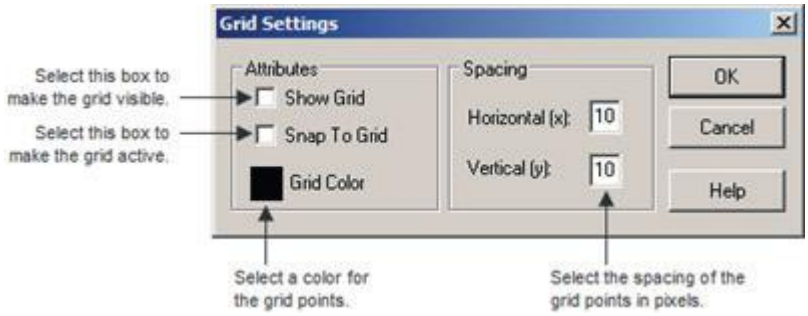
The grid is visible during application development only. It is not visible at runtime.

Set up the grid

Use the Grid settings dialog box to set up the grid.

To set up the grid

1. From the **View** menu, select **Grid Settings**, or right-click an empty area of the display and select **Grid Settings**.



2. Specify the color and spacing of the grid points.
3. To turn on the grid, click **Show Grid.**
   When the **Grid Settings** dialog box is not open, you can turn on the grid by clicking **Show Grid** on the **View** menu.
4. To make the grid active, click **Snap To Grid.**

When the **Grid Settings** dialog box is not open you can make the grid active by clicking **Snap On** on the **View** menu.

5. Click **OK**.

Make the grid passive

You might want to make the grid passive.

To make the grid passive

- In the **Grid Settings** dialog box, clear the **Snap To Grid** checkbox, or from the **View** menu select **Snap On** to turn off the option.

There is no check mark beside the menu item when it is turned off.

Turn off the grid

You might want to turn off the grid.

To turn off the grid

- In the **Grid Settings** dialog box, clear the **Show Grid** checkbox, or from the **View** menu, select **Show Grid** to turn off the option.

Zoom in and out

To magnify or reduce your view of a graphic display, use Zoom In and Zoom Out. Zoom In magnifies objects and Zoom Out reduces magnification.

To magnify an area

1. Select the object or objects to magnify.
2. From the **View** menu, select **Zoom In**, or on the toolbar, select .

Use the scroll bars to view the desired area of the graphic display.

To reduce the magnification of a display

- From the **View** menu, select **Zoom Out**, or on the toolbar, select .

Correct mistakes

If you change your mind about something you did, you can undo the action. If you change your mind again, you can redo the action.

You can undo and redo all the operations you performed since you last saved the display, one operation at a time.

The operations you perform between opening and closing a dialog box are treated as a single operation. Operations you perform in the **Property Panel** are treated as separate operations.

To undo an operation

- From the **Edit** menu, select **Undo**, or click the **Undo** tool .

To redo an operation

- From the **Edit** menu, select **Redo**, or click the **Redo** tool .

Test your displays as you work

To test the objects in your displays as you work, use the **Test Display** tool to switch to test mode. When you are finished testing, switch back to edit mode to continue editing.

Tip: Test mode is not the same as running the display. It does not change the appearance or position of the display as set up in the **Display Settings** dialog box. Alarm and information messages are not shown, although if communications are set up, tag values are read and written. Display navigation, data logging, and macros do not work in this mode. If you want to test these features, test the application as described on Test your application on page 177.

If you set up local messages and graphic objects to use multiple languages, the messages and objects are shown in the current application language during test mode.

To use test mode

- From the **View** menu, select **Test Display**, or click the **Test Display** tool .

To return to edit mode

- From the **View** menu, select **Edit Display,** or click the **Edit Display** tool .

Use the Diagnostics List when in test mode

The **Diagnostics List** shows messages about system activities when you test your graphic displays. You can specify the types of messages to show in the **Diagnostics List**, move the list, resize it, and clear the messages in it.

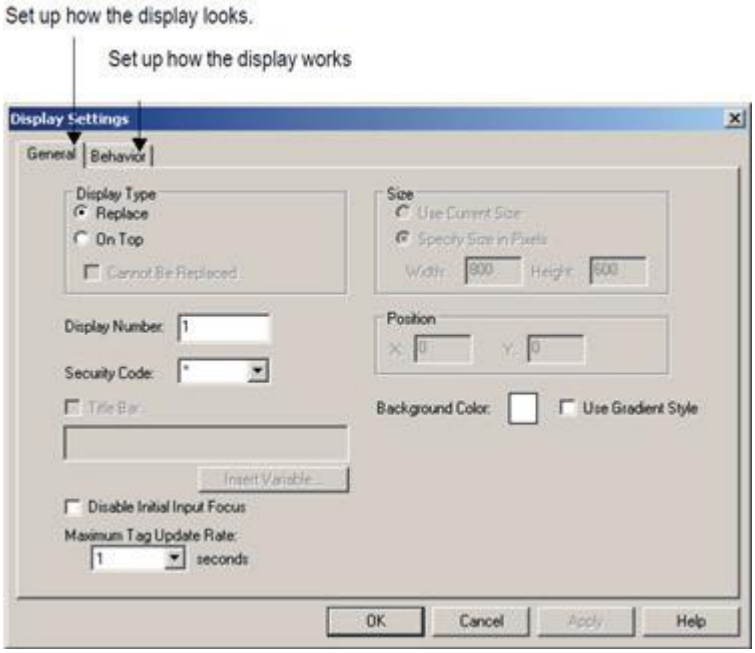
For information about specifying the types of messages to show in the **Diagnostics List**, see Show diagnostics messages during application development on page 134.

Set up graphic displays

To set up a graphic display, specify its type, background color, and runtime behavior in the **Display Settings** dialog box. You can specify and edit the display settings at any time while you work on your displays.

To set up graphic displays

- From the **Edit** menu, select **Display Settings**, or right-click an empty area of the display and then select **Display Settings**.



For details about the options in the **Display Settings** dialog box, see Help.

Replace displays

Replace is the default display type. Replace displays are full-sized displays. They use the project window size specified in the **Project Settings** editor.

For more information about the project window size, see Project window size and runtime screen resolution on page 57.

At runtime, you can have only one Replace display open at a time. When the operator opens a Replace display, the following proceedings occur.

- The Replace display that was open closes.
- On Top displays that do not use the **Cannot Be Replaced** option are closed.
- The new Replace display opens.
- On Top displays that use the **Cannot Be Replaced** option remain open, on top of the new Replace display.

If the operator attempts to open a Replace display that is already open (for example, using a **Goto display** button to which the same display is assigned), the display does not close and FactoryTalk View sends an error message to FactoryTalk Diagnostics.

On Top displays

Use the On Top option to create pop-up displays that open on top of the current Replace display. Usually, On Top displays are smaller than Replace displays, so the operator does not lose track of display navigation.

You can open multiple On Top displays. If more than one On Top display is open, the display that has focus, or had the most recent focus, appears on top.

When an On Top display closes, the display that had the most recent focus appears on top.

Use the Cannot Be Replaced option if you want the On Top display to remain open when a new Replace display is opened.

On Top displays do not have a Close button in the title bar. Be sure to create a close button graphic object in On Top displays so the operator can close them.

The operator cannot move an On Top display by dragging its title bar. The runtime position of the display is fixed (according to the position settings defined for the display).

You can specify unique titles for On Top displays. You can use embedded variables in the title, and the title text can switch languages at runtime.

Resize displays

The project window size is used for all Replace displays. If you change the project window size after you have designed any graphic displays, you have the option of scaling graphic displays.

If you choose to scale graphic displays, all Replace and On Top displays are resized and the objects in them are scaled to fit the new size. You can also specify whether to scale the font size and border size of the graphic objects, and the size of graphic images in the Images folder.

If you choose not to scale displays when you change the project window size, Replace displays are resized, but any objects on the displays remain the same size and in the same position as before. On Top displays are not resized.

For information about specifying the project window size, see Project window size and runtime screen resolution on page 57.

Change the display area while working

You can change the display area of a display while you're working on it by dragging the border of the display area. For example, you might want to make the display area smaller so you can see parts of two displays in order to drag and drop objects between them.

Changing the display area in this manner does not change the display size for Replace displays. For On Top displays that are sized with the **Use Current Size** option, dragging the border does resize the display.

Another way to arrange multiple displays while working on them is to use the options on the Windows menu. For example, Tile Horizontal arranges all the open displays with as much of the top part of each display showing as possible.

Specifying the display type

You can set up the following display types in a FactoryTalk View application:

Replace is the default display type. If you want the graphic display to replace other open displays when it opens, use the **Replace** option. A display of Replace type closes displays that it overlaps. You don't need to run separate commands to close other open displays.

Overlay graphic displays will layer with other displays, overlapping some and being overlapped by others as the focus changes between open displays. If the display doesn't need to replace other displays or appear on top at all times, use the **Overlay** option.

For overlay displays, if you want the display always at the back, you can select the checkbox, **Keep at Back**. However, it is recommended that you use the On Top display type to control the layering of displays.

Overlay displays always appear behind On Top displays, and are replaced by Replace displays. Use the Overlay type with care; keeping multiple displays open at runtime can affect system performance.

On Top graphic displays will stay on top at all times, even if another display has focus. If you want the display to always appear on top, use the **On Top** option.

If more than one graphic display of the On Top type is open at once, the display that appears on top is the one that has focus, or the one that had focus most recently.

You can use the **PullForward**, **PushBack**, and **SetFocus** commands to cycle through multiple On Top and Overlay displays at runtime. For more information about these commands, see the FactoryTalk View SE Help.

Create a background for your display

You can create a background for your graphic display by converting graphic objects to wallpaper. When objects are converted to wallpaper, they are locked into position and become an unchanging background for the other objects on the display.

Convert objects that do not need to be animated or updated with tag values can significantly improve the runtime performance of a graphic display.

Objects that have been converted to wallpaper cannot be selected or edited until you unlock the wallpaper. Also, animations attached to the wallpaper objects are not in effect. However, animations are restored when you unlock the wallpaper.

Similarly, any tags or expressions assigned to an object become inactive when the object is converted to wallpaper. Connections are restored if you unlock the wallpaper.

To manage a number of objects easily, group the objects, and then convert the group to wallpaper.

Convert graphic objects to wallpaper

You may need to convert a graphic object or a group of objects to wallpaper.

To convert a graphic object or group to wallpaper

1. Select the object or group you want to convert to wallpaper.
2. Do one of the following:
   - Select **Edit > Wallpaper > Convert to Wallpaper**.
   - Right-click the selected object or group and select **Convert to Wallpaper**.

Tip: To unlock the wallpaper objects, right-click any empty area of the display and select **Unlock All Wallpaper**. Alternatively, select **Edit > Wallpaper > Unlock All Wallpaper**.

For information about selecting objects, see Select and deselect objects on page 291.

Graphic libraries

There are two graphic libraries in FactoryTalk View: Symbol Factory and Graphics Library.

Graphics Library comes with a set of libraries, contained in the **Libraries** folder. As with a public library, the graphics libraries can provide you with source materials and reference information.

Each graphic library consists of a graphic display, with the file extension .gfx. The library displays contain graphic objects that you might find useful in your own application. Many of the objects are preconfigured with animation. For example, see the Conveyor parts library.

You can:

- Look at the objects and displays to get ideas for your own application.
- Drag and drop (or copy and paste) objects from the libraries to your graphic displays.
  For information about copying objects from a library to your graphic display, see Copy objects on page 307.
- Use the objects as they are or change them to suit your needs.
- Create your own libraries of objects.
- Create libraries of displays that are translated into multiple languages.
- Use libraries as displays in your application.

For more information about Symbol Factory, see Work with Symbol Factory on page 265.

Open a graphic library

Use the following steps to open a graphic library.

To open a graphic library

1. In the **Explorer** window, open the **Graphics** folder, and then open the **Libraries** folder.
2. Double-click the library name, or right-click it and then select **Open**.

Create graphic libraries

You can create a graphic library in the **Libraries** folder, or create a graphic display in the **Displays** folder and then add the display to the **Libraries** folder.

To create a graphic library

1. In the **Graphics** folder, right-click **Libraries** and select **New**.
2. Create the objects you want to put in the library.
   For information about creating graphic objects, see Use graphic objects on page 275.
3. From the **File** menu select **Save**, or click the **Save** tool  .
4. In the **Component name** box, type a name for the library, and then click **OK**.
   The library is added to the list in the **Libraries** folder.

You can also create a graphic display and then use **Add Component into Project** to add the display to the **Libraries** folder.

Add a graphic display to the library

You might want to add a graphic display to the library.

To add a graphic display to the library

1. Create the graphic display.
2. In the **Explorer** window, in the **Graphics** folder, right-click the **Libraries** icon.
3. Click **Add Component into Project**.
4. In the dialog box, navigate to the **Gfx** folder and select the .gfx file for the display to use.
   By default, the **Gfx** folder is in **\Users\Public\Public Documents\RSView Enterprise\ME\HMI projects ***Application Name*
   where *Application Name* is the name of your application.
5. Click **Open**.

The display is copied into the **Libraries** folder.

Use libraries as displays in your application

The graphic libraries are available on the development computer, but do not appear at runtime. To use a library as a graphic display at runtime, you must add the library into your application's folder of graphic displays.

If the library does not contain strings for languages supported by the current application, the undefined strings are shown with question marks (?).

To use a library as a display in your application

1. In the **Explorer** window, in the **Graphics** folder, right-click the **Displays** icon.
2. Click **Add Component into Project**.
3. In the dialog box, navigate to the **Libraries** folder and select the .gfx file for the library to use. The **Libraries** folder is **\Users\Public\Public Documents\RSView Enterprise\ME**
4. Click **Open**.

The library is copied to the **Displays** folder.

Use libraries to store displays with multiple languages

The maximum number of languages an application can use is 40 during development, and 20 at runtime. Since the libraries are stored outside of the HMI project folder and are available to all applications, they do not have this limit. However, since they are not stored in the HMI project folder, you cannot export the strings in library displays for translation. Instead, export a graphic display for translation, import the translated strings, and then add the display into the library.

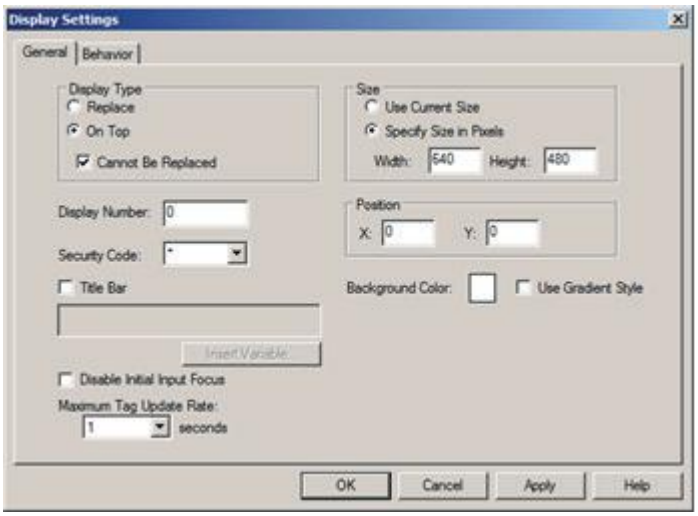
When you use a library display in your application, make sure you add to your application the languages that you plan to use from the library.

To save a library display with multiple languages

1. Create a graphic display.
2. Export the text for your application, which will include the text used in the graphic display.
3. Translate the text strings into each desired language, saving the file with a new name for each language.

4. Import the files for all the new languages.
5. Add the graphic display to the **Libraries** folder.

   The library is created with the option **Support Multiple Languages** selected in the **Display Settings** dialog box).



For detailed information about importing and exporting to use multiple languages, see Set up language switching on page 160.

To save a library display with more than 40 languages

1. Create a display in one application, with up to 40 languages, and add it to the library, as described in the previous section.
2. Add the display into a new application, with up to 40 different languages in the new application.
3. Export the text for your application, which will include the text used in the graphic display.
4. Translate the text strings into each desired language, saving the file with a new name for each language.
5. Import the files for the new languages.
6. Add the graphic display (with the same name as the display in step 1) to the **Libraries** folder.

The new languages are added to the library. Make sure the display contains the same objects as the original display. Otherwise the new display will overwrite the previous display, and all the original language strings will be undefined.

When you open a library that supports multiple languages, the strings are shown using the current application language, if available. If the library does not contain the current language, the strings are shown as single question marks. Similarly, if the library contains the language but not all the strings are defined for the language, undefined strings are shown as single question marks.

For more information about using multiple languages with graphic libraries, see Set up language switching on page 160.

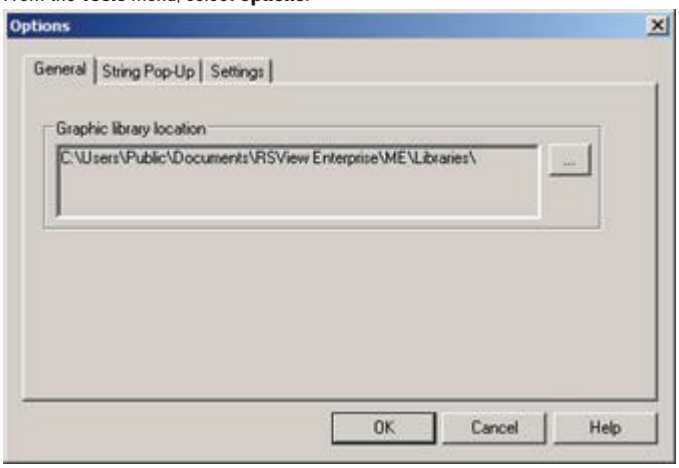
Specify a path for graphic libraries

The **Libraries** folder is in **\Users\Public\Public Documents\RSView Enterprise\ME**. You can specify a different directory in which to store library components. If you specify a different directory, FactoryTalk View saves graphic libraries that you add or create in the new directory. Similarly, when you open a library component FactoryTalk View looks for the component in the specified directory.

If you specify a different directory but want to use the libraries that come with FactoryTalk View, use My Computer or Windows Explorer to copy the library component files into the directory you have specified, or else change the path back to the default path when you want to open a FactoryTalk View library component.

To specify a path for graphic libraries

- From the **Tools** menu, select **Options**.



For details about using the **Options** dialog box, see Help.

Work with Symbol Factory

Symbol Factory is a third-party graphics library that provides a large number of quality graphic elements for industrial automation, including pumps, pipes, valves, tanks, mixers, motors, ducts, electrical symbols, flow meters, material handling, sensors, PLCs, transmitters, and ISA symbols. These graphics can be used on graphic displays.

To open Symbol Factory, use one of these methods:

- In the **Explorer** window, double-click **Symbol Factory**.
- Select **Objects > Symbol Factory**.
- Drag from the toolbox.

Import images into your application

Use the **Image Browser** to:

- Import images into the application
- Select the image to use on a graphic object
- Delete images from the application

You can use these types of external graphic files on graphic displays:

- .bmp - bitmap images
- .jpg - JPEG images
- .png - PNG images
- .dxf - AutoCAD files
- .wmf - Windows metafiles

To use bitmap, JPEG, and PNG images:

- Import the images first, and then place them on your displays as needed. This method is useful for images that you use to illustrate your displays.
- Use the **Image Browser** to import images as needed while setting up your graphic objects. This method is useful for images that you use as labels on your graphic objects.
- Copy and paste images from the graphic libraries.
- Copy and paste images from one application to another.

Tip:

- To add an image to a graphic display, first create an image object.
- To place an image on a graphic object, use the **Property Panel**, or open the object's **Properties** dialog box. If you use the **Property Panel**, you can place the same image on multiple selected objects simultaneously.
- To use images that have more than 256 colors, before importing the images, set up your video adapter to show more than 256 colors. This will ensure that the colors of imported images appear the same as in the original.
- To remove a bitmap, JPEG, or PNG image from an application, right-click the image's name in the **Images** folder and select **Remove**. Remove will not delete the file from the disk.
- To delete a bitmap, JPEG, or PNG image from an application, right-click the image's name in the **Images** folder and select **Delete**, or in the **Image Browser**, select the image, click **Delete**, and click **Yes**

For information about using the **Image Browser**, see Use the Image Browser to import images on page 267.

For information about copying and pasting objects, see Copy objects on page 307.

For more information about using .dxf and .wmf files, see Import a .dxf or .wmf graphic image on page 289.

For more tips about using images, see Tips for using images on page 268.

Bitmap images that come with FactoryTalk View Studio

FactoryTalk View Studio comes with sets of bitmaps that are useful for illustrating graphic objects and displays:

- Arrows
- DIN symbols
- ISA symbols
- Keyboard button symbols such as Enter and Page Up
- Parts such as buttons, conveyors, pipes, tanks, and valves

The symbols and most of the arrows are monochrome (that is, use only two colors, one for the line and one for the fill).

You can set up the line and fill colors for the monochrome images that you use for your objects.

For color images, including JPEG images, you can specify whether to use a gradient, transparent, or solid background fill. If you select the transparent background fill style, the black portions of the image become transparent.

The bitmap files that come with FactoryTalk View Studio are stored in **\Users\Public\Public Documents\RSView Enterprise\Images**

The images are in folders called Arrows, DIN, ISA, ListKey, and Parts.

For detailed information about setting up objects, see Help.

Import bitmap, JPEG, and PNG images

To use a bitmap, JPEG, or PNG image, import the image and then place it on the display as needed.

FactoryTalk View installs a set of commonly used PNG images in C:\Users\Public\Documents\RSView Enterprise \Images\Rockwell Library Images. Import them to the **Images** folder to simplify your application development.

To import a bitmap, JPEG, or PNG image

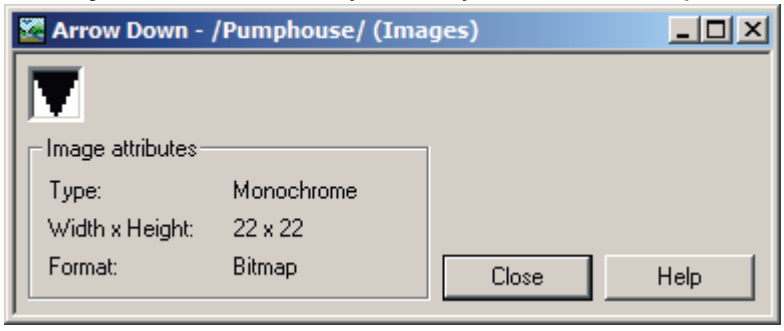
1. In the **Graphics** folder, right-click **Images** and then select **Add Component into Project.**
2. In the **Files of type** box, select the type of image to add.
3. Navigate to the directory where the .bmp, .png, or .jpg file is stored, and then click the file name. Shift-click or Ctrl-click to select additional files.
4. Click **Open** to add the selected files to the list in the **Images** folder.

View an image that you have imported

You might want to view the image you have imported.

To view an image that you have imported

- In the **Images** folder, double-click the image name, or right-click it and then click **Open**.



Use the Image Browser to import images

Use the **Image Browser** to import images as needed while setting up a graphic object.

To import an image with the Image Browser

1. Open the **Image Browser** using one of the following 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 **Image** and then click the **Browse** button.

2. Click **Add from File** to open a Windows **Open** dialog box.
3. Select the images to import and click **OK.**
   The dialog box closes and the images appear in the **Image Browser's Select image** list.
4. Click **OK**.

You can also use the **Image Browser** to import images from Symbol Factory. For more information, see Add an image from Symbol Factory to a graphic object on page 284.

For information about opening the **Properties** dialog box, see Use the Properties dialog box on page 295.

For information about opening the **Property Panel**, see Use the Property Panel on page 296.

You can also remove an image by right-clicking it in the **Images** folder and then select **Remove** or **Delete**. For more information, see Remove components on page 250.

Import a Symbol Factory object directly to the graphic display

Use the following steps to import a Symbol Factory object directly to the graphic display.

To import a Symbol Factory object directly to the graphic display

1. Open the graphic display you intend to put the object on.
2. Open Symbol Factory.
3. Browse through the **Categories** and select the symbol to be used.
4. (optional) Modify the symbol options below the symbol tab.
5. Drag and drop, or copy and paste a basic symbol to the graphic display.

Tip: It is not supported to drag and drop, or copy and paste an enhanced symbol directly to the graphic display. You can use Image Browser to add an enhanced symbol.

Tips for using images

Use bitmaps versus JPEG images

When deciding whether to use a bitmap image or a JPEG image, consider these points:

- FactoryTalk View supports 256 color (grayscale) and 16 million color JPEG images only. For all other color types, use bitmaps.
- For large color images (16 million color, 320 x 240 pixels or larger), JPEG images load faster than bitmaps. In all other cases, bitmaps load faster than JPEGs.
- Color JPEG images have a much smaller file size than the equivalent bitmap image, and therefore require less disk space at runtime.

Guidelines for using images

Images consume Windows resources, so when using graphic images use the lowest color depth possible.

The more colors that you use, the more memory is consumed, and the longer the image takes to load and show.

This bitmap type	Consumes this many bits per pixel
Monochrome	1

This bitmap type	Consumes this many bits per pixel
16 color	4
256 color	8 (1 byte)
64 K color	16 (2 bytes)
16 million color	24 (3 bytes)

For example, a 24-bit bitmap image that measures 800x600 pixels consumes 1440 KB of memory. If the bitmap color depth is decreased to 256 colors, the image might have minor color loss, but the new image uses only 480 KB of memory.

Use images with a similar pixel size to the size of the FactoryTalk View object on which the image will be placed. FactoryTalk View resizes the image to fit the object, but if you use an image that is much larger than the object, the display will be slow to open at runtime, due to the time required to resize the image.

Place a bitmap, PNG, or JPEG image on a display

You can place a bitmap, PNG, or JPEG image on a display by creating an image object. For more details, see .Create images on page 284.

Local messages

Use local messages to give the operator ongoing information about the status of devices and processes. For example, you might use local messages to describe the status of a device whose condition cannot be shown graphically with complete accuracy.

The messages you create in the **Local Messages** editor are shown on local message display objects on graphic displays. You can use multiple local message display objects in your application, and link each object to a different file of messages. Or, you can use the same file of messages for multiple local message display objects.

Tip: To give the operator information no matter which display is open, use information messages. For details about information messages, see Information messages on page 392.

Set up local messages

Use the following steps to set up local messages.

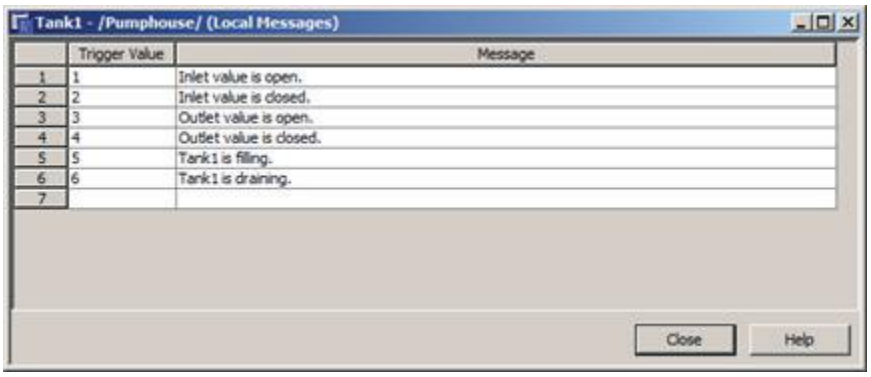
To set up local messages

1. In the **Local Messages** editor, set up the messages and their trigger values.
2. In the **Graphics** editor, create a local message display object on the graphic displays, in which you want the messages to appear at runtime.
3. Specify how the local message display looks, the message file to use with the display, and its tags.

For information about creating graphic objects, see Use graphic objects on page 275.

Local Messages editor

Use the **Local Messages** editor to create one or more files of local messages. Each file is stored in the editor's folder. You can open and work on multiple message files at the same time.



You can define up to 10,000 messages in each message file.

For details about using the **Local Messages** editor, see Help.

Prepare to set up local messages

As your application is running, information is continually sent to the data source about the state of various processes. For example, your application might be monitoring whether a valve is open or closed, or the temperature in a boiler. Values representing the status of these processes are sent to the data source.

To set up local messages, determine which tags associated with machine processes to monitor, and identify the values for those tags that will trigger local messages.

For information about creating HMI tags, see Use HMI tags on page 79.

How local messages work

The following are the key parts of the local message system.

- **Local message files** – text files containing lists of messages, with a numeric trigger value for each message
- **Local message display object** –a graphic object that shows local messages when the **Value connection** assigned to the object matches a message's trigger value
- **Value connection** – a tag or expression. When the value of this connection matches a message's trigger value, the local message display object shows the associated message.
- **Message File connection** – a string tag or expression. When variable local message file is used, if the value of this connection matches a local message file name, the local message display object uses this local message file.
- **Use Echo Message connection** – a string tag. When using an echo message, the triggered message will be written to the tag assigned to the connection. The echo message function works whether the local message display object is visible or hidden at runtime.

The local message display object always appears on the graphic display that it is placed in, whether there is a message to show. However, the operator does not see the message unless the object is located on the display the operator is viewing.

Example: Show local messages

This example shows how to notify the operator of the status of a hoist. In the example, the key parts of the local message system work together.

1. Create a tag called *Hoist_Status*. This tag points to an address in a programmable controller that is linked to sensors on the hoist. The tag has five possible values:

The tag has this value	When the hoist has this status
1	At bottom
2	Raising
3	Stopped between the top and bottom
4	Lowering
5	At top

2. In the **Local Messages** editor, create these messages with trigger values matching the values that will be sent to 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.

   Save the message file with the name *Hoist status*.
3. In the **Graphics** editor, create a local message display object. In the object's **Properties** dialog box, assign the **Hoist_Status** tag to the **Value** connection, and select the Hoist status message file.

At runtime, when the operator views the graphic display containing the local message display object, the status of the hoist is shown.

Local messages and trigger values

Create messages associated with each tag value that you want to inform the operator about. Assign each message a trigger value, and set up the data source to send the trigger value to the **Value** connection. You can use both HMI and data server tags

The trigger value 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 values of 10, 20, and 30.

If you use an analog tag or an expression, you can use any non-zero integer or floating point value to trigger an alarm. Floating point values are rounded to the nearest integer. For information about how values are rounded, see How values are rounded on page 80.

Trigger values cannot be 0. Digital tags have two possible values, 0 and 1. Therefore, if you use a digital tag you can only use the value 1 to trigger a message. If you want to use a digital tag to trigger two different messages, create an expression that adds 1 to the digital tag's value. That way, you can use the trigger values 1 and 2.

When the **Value** connection value is 0, the local message display object is cleared.

Create local messages in multiple languages

FactoryTalk View version 5.00 and later support local messages in multiple languages. When you create local messages, they are in the current application language. You can export the local messages for translation and then import them back into the application. For details, see Set up language switching on page 160.

For applications that will run in FactoryTalk View ME Station version 4.00, use the CurrentLanguage( ) expression function to specify message offsets in the local message file. In the file, divide your messages into sections for each language. For information about the CurrentLanguage( ) function, see Language function on page 363.

How the local message display graphic object works

When you open a graphic display at runtime, FactoryTalk View reads the value of the **Value** connection and updates the local message display object based on the value.

If a variable message file is used, FactoryTalk View reads the value of the **Message File** connection to select the local message file.

What is shown on the local message display?

- If the **Value connection** is unassigned, the display is filled with question marks (?).
- The **Value connection** is rounded to the nearest integer. If the value does not match any of the trigger values in the specified message file, the display is filled with question marks.
  For information about how values are rounded, see How values are rounded on page 80.
- If the message is too long to fit in the object, the last shown character is replaced with an asterisk (*).
- When the **Value connection**'s value is 0, the display is cleared.
- If you set up local messages in multiple languages, messages are shown in the current application language. When a language switch occurs, the message that was already in the local message display remains in the language that it originally appeared in. New messages are shown in the new language.
- If a variable message file is used, you can use a single local message display object to show the messages from multiple message files.
- If no message file is selected, or the selected message file is corrupt or does not exist, the following message shows on the local message display:
  "Referenced Local Message file is missing or corrupt."

Print displays

You can print your graphic displays on the development computer. This might be useful if you want other people to review the displays before implementing the application, or if you want to keep a visual record of the displays.

You can also print graphic displays at runtime, to provide a printed record of process values such as trend data.

For information about printing on the development computer, see Print on page 40.

Print displays at runtime

When you print a display at runtime, everything on the screen is printed, including the current display, pop-up windows, and any visible background applications.

For information about specifying which printer to use at runtime for applications that will run on a personal computer or a MobileView terminal, see Specify the printers to use at runtime on page 191.

For information about specifying printer options for applications that will run on a PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, PanelView Plus CE terminal, see the *PanelView Plus Terminals User Manual*.

To print graphic displays at runtime

Use one or both of these methods:

- Create display print buttons on the graphic displays you want to print. At runtime, the operator presses the buttons to print the displays.
  For information about creating graphic objects, see Use graphic objects on page 274.
- Assign a tag or expression to the **Remote Display Print** connection (in the **Global Connections** editor). When the value of the tag or expression changes from 0 to a non-zero value, the current displays are automatically printed.
  Program the data source to trigger the change as often as you want the data printed.
  For more information about setting up remote display printing, see Print displays on page 272.

Use graphic objects

This chapter describes the types of graphic objects and outlines how to:

- Create graphic objects, including drawing and ActiveX objects
- Select and deselect objects and use the **Object Explorer**
- Use the **Properties** dialog box and **Property Panel** to set up objects
- Color and name objects
- Test how objects look in different states
- Assign tags and expressions to objects
- Replace tags using tag substitution
- Use tag placeholders
- Move, copy, duplicate, resize, reshape, and delete objects
- Group and ungroup objects, and edit group objects
- Arrange objects and lock objects into position

For information about setting up graphic objects once you have created them, see Set up graphic objects on page 319.

For information about setting up global objects, see Use parameters and global objects on page 374.

Graphic objects

The elements that make up a graphic display are called graphic objects. Use objects to control your process, machines, and application.

FactoryTalk View comes with a complete range of configurable objects such as push buttons, list selectors, bar graphs, and trends. Some objects interact with the data source, allowing the operator to change or view tag values. For example, the operator can push a button to set a tag value to 1, causing a programmable controller to start a conveyor belt. Other objects are used to control your application. For example, there are button objects that you can use to change displays and scroll through lists.

FactoryTalk View also comes with drawing objects that you can use to illustrate your graphic displays. The drawing objects include text, bitmap images, and geometric and freehand shapes.

FactoryTalk View also supports the use of ActiveX objects — third-party software components that you can use to control processes and display information. The ActiveX objects available depend on which third-party applications are installed on your development computer. For example, products such as Microsoft Visual Basic®, Rockwell Automation, and Microsoft Office provide ActiveX objects that are configurable in FactoryTalk View Studio.

Connections

Many of the objects use connections to interact with the data source. A connection is the link between the object and the data source. Depending on an object's function, the object may have more than one connection. For example, a Momentary push button has a **Value** connection and an **Indicator** connection. The **Value** connection is used to set a value at the data source, and the **Indicator** connection is used to show the data source value in your graphic display.

By assigning tags or expressions to an object's connections, you control the flow of data between the application and the data source, which in turn controls your process or machines. Assign tags or expressions to an object's

connections on the **Connections** tab of the object's **Properties** dialog box (see Use the Properties dialog box on page 295), or on the **Connections** tab of the **Property Panel** (see Use the Property Panel on page 296).

Use graphic objects

The following tables will help you choose which objects to use to control your application, machines, and process. The tables group the objects according to function, provide an overview of what each object does, and list cross-references to more detailed information about using the objects.

Illustrate your displays

Use this graphic object	To do this
Text	Create text for labels or instructions on the display. See Create text on page 283.
Image	Place images in your display. For more information, see Create images on page 284.
Panel	Draw rectangles that have borders. See Create panels on page 285.
Arc	Draw an arc (a segment of an ellipse or circle's perimeter). See Create arcs and wedges on page 285.
Ellipse	Draw ellipses and circles. See Create ellipses and circles on page 286.
Freehand	Draw freehand shapes as you would with a pen on paper. See Create freehand shapes on page 287.
Line	Draw straight diagonal, horizontal, and vertical lines. See Create lines on page 287.
Polygon	Draw a series of connected straight lines forming a closed shape. See Create polygons and polylines on page 287.
Polyline	Draw a series of connected straight lines. See Create polygons and polylines on page 287.
Rectangle	Draw rectangles and squares. See Create rectangles and squares on page 288.
Rounded rectangle	Draw rectangles and squares with rounded corners. See Create rounded rectangles and squares on page 289.
Wedge	Draw a filled segment of an ellipse or circle. See Create arcs and wedges on page 285.
Symbol Factory	Select images to place in your display.

Control the application

Use this graphic object	To do this
Goto display button	Open a graphic display. For details, see Goto display buttons on page 173.
Return to display button	Close a display and return to the previous display. For details, see Return to display buttons on page 174.
Close display button	Close a display. Can also send a value to a tag. For details, see Close display buttons on page 174.
Display list selector	Select a display to open from a list of displays. For details, see Display list selectors on page 175.
Display print button	Print the current display. For details, see Print displays at runtime on page 273.

Use this graphic object	To do this
Language Switch button	Switch the application language. For details, see About language switching on page 159.
Login button	Open the **Login** dialog box and then log in. For details, see Provide a way for users to log in and log out on page 148.
Logout button	Log out of the application. For details, see Log out on page 149.
Password button	Change the current user's password, or any user password. For details, see Change passwords on page 212.
Add User/Group button	Add a FactoryTalk security user, a Windows-linked user, or a Windows-linked group to the current running application. For details, see Add a user or group on page 216.
Delete User/Group button	Delete a FactoryTalk security user, a Windows-linked user, or a Windows-linked group from the current running application. For details, see Delete a user or group on page 219.
Modify Group Membership button	Add a FactoryTalk user, a Windows-linked user, or a Windows-linked group to the current running application, or remove from the application. For details, see Modify a user or group membership on page 220.
Unlock User button	Unlock a FactoryTalk user account, which has been locked from the current running application. For details, see Unlock a user on page 222.
Enable User button	Enable a disabled FactoryTalk user account from the current running application. For details, see Enable a user on page 224.
Disable User button	Disable a FactoryTalk user account from the current running application. For details, see Disable a user on page 223.
Change User Properties button	Modify a FactoryTalk user's properties in the current running application. For details, see Change User Properties on page 224.
Shutdown button	Stop the application and shut down FactoryTalk View ME Station. For details, see Shutdown buttons on page 175.
Goto configure mode button	Stop the application and open the **FactoryTalk View ME Station** dialog box. For details, see Goto configure mode buttons on page 173.
Print alarm history button	Print a report of alarm messages in the alarm log file. You can print a report for all alarms, or for a specified alarm trigger.
Print alarm status button	Print a report of the status of alarms, including how many times each alarm was triggered and the time in alarm. You can print a report for all alarms, or for a specified alarm trigger.

Start and control processes

Use this graphic object	To do this
Momentary push button	Start a process or action by sending one value to the tag when pressed, and another value when released.

Use this graphic object	To do this
Maintained push button	Toggle between two values by sending one value to the tag when pressed, and a second value the next time the button is pressed and released. This button is useful for changing a setting within a machine or process, but not for starting the machine or process.
Latched push button	Start a machine or process. The button remains set (latched) until the process is complete. For example, use this button to start a bag filling machine. When the process is complete (the bag is full), the button is reset (unlatched) by the **Handshake** connection.
Multistate push button	Cycle through a series of values. Each time the operator presses the button, the value for the next state is sent to the tag. When the button is in its last state, pressing it changes the button to its first state and writes out the first state value. This button is useful when you want the operator to see and select multiple options in sequence, using a single button. The button shows the current state of an operation by showing a different color, caption, or image to reflect the different states.
Interlocked push button	Use a group of buttons to send values to the same tag. When the operator presses one button in the group, the button's value is sent to the tag, and the button remains highlighted as long as the tag value is the same as the button's value. Pressing another button in the group releases the first button, and sends a new value to the tag. You can also use a single Interlocked push button to send a value to a tag.
Ramp button	Increase or decrease the value of a tag by a specified integer or floating-point value. For example, use two Ramp buttons together to create a raise/lower control.
Control list selector	Select from a list of states for a process or operation. The list is highlighted to show the current state, and the operator can scroll through the list to select a different state. The value assigned to the selected state is written to the tag.
Piloted control list selector	Select from a list of states for a process or operation. The list is highlighted to show the current state, and the operator or a remote device such as a programmable controller can scroll through the list to select a different state.
Drawing object with horizontal or vertical slider animation	Set the value of a tag by dragging the slider object. The pixel position of the slider is translated into a value that is written to the tag. If the value of the tag is changed at the data source, the position of the slider changes to reflect this. For information about animation, see Animate graphic objects on page 338.
ActiveX object	Change tag values using a third-party object connected to an analog, digital, or string tag, including both HMI and data server tags. When the object's property value changes, the new value is written to the associated tag. For details, see ActiveX objects on page 290.

Show processes and values graphically

Use this graphic object	To show this
Bar graph	Numeric values in bar graph format. The bar graph increases or decreases in size to show the changing value.
Gauge	Numeric values in dial format. The gauge's needle moves around the dial to show the changing value.

Use this graphic object	To show this
Scale	A static indication of the range of values for a bar graph.
Multistate indicator	The state of a process, on a panel that changes its color, image, or caption to indicate the current state. Each state is set up to correspond to a numeric tag value.
Symbol	The state of a process, using a monochrome image that changes color to indicate the current state. Each state corresponds to a numeric tag value. This object is useful for showing the state of a process or operation at a glance.
List indicator	The state of a process, using a list of possible states with the current state highlighted. Each state is represented by a caption in the list, and corresponds to a numeric tag value. This indicator is useful if you want to view the current state, but also want to see the other possible states. For sequential processes, the list can alert the operator about what happens next in the process.
Trend	Historical or current numeric tag values, plotted against time or shown in an XY plot (where one or more tags' values are plotted against another tag's values to show the relationship between them). For details, see Set up trends on page 398.
Time and date display	Show the current time and date.
ActiveX object	Data using a third-party object connected to an analog, digital, or string tag, including both HMI and data server tags. The format of the data shown depends on the object. For details, see ActiveX objects on page 290.
Drawing object with rotation, width, height, fill, color, or horizontal or vertical position animation	The value of a tag using a pictorial representation of the current value in relation to a range of possible values. For example, use rotation animation to show the tag value as a needle's position on a dial. For color animation, assign different colors to represent different values. For information about animation, see Animate graphic objects on page 338.

Work with lists, trends, alarm banners, audit trail list, and numeric input objects

Use this button	With this graphic object	To do this
Pause button	Trend	Toggle a trend between pausing and automatic scrolling.
Next pen button	Trend	Change the vertical axis labels for a trend to the scale for the next pen.
Backspace button	Control list selector Piloted control list selector	Move the cursor back to the highlighted item in the list.
End button	Lists and trends	**List** – move to the bottom item in the list. **Trend** – resume trend scrolling and move to the current (latest) data in the trend.
Enter button	Lists Alarm list and alarm banner	Select the item the cursor is pointing to. Acknowledge the currently selected alarm.
Home button	Lists and trends	**List** – move to the top item in the list. **Trend** – pause the trend and move to the earliest data in the trend.
Move left / right buttons	Trend, Diagnostics list	**Trend** – pause the trend and scroll to the left or right.

Use this button	With this graphic object	To do this
		Diagnostics list — scroll the diagnostic message content across the display to view the full content.
Move up / down buttons	Lists, trends, and numeric input objects	**List** — move up or down one item in the list. **Trend** — scroll up or down to show higher or lower values on the vertical scale. **Numeric input cursor point** and **Numeric Input Enable** button — ramp the value up or down.
Page up / down buttons	Lists	Move up or down one page in the list.
Acknowledge alarm button	Alarm list Alarm banner	Acknowledge and silence the selected alarm.
Alarm status mode button	Alarm status list	Change the type of alarms shown in the alarm status list, from all alarms to active alarms to past alarms.
Clear alarm banner button	Alarm banner	Clear the alarm in the alarm banner without removing the alarm from the alarm log file and alarm lists.
Diagnostics clear button	Diagnostics list	Remove the selected message from all diagnostics lists.
Information acknowledge button	Information message display	Acknowledge the current message on the display.

For more information about using the buttons with lists and trends:

For information about	See
Link a button to a specific list, alarm banner, or trend object	Link buttons to objects on page 332
Use buttons with alarm lists, alarm banners, and alarm status lists	Use buttons with the alarm history and alarm objects on page 122
Use buttons with information message displays	Buttons on the [INFORMATION] display on page 395
Use buttons with diagnostics lists	Use buttons with the diagnostics list on page 138
Use buttons with audit trail lists	Use buttons with the audit trail list on page 141
Use buttons with trends	Use buttons to control the trend at runtime on page 407

Unlike the buttons in the previous table, the following buttons do not work with specific graphic objects. You can use them to work directly with alarms in the alarm history, and with the application's audit and diagnostics messages.

Use this button	To do this
Acknowledge all alarms button	Acknowledge and silence all currently unacknowledged alarms, or the alarms for a specific alarm trigger.
Clear alarm history button	Remove alarms from the alarm log file and all alarm lists. You can remove all alarms, or just the alarms for a specific alarm trigger. You can also reset the cleared alarms. This resets the number of times an alarm has been triggered to 0, and the accumulated time in alarm to 0.
Reset alarm status button	Reset the number of times an alarm has been triggered to 0, and the accumulated time in alarm to 0, for all alarms.
Silence alarms button	Silence the audio indicator for all current alarms.

Use this button	To do this
Sort alarms button	Toggle between sorting alarms in alarm lists and the alarm log file by time and by trigger value.
Diagnostics clear all button	Remove all diagnostics messages from all diagnostic lists.
Clear audit trail button	Remove all audit information from the audit log file, audit trail list, and audit trail RAM cache.

Enter and show numeric and string values

Use this graphic object	To do this
Numeric input enable button	Enter a numeric value and then write the value to a tag or an expression, or ramp a value at the data source.
Numeric input cursor point	Enter a numeric value and then write the value to a tag or an expression, or ramp a value at the data source.
String Input Enable button	Enter a string value and then write the value to a tag.
Numeric display	Display numeric tag values. For example, show the current temperature of an oven.
String display	Display string tag values. For example, set up the data source to generate strings that report on the state of a process of operation, or that provide the operator with instructions about what to do next.
RecipePlus button	Display data set and tag values for ingredients in the **RecipePlus** table. Write tag values from the selected data set to the data source. Write tag values from the data source to the selected data set or to a new data set. Save data set values from the table to a recipe file. Rename or delete recipe units. For details, see RecipePlus button on page 412.
RecipePlus selector	Select the recipe unit to show, download from, write to, rename, or delete. For details, see RecipePlus selector on page 412.
RecipePlus table	Display and compare recipe data set values and tag values. Edit data set values For details, see RecipePlus table on page 412.
ActiveX object	Enter or show data using a third-party object connected to an analog, digital, or string tag, including both HMI and data server tags. The format of the data entered or shown depends on the object. For details, see ActiveX objects on page 290.

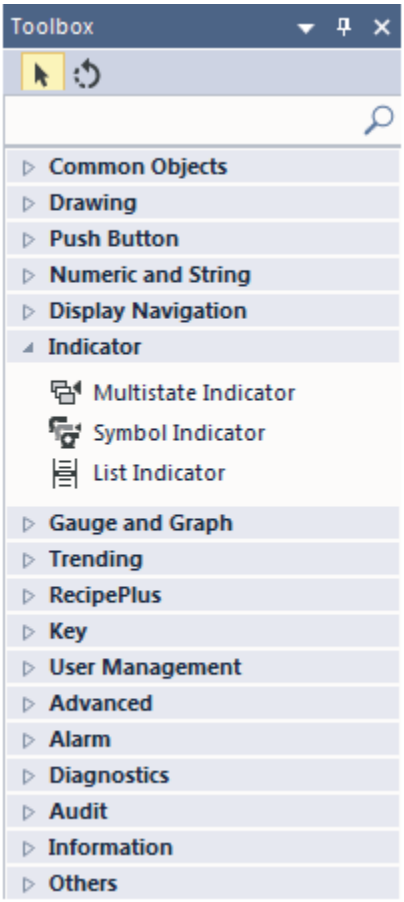
Show alarms, audits, and messages

Use this graphic object	To show this
Alarm list	Multiple alarm messages, including the time the alarms are triggered and acknowledged. For details, see How the alarm list graphic object works on page 119.
Alarm banner	A single unacknowledged alarm message. For details, see How the alarm banner graphic object works on page 120.
Alarm status list	The status of alarms, including how many times an alarm has been triggered and how long it has been active. For details, see How the alarm status list graphic object works on page 121.

Use this graphic object	To show this
Diagnostics list	Messages about system activity such as tag reads, tag writes, and communications errors. For details, see How the diagnostics list graphic object works on page 138.
Audit trail list	Information about operator actions at runtime. For details, see How the audit graphic objects work on page 140.
Audit trail detail	Information about the selected audit entry on an audit trail list. For details, see How the audit graphic objects work on page 140.
Information message display	Messages about the process, prompts or instructions, and information about current states. For details, see The [INFORMATION] display on page 395.
Local message display	Ongoing information about the status of devices or processes. For details, see How the local message display graphic object works on page 272.

Select tools for creating graphic objects

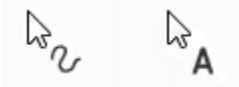
The **Objects** menu contains the items for creating objects, and selecting and rotating objects. You can also use the toolbox to create objects.



To select a tool

- Select the tool from the **Objects** menu, or drag the tool from the toolbox.

The following are selected freehand tool and text tool.



To deselect a tool, use one of these methods:

- Double-click an empty area on the display.
- Click the **Select** tool .
- Select another tool.

For some drawing objects, double-clicking an empty area of the display creates another instance of the object. For these objects, to finish drawing, click the Select tool. For more information, see Drawing objects on page 282.

Use the grid

To size and position objects precisely as you create them, use the grid.

To use the grid

- From the **View** menu, select **Show Grid > Snap On**.
  For information about setting up the grid, see Use the grid on page 256.

For information about setting up graphic objects once you have created them, see Set up graphic objects on page 319.

For information about attaching animation to the objects you create, see Animate graphic objects on page 338.

Create graphic objects

This section applies to graphic objects in general.

ActiveX objects and some drawing objects require extra steps to create them. For details about creating drawing objects, see the next section. For information about ActiveX objects, see ActiveX objects on page 290.

To create a graphic object

1. Select the tool for the object to create.
2. Click the mouse where you want to position the object, and then drag to draw a rectangle the general size you want the object to be.
3. Double-click the object to open its **Properties** dialog box.
4. In the dialog box, specify how the object looks, its behavior, and connections. For more information about the **Properties** dialog box, see Use the Properties dialog box on page 295.
   You can also use the **Property Panel** to set up objects. For information about using the **Property Panel**, see Use the Property Panel on page 296.

Drawing objects

Drawing objects are static objects used to illustrate your graphic displays. Create drawing objects on your graphic displays to help the operator understand how to use the display. For example, draw a rectangle around a group of

buttons to separate the group from other elements in the display, or place text on the display to give the operator instructions.

All the objects on the **Object > Drawing** menu are drawing objects.

The following instructions for creating drawing objects describe how to create the objects and then open the objects' **Properties** dialog boxes to set up how the objects look. For information about using the **Properties** dialog box, see Use the Properties dialog box on page 295.

You can also use the **Property Panel** to set up objects. For information about using the **Property Panel**, see Use the Property Panel on page 296.

Create text

Use the text tool to draw a text object to use for labels or instructions on the display.

To create a text object

1. In the **Graphics Display** editor, select **Objects > Drawing > Text** or drag A from the toolbox.
2. Press the mouse button, drag the mouse diagonally to draw the object to the desired size, and release the button.
   The **Text Properties** dialog box opens.
3. Enter the text and specify the options.

   Tip:
   - To move the cursor to the next line of text, press **Enter**.
   - To use an embedded variable in the text, click **Insert Variable**.
   - To create another text object, move to an empty spot in the drawing area, click and drag to create another text object.

4. Click **OK**.

After you have set up one text object to look the way you want, copy and paste it to create additional text objects with the same formatting. Double-click the text object to edit the text of the new objects.

For more information about text object, see Help.

Choose fonts

You can select any font you have installed, but **TrueType** and **OpenType** fonts are recommended. These fonts can be resized easily, without losing text quality. For PanelView Plus and PanelView Plus CE terminals, you must use **TrueType** fonts.

If you run an application on a computer that does not have the fonts you used when setting up the application, Windows substitutes with the fonts that most closely match the fonts you specified.

If you are going to use the application with multiple languages, we recommend using Microsoft Sans Serif or Tahoma. These fonts allow for font linking to support the character sets of other languages. PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, and PanelView Plus CE terminals are shipped with font linking turned on.

For more information about font linking, see Help. For more information about setting up languages for your application, see Set up language switching on page 159.

Create images

Use the image graphic object to place images on your graphic displays. You can use images already contained in the **Images** folder or use Symbol Factory to select and add a graphic to the display and the **Images** folder.

Prerequisite

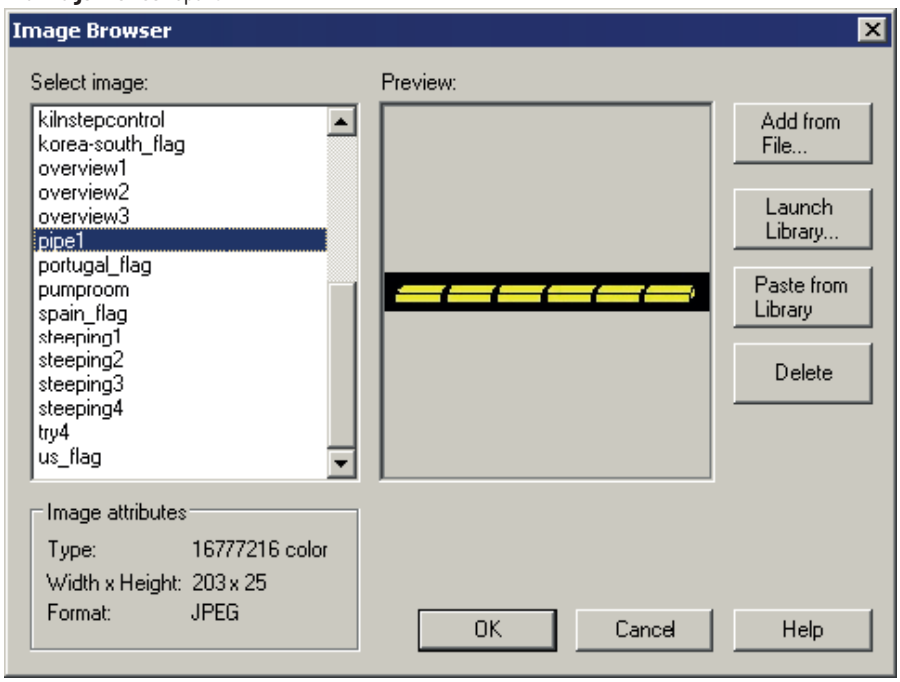
- Import the image into your project.

For more information, see Import bitmap, JPEG, PNG, and SVG images on page 267.

To create an image object

1. In the **Graphics Display** editor, select **Objects > Drawing > Image** or drag from the toolbox.
2. Press the mouse button, drag the mouse diagonally to draw the object to the desired size, and release the button.

   The **Image Browser** opens.



3. From the **Select image** list, select the image to be shown on the object.
4. Click **OK**.

The image is placed where you drew the rectangle, but the actual size of the image is used, rather than the size of the rectangle you drew. To change the image's attributes, double-click the image to open the **Image Properties** dialog box and specify the options.

For more information about using the **Image Browser**, see Use the Image Browser to import images on page 267.

For more information about the image object, see Help.

Add an image from Symbol Factory to a graphic object

Use the **Image Browser** to add an image from Symbol Factory as needed while setting up a graphic object.

To add an image from Symbol Factory to a graphic object using the image browser

1. In the **Image Browser**, click **Launch Library.**
2. Browse through the **Categories** and select the graphic to be used.
3. (optional) Modify the graphic options below the graphic tab.
4. Click **Copy as**.
   The Symbol Factory minimizes, returning to the **Image Browser**.

   Tip: Copy as SVG is not supported.

5. Click **Paste from Library**.
6. Type a unique name for the image and click **OK**.
   The graphic will now be shown in the **Preview** window. The graphic has been added to the list in the **Image Browser**. This also adds the image to the **Images** folder in the **Explorer** window.
7. Click **OK**.

Create panels

Use the panel tool to draw rectangles and squares that have borders. You can set up panel objects to blink at runtime. The panel object only supports visibility animation.

To create a panel object

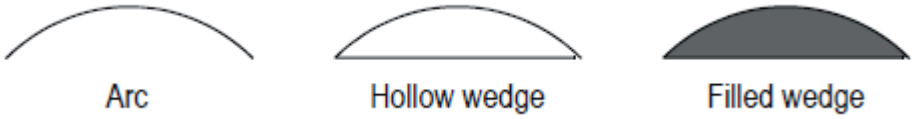
1. In the **Graphics Display** editor, select **Objects** > **Drawing > Panel** or drag from the toolbox.
2. Press the mouse button, drag the mouse diagonally to draw the object to the desired size, and release the button.

   Tip: Press **Ctrl** as you drag the mouse to draw a square panel.

3. Double-click the panel to open its **Properties** dialog box.
4. In the **Properties** dialog box, specify how you want the panel to look.
5. Click **OK**.

Create arcs and wedges

Arcs and wedges are drawn in two steps: first you create an ellipse or circle, and then you reshape it into the segment you want.

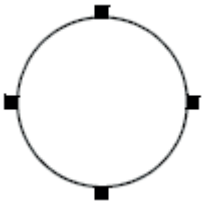


To create an arc or wedge

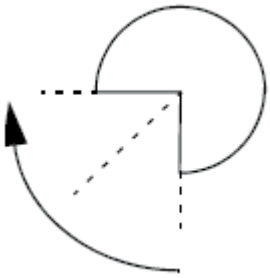
1. In the **Graphics Display** editor, select **Objects > Drawing > Arc** (or **Wedge**), or drag (or) from the toolbox.
2. Click the mouse where you want to position the object, and then drag to draw an ellipse or circle.

To base the arc or wedge on a circle rather than an ellipse, hold down **Ctrl** while you drag.

When you release the mouse button, a set of handles appears so you can decrease the angle of the wedge or arc from 360 degrees to the desired angle.



3. Click a handle, and drag the mouse to 'cut out' part of the circle.



4. To finish drawing, click the object.
5. To change how the object looks, click **Properties** on the context menu, or double-click the object to open its **Properties** dialog box.
6. Select arc or wedge options.

   For information about the options in the dialog box, see Help.
7. To close the dialog box, click **OK**.

You can also use the Arc and Wedge tools to reshape arcs and wedges. For more information, see Reshape drawing objects on page 310.

Create ellipses and circles

Use the ellipse tool to draw ellipses and circles.

To create an ellipse or a circle object

1. In the **Graphics Display** editor, select **Objects > Drawing > Ellipse** or drag from the toolbox.
2. Press the mouse button, drag the mouse diagonally to draw the object to the desired size, and release the button.

   Press **Ctrl** as you drag the mouse to draw a circle
3. Double-click the object to open its **Properties** dialog box.
4. In the **Properties** dialog box, specify how you want the object to look.
5. Click **OK**.

Tip: Because of a limitation in the Windows CE operating system, ellipses or circles that look fine on the desktop may appear ragged when shown on the PanelView Plus family terminals. To avoid this, do not use ellipses with a line width greater than 1 pixel.

Create freehand shapes

Use the freehand tool to draw shapes as you would with a pen on paper.

To create a freehand object

1. In the **Graphics Display** editor, select **Objects** > **Drawing > Freehand** or drag from the toolbox.
2. Press the mouse button, drag the mouse diagonally to draw the object to the desired size, and release the button.
3. Double-click the object to open its **Properties** dialog box.
4. In the **Properties** dialog box, specify how you want the object to look.
5. Click **OK**.

Create lines

Use the line tool to draw straight diagonal, horizontal, and vertical lines.

To create a line object

1. In the **Graphics Display** editor, select **Objects > Drawing > Line** or drag from the toolbox.
2. Press the mouse button and drag the mouse to draw a line.

> **Tip:** Press **Ctrl** as you drag the mouse to draw a horizontal or vertical line.

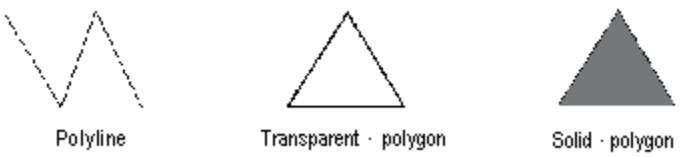
3. Double-click the line to open its **Properties** dialog box.
4. In the **Properties** dialog box, specify how you want the line to look.
5. Click **OK**.

> **Tip:** When you put a line into **Edit** mode and add a point to it, it is converted into a polyline object. If you remove a point from a polyline with three points, it becomes a line object.

You can use the **Polyline** tool to convert the line into a polyline. For more information, see Reshape drawing objects on page 310.

Create polygons and polylines

A polyline is a series of connected line segments. A polygon is a multi-sided object (with three or more sides). For example, use the polygon shape if you want to create triangles.



To create a polygon or polyline

1. In the **Graphics Display** editor, select **Objects > Drawing > Polygon** (or **Polyline**), or drag (or) from the toolbox.
2. Click and drag to create the first segment of the object. Release the mouse button.
   To draw horizontal or vertical lines, hold down **Ctrl** while you drag.

3. Move the mouse to where you want the next segment to end, and then click.
   Repeat this step until you have completed the object.

4. To finish drawing, double-click an empty area of the display, or click the **Select** tool.
5. To change how the object looks, double-click it to open the object's **Properties** dialog box.
6. Select polygon or polyline options.
   For information about the options in the dialog box, see Help.
7. To close the dialog box, click **OK**.

You can use the **Polygon** tool to reshape lines, polygons, polylines, and rectangles. For more information, see Reshape drawing objects on page 310.

Create rectangles and squares

Use the rectangle tool to draw rectangles and squares.

Tip: Squares and rectangles are specialized polygons, so the Properties dialog boxes are the same as for the polygon object.

To create a rectangle or square object

1. In the **Graphics Display** editor, select **Object > Drawing > Rectangle** or drag from the toolbox.
2. Press the mouse button, drag the mouse diagonally to draw the object to the desired size, and release the button.

   Tip: Press **Ctrl** as you drag the mouse to draw a square.

3. Double-click the object to open its **Properties** dialog box.
4. In the **Properties** dialog box, specify how you want the rectangle to look.
5. Click **OK**.

You can use the Polygon tool to reshape the rectangle into a polygon. For more information, see Reshape drawing objects on page 310.

Create rounded rectangles and squares

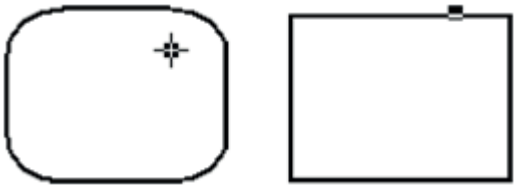
Use the rounded rectangle tool to draw rectangles and squares with rounded corners.

To create a rounded rectangle or square object

1. In the **Graphics Display** editor, select **Objects > Drawing > Rounded Rectangle** or drag ▢ from the toolbox.
2. Press the mouse button, drag the mouse diagonally to draw the object to the desired size, and release the button.

> **Tip:** Press **Ctrl** as you drag the mouse to draw a rounded square.

3. To change the degree of rounding, click and drag the single handle that appears inside the object, until the object is the desired shape.



4. Double-click the rectangle to open its **Properties** dialog box.
5. In the **Properties** dialog box, specify how you want the rounded rectangle to look.
6. Click **OK**.

> **Tip:** To change the degree of roundness after the object has been created, right-click the object and select **Edit**. The single handle will appear in the object.
> Due to a Windows limitation, you cannot rotate rounded rectangles and rounded squares.

For information about using the **Rounded Rectangle** tool to reshape the rounded rectangle, see Reshape rounded rectangles on page 311.

Import a .dxf or .wmf graphic image

When importing .wmf and .dxf files, they are converted to drawing objects, such as lines, ellipses, and polygons.

> **Tip:** Depending on the complexity of the file, the converted image could consist of 500 or more drawing objects. This would lead to long display load time. In this case, it would be better to covert the .wmf or .dxf file to a bitmap, and then show the bitmap in an image object.

To import a .wmf or .dxf file on a display

1. From the **Objects** menu, select **Import**.
2. Click the mouse where you want to position the file, and then drag to draw a rectangle.
3. In the **Files of type** box, select the type of file to import.
4. Navigate to the directory where the file is stored, and then select the file to import.
5. Click **Open**.

The file is converted to drawing objects and grouped, and then the grouped object is placed on the graphic display.

ActiveX control

An ActiveX control is an off-the-shelf software component created by a third-party vendor. An ActiveX control provides functions which can be accessed by FactoryTalk View ME Station through the object's properties. ActiveX objects use tags or expressions to exchange information with the data source. The properties and connections available for a particular ActiveX object depend on the third-party vendor's implementation.

IMPORTANT: The runtime behavior of an ActiveX control depends on the vendor's implementation. Test the object thoroughly before you run an application to ensure the object behaves as expected.

The information passed between FactoryTalk View ME Station and an ActiveX control must be in one of the formats used by a tag:

- Analog number
- Digital number
- String

ActiveX controls must be available for the supported operating systems and for the PanelView Plus family platforms. To use an ActiveX control at runtime, you must have the version of the object that is compatible with the operating system of the runtime computer.

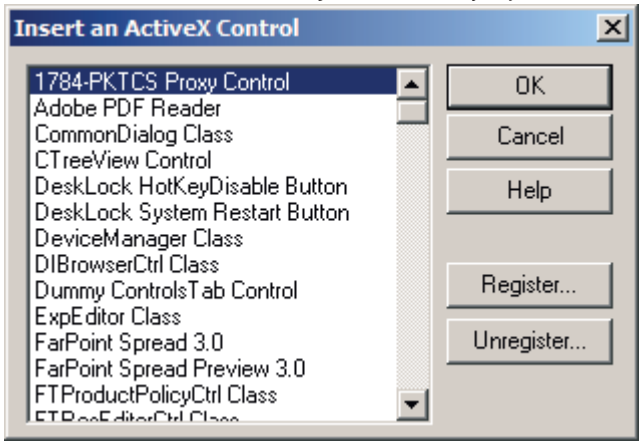
ActiveX objects support visibility animation only.

Add an ActiveX control to a graphic display

You may need to add an ActiveX control to a graphic display.

To add an ActiveX control to a graphic display

1. In FactoryTalk View Studio, select **Objects > ActiveX Control** or drag  from the toolbox.
2. On the graphic display, drag the ActiveX tool to draw a rectangle the size you want the object to be.
3. In the **Insert an ActiveX Control** dialog box, select the object you want to add and click **OK**.



For information about the options in the dialog box, see Help.

The object is placed on the display. Depending on how the third party implemented the object, it might be a different size than the rectangle you drew.

4. To specify the object's properties and assign tags or expressions to its connections, do one of the following:
   - Right-click the object, and then select **Properties** to open the object's **Properties** dialog box.
     Depending on how the third party implemented the object, it might not have a **Properties** dialog box. If the **Properties** menu item is not available, use the next method.
   - Right-click the object and select **Property Panel**.
5. In the **Properties** dialog box or **Properties** tab of the **Property Panel**, specify the object's properties.
6. (Optional) Set up the properties on the **Common** tab.
   For more information, see Set up objects' spatial properties, names, and visibility on page 327.
7. On the **Connections** tab, assign tags or expressions to the object's connections.
8. Click **OK** to close the **Properties** dialog box, or click **Close** to close the **Property Panel**.

For information about using the **Properties** dialog box, see Use the Properties dialog box on page 295.

For information about using the **Property Panel**, see Use the Property Panel on page 296.

Tools and tips for working with objects

This section describes features of the **Graphics** editor that help you work with the objects you create. It describes how to:

- Select and deselect objects.
- Use the **Object Explorer** to view and select objects.
- Use an object's **Properties** dialog box to set up the object's properties and assign tags and expressions to its connections.
- Use the **Property Panel** to set up individual and group object properties, and to assign tags and expressions to individual objects' connections.
- Color objects.
- Name objects.
- Test how objects look in different states.

Select and deselect objects

To work with an object, you must first select it. You can use the **Select** tool or the **Object Explorer** to select objects.

For information about using the **Object Explorer**, see Use the Object Explorer on page 292.

To select the Select tool

- Select **Objects > Select** or click  on the **Objects** toolbar.

The mouse pointer changes to a single arrow.

To select	Do this
An object or group of objects	Click the object or group. In the **Object Explorer**, click the object or group.
An object within a group of objects	Double-click the group, and then click the object. In the **Object Explorer**, open the group, and then click the object.
Several objects	Click the first object, and then Ctrl-click additional objects.

To select	Do this
All objects in an area	Click and drag diagonally to draw a selection border around the objects. Ctrl-click objects outside the border to add them to the selection.
All objects in the drawing area or in a group you are editing	From the **Edit** menu, select **Select All**, or press **Ctrl+A**.

To deselect	Do this
An object	Ctrl-click the object.
Several objects	Press **Ctrl** and drag a selection border around the objects.
All selected objects	Click in the **Drawing** area, away from any objects.

Use the Object Explorer

Object Explorer provides a tree-list naming all the objects on the graphic display and allows you to select, hide, and highlight objects from the list. Groups are listed as expandable items in the tree, with a + icon.

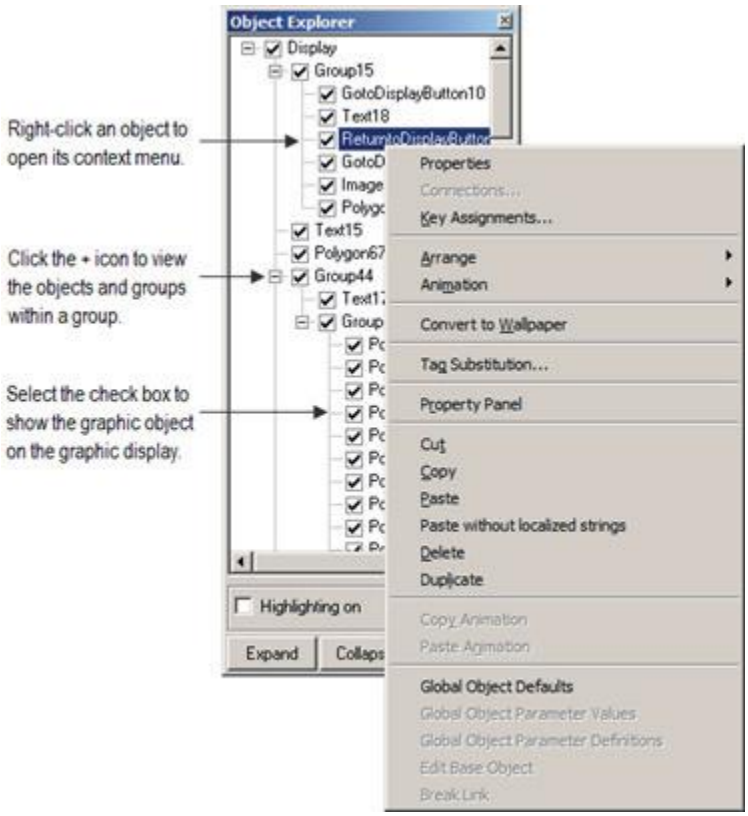
You can use the **Object Explorer** to:

- Select an object that is hidden behind other objects on the graphic display, without bringing the object to the front. Objects are listed in front-to-back order. The object you created most recently is at the front, unless you move it back using the **Send to Back** option.
- Show or hide an object on the graphic display by selecting or clearing the checkbox in front of the object in the **Object Explorer**.
- Highlight objects by object type, highlight objects that have animation attached, and to highlight objects with specific tag or expression assignments.

The **Object Explorer** does not show wallpaper objects or objects within ActiveX composite objects.

When you open a project, the **Object Explorer** is shown on the right in the main window. You can hide it, and change its size and position. If it is not shown, open it using one of the following methods:

- On the **Graphics** toolbar, click  .
- From the **View** menu, select **Object Explorer**.
- Right-click an object and select **Object Explorer**.



For more information about using the **Object Explorer**, see Help.

For more information about layering objects, see Layer objects on page 314.

For more information, see Show or hide an object on the graphic display on page 293.

For more information, see Highlight objects in the Object Explorer on page 294.

Show or hide an object on the graphic display

You can use the checkbox in front of a graphic object in the **Object Explorer** to show or hide the object on the graphic display.

To show or hide an object on the graphic display

- In the **Object Explorer**, select the checkbox in front of the object you want to show, or clear the checkbox in front of the object you want to hide.

Tip:

- By default, all checkboxes in the **Object Explorer** are selected and all objects are shown on the graphic display.
- If you select or clear the checkbox in front of a group in the **Object Explorer**, all objects in the group are shown or hidden on the graphic display. If you select or clear the checkbox in front of the root node, **Display** in the **Object Explorer**, all the objects on the graphic display are shown or hidden.
- At runtime, the visibility state of a graphic object depends on the expression. The **Object Explorer** visibility state of a graphic object has no effect on the default visibility state of the object at runtime.

Highlight objects in the Object Explorer

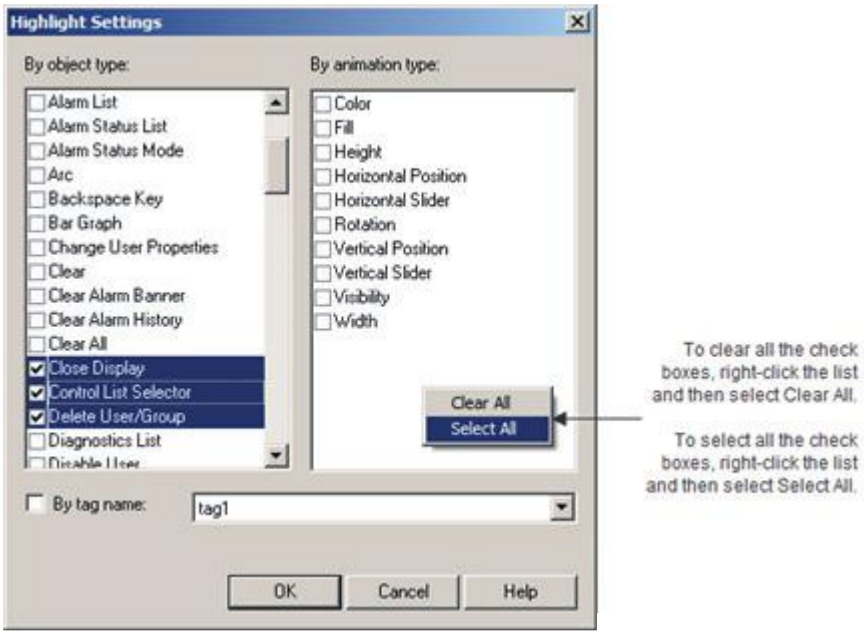
You can use the **Object Explorer** to highlight:

- Specific types of objects
- Objects that have animation attached
- Objects that have a specific tag or expression assigned to them

The objects are highlighted in red in the **Object Explorer** and in the graphic display. If your graphic display uses a red background, the highlighting is not visible in the graphic display.

To highlight objects in the Object Explorer

1. In the **Object Explorer**, select the **Highlighting on** checkbox, and then click **Settings**.
2. In the **Highlight Settings** dialog box, select the object type(s), animation type(s), and tag name (if any) to be highlighted.



For details about the options in the **Highlight Settings** dialog box, see Help.

Use the Properties dialog box

Every graphic object has a **Properties** dialog box that you can use to set up the object. Depending on how the vendor implemented the object, third-party ActiveX objects might have a **Properties** dialog box as well.

The **Properties** dialog box contains tabs that you can use to set up the object's properties and connections:

On this tab	Do this
General	Set up the object's appearance, audio indicator, and touch margins (for buttons), and settings that are unique to the object, such as the button action for a push button, whether to use key navigation to select the object, or whether to link a button to a specific object. For information about touch margins, see Use touch margins on page 329. For information about key navigation, see Remove objects from and adding objects to the tab sequence on page 332. For information about linking buttons to objects, see Link buttons to objects on page 332.
States	Set up the states for the object, including the value for each state and whether to show a caption or image for the state. For information about checking that the states are set up the way you intended, see Test how objects look in different states on page 300.
Label	For objects that don't have multiple states, specify whether to use a caption or image on the object. For information about using the **Image Browser** to select an image to use in the label, see Use the Image Browser to import images on page 267.
Timing	Set up the object's auto repeat (see Repeat a button's action by holding down the button on page 334), or **Enter key handshaking** (see Ensure that values are read by the data source before sending new values on page 335) settings.
Common	Set up the object's spatial properties, name, and visibility. For details, see Set up objects' spatial properties, names, and visibility on page 327.
Connections	Assign tags and expressions to the object's connections. For information, see Assign tags and expressions to the object's connections. on page 301

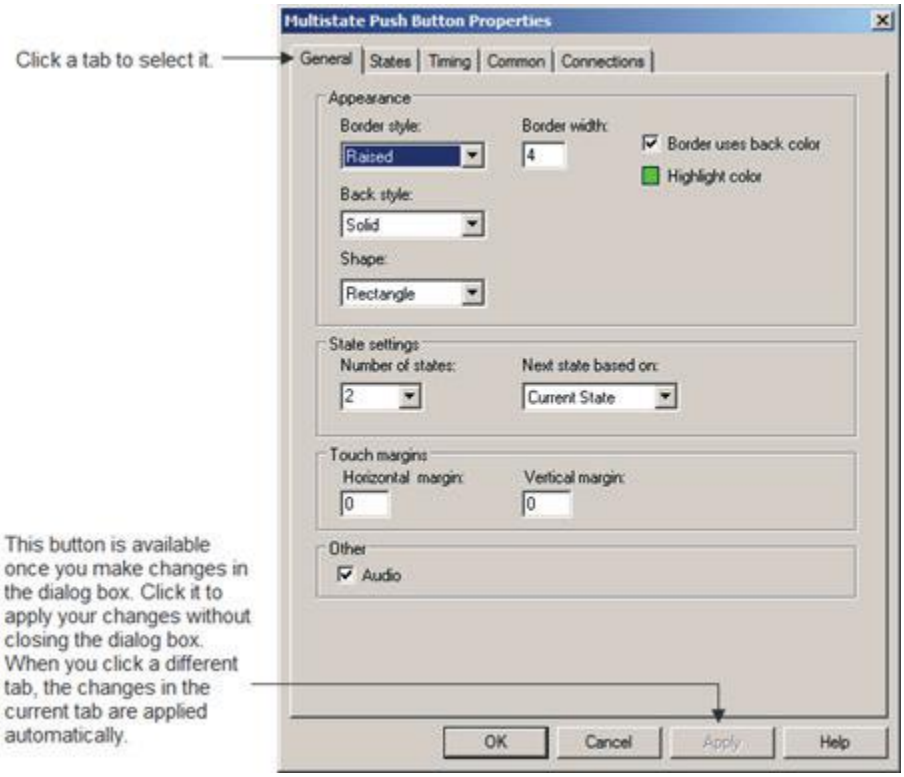
The tabs that are available depend on the object:

- Some objects have only **General** and **Common** tabs.
- If an object can have more than one state, the object's **Properties** dialog box contains a **States** tab.
- Some objects have unique tabs that are not listed in the table above.
- The tabs that an ActiveX object has depends on the vendor's implementation, though if you can assign tags or expressions to the object it has a **Connections** tab.

To open an object's Properties dialog box, use one of these methods

Tip: To automatically open the object's property dialog box when you place the object on a display, use the **Settings** tab (**Tools** > **Options** >**Settings**).

- Double-click the object.
- Right-click the object and select **Properties**.
- Select the object, and then from the **Edit** menu, select **Properties**.
- In the **Property Panel**, with the object selected, click the **(Custom)** property and then click the **Browse** button ...
- In the **Property Panel**, with the object selected, double-click the **(Custom)** property.
- In the **Object Explorer**, double-click the object.



You can also use the **Property Panel** to set up an object's properties, as described next.

For details about setting up a particular object, see Help.

Use the Property Panel

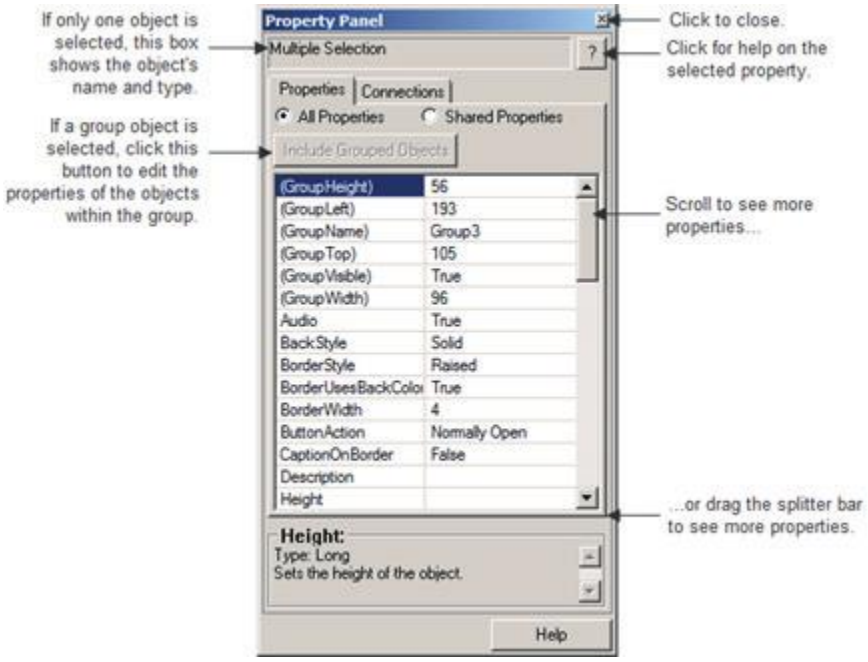
Use the **Property Panel** to modify the properties of graphic objects and assign tags and expressions to the objects. The **Property Panel** is especially useful for making changes to the properties of multiple objects at the same time.

When you open a project, the **Property Panel** is shown on the right in the main window. You can hide it, and change its size and position. If it is not shown, open it using one of the following methods:

- On the **Graphics** toolbar, click
- From the **View** menu, select **Property Panel**
- Right-click an object and select **Property Panel**
- Right-click an empty area of a display and select **Property Panel**

Set up properties

Use the **Properties** tab of **Property Panel** to set up the properties of the selected object or objects.



You can keep the **Property Panel** open as you work in the **Graphics** editor. You can drag the panel's borders to make the **Property Panel** larger or smaller.

For more information about using the **Property Panel** to set up an object's properties, see Help.

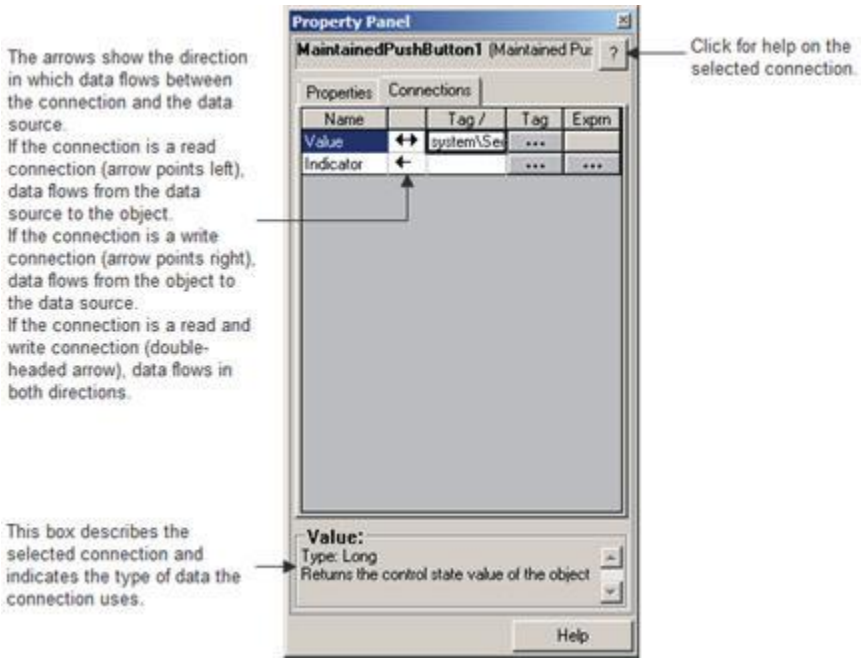
Assign tags and expressions to an object's connections

Use the **Connections** tab of **Property Panel** to assign tags or expressions to the selected object's connections. If multiple objects are selected the tab is blank, because you can assign tags or expressions to only one object at a time.

How values are updated

The arrows indicate the direction in which the data flows between the connection and the data source:

- A right arrow indicates that the connection sends values to the data source. The connection is a write connection.
- A left arrow indicates that the data source sends values to the connection. The connection is a read connection.
- A double arrow indicates that the data flows in both directions. The connection is a read/write connection.



For more information about using the **Property Panel** to assign tags and expressions to an object's connections, see Help.

For more information about assigning tags and expressions to objects, see Assign tags and expressions to graphic objects on page 301.

Color objects using the color toolbars

The **Foreground Color** and **Background Color** toolbars contain a selection of colors that you can assign to objects' color properties.

To show a color toolbar

- From the **View** menu, select **Toolbars**, and select **Foreground Colors** or **Background Colors**.

To close a color toolbar

- From the **View** menu, select **Toolbars**, and select **Foreground Colors** or **Background Colors**, or click the toolbar's **Close** button.

About color properties

The number of color properties an object has depends on the type of object and how you set it up. For example, a button with states can use up to seven different colors for each state. When you select colors using the color toolbars, some properties are assigned the foreground color and some are assigned the background color. Other color properties, such as Fill color, cannot be assigned using the color toolbars (instead, use the object's **Properties** dialog box or the **Property Panel**).

This table lists the color properties that you can assign using the color toolbars:

Property	Foreground color	Background color
Back color	No	Yes
Background color	No	Yes
Border color	Yes	No
Caption color	Yes	No
Caption back color	No	Yes
Fore color	Yes	No
Foreground color	Yes	No
Image color	Yes	No
Image back color	No	Yes
Legend color	Yes	No
Needle color	Yes	No
Pattern color	Yes	No

For objects with states, the selected color is applied to the current state's color properties only. In the **Property Panel**, properties that apply to states have *St_* at the beginning of the property name.

When to select colors using the toolbars

For all the drawing objects except image and panel, you can select colors from the color toolbars before you draw an object (either before or after you click the object's tool).

The other objects are always drawn using their default colors, but you can select the objects and then click the toolbars to change their colors. The toolbars are especially useful for quickly assigning the same colors to multiple objects.

Other methods for assigning colors

You can also assign colors using an object's **Properties** dialog box or the **Property Panel**. Use one of these methods if you want to assign separate colors to different foreground or background color properties, or to choose colors that don't appear on the toolbars.

For example, if you want to use a dark blue background color for a button, with a light blue background color for its image label, you must assign the colors separately. Using the toolbar would assign the same color to both properties.

Also use the **Properties** dialog box or **Property Panel** to change the default colors for properties that cannot be assigned using the color toolbars.

Name objects

Objects (and groups of objects) are automatically given a name and number when you create them, for example, NumericInputEnable4. If desired, you can assign a more meaningful name to the object, for example, Conveyor_speed_input. Each object on a display must have a unique name.

The object name appears in the status bar, **Diagnostics List**, **Property Panel**, and **Object Explorer** in FactoryTalk View Studio, and in diagnostics log messages at runtime.

Name an object in the Property Panel

Use the following steps to name an object in the Property Panel.

To name an object in the Property Panel

1. Select the object.
2. In the **Property Panel**, click the **Properties** tab.
3. Double-click the **(Name)** row, and then type the name.

   The name must start with a letter, and cannot contain spaces. You can use the underscore character (_).

Name an object in its Properties dialog box

Follow the steps below to name an object in its **Properties** dialog box.

To name an object in its Properties dialog box

1. Double-click the object to open its **Properties** dialog box.
2. Click the **Common** tab.
3. In the **Name** box, type the name.

   The name must start with a letter, and cannot contain spaces. You can use the underscore character (_).

Name a group object

Use the following steps to name a group object.

To name a group object

1. Select the group object.
2. In the **Property Panel**, click the **Properties** tab.
3. Double-click the **(GroupName)** row, and then type the name.

   The name must start with a letter, and cannot contain spaces. You can use the underscore character (_).

Test how objects look in different states

To ensure that the different states for an object are set up correctly, you can view them using the **States** toolbar or the **Property Panel**.

View an object's states using the States toolbar

Use the following steps to view an object's states using the **States** toolbar.

To view an object's states using the States toolbar

1. From the **View** menu, select **Toolbars > States**.
2. Select one or more objects.
3. On the **States** toolbar, click the state to view.

   If you selected multiple objects, the toolbar shows the states that are common to all the objects.
4. To view the next state, select it in the toolbar or press the **Down Arrow** key on your keyboard.

You can also use these keys to select the next state to view:

- To view the previous state, press the **Up Arrow** key.
- To view the first state, press the **Home** key.
- To see the last state, press the **End** key.

View an object's states using the Property Panel

Follow the steps below to view an object's states using the **Property Panel**.

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 select the state to view.
3. To view the next state quickly, double-click the row or press the **Enter** key on your keyboard.

Assign tags and expressions to graphic objects

You can assign tags and expressions to many graphic objects, including ActiveX objects (depending, of course, on how the vendor implemented the object). This section describes how to:

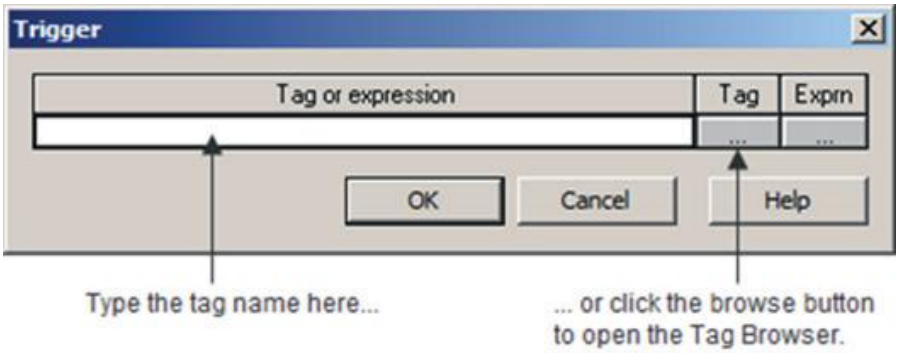
- Assign tags to graphic objects.
- Use expressions to manipulate tag values.
- Replace tags using tag substitution.
- Use tag placeholders so the same display can be used with different sets of tags.

Assign tags

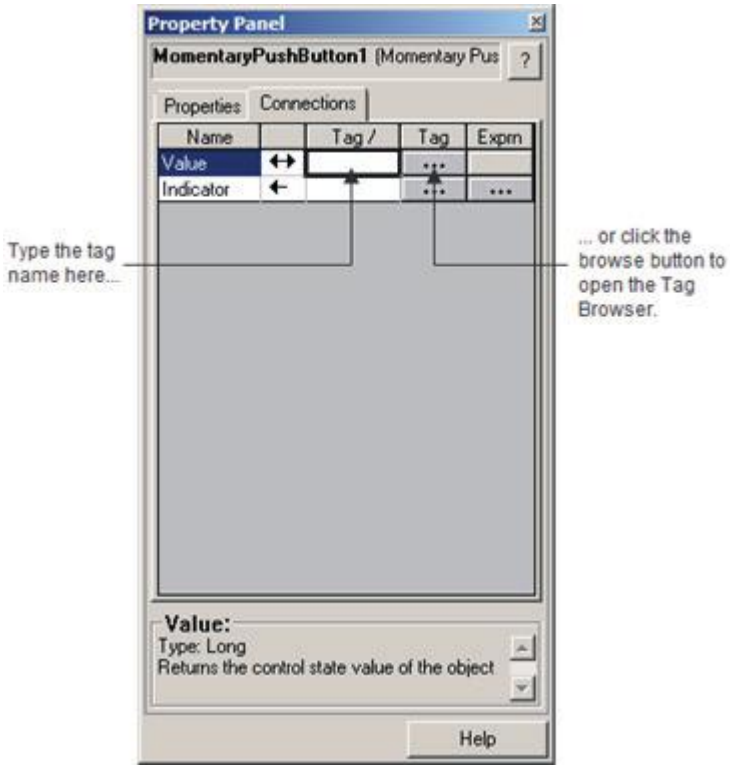
Follow the steps below to assign tags to a graphic object.

To assign tags to a graphic object, use one of these methods:

- Double-click the object to open the object's **Properties** dialog box, and then assign tags on the **Connections** tab.



- Select the object and then assign tags on the **Connections** tab of the **Property Panel**.



- Select the object, and then from the **Edit** menu, select **Connections**. Assign tags on the **Connections** tab of the **Properties** dialog box.
- Right-click the object, and then select **Connections**. Assign tags on the **Connections** tab of the **Properties** dialog box.

For information about:

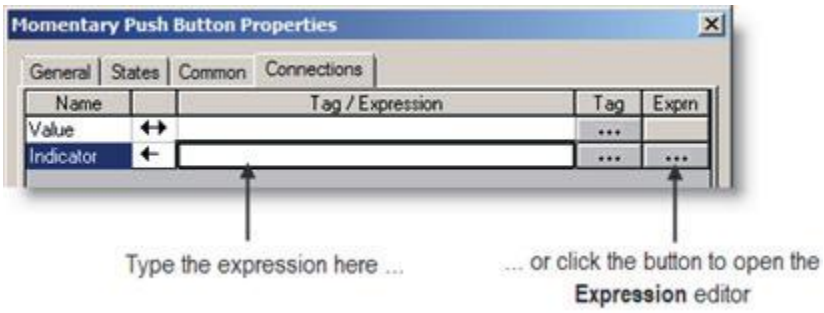
- Use the **Tag Browser**, see Use the Tag Browser on page 72.
- Use the **Properties** dialog box, see Use the Properties dialog box on page 295.
- Use the **Property Panel**, see Use the Property Panel on page 296.

Use expressions to manipulate tag values

Many of the connections to which you can assign a tag also permit the use of expressions to perform logical or mathematical calculations on tag values. If you assign an expression, FactoryTalk View monitors the expression value

rather than the original tag value. For example, your machine might send values to the data source in the form of temperature in degrees Celsius. You could use an expression to convert the value to degrees Fahrenheit, and then monitor the expression result rather than the original tag value.

If you can assign an expression, a Browse button is present in the **Exprn** column on the **Connections** tab.



To specify an expression, use one of these methods:

- In the **Tag / Expression** column, type the expression.
- In the **Exprn** column, click the **Browse** button and then create an expression in the **Expression** editor. Use this method if you want to check the expression syntax, or to use multiple lines for the expression.

For more information about expressions, see Use expressions on page 352.

Replace tags using tag substitution

You can replace tags assigned to the graphic objects in your display by using tag substitution. You can also replace the tags used in expressions assigned to graphic objects.

For example, if you assign a tag called *HoistHeight* to multiple objects on the display, and then decide to use the tag Hoist_height instead, you can use tag substitution to quickly replace the old tag with the new tag.

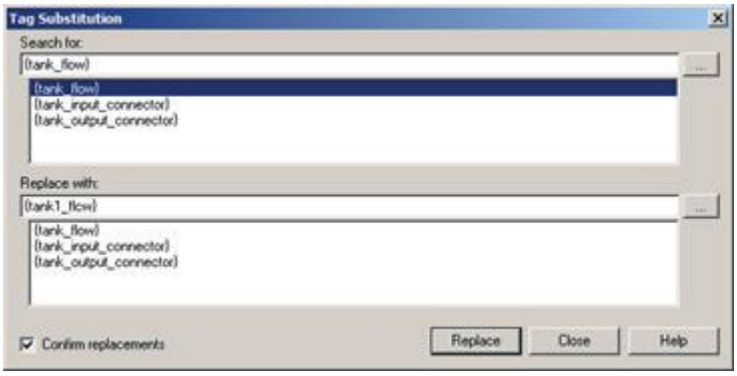
You can replace:

- A tag name (with or without folder names)
- A folder name
- The text in an expression
- Tags contained in embedded variables

To replace tags

1. Select one or more objects.
   To select all the objects on the display, from the **Edit** menu, select **Select All**.

2. From the **Edit** menu, select **Tag Substitution**.
3. To replace the tags for a single object, you can right-click it and then select **Tag Substitution**.



For details about the options in the **Tag Substitution** dialog box, see Help.

A log file is shown after substitution. It provides details on the substitution process and results. If the substitution fails, the reason will be shown. The last line shows where the log file is located in case you want to review it in the future.

Replace tags using Find and Replace

You can replace tags or components that refer to tags using Find and Replace.

To replace tags using Find and Replace

1. Open the **Find and Replace** dialog box by doing one of the following:
   - On the toolbar, click
   - From the **Tools** menu, select **Replace**
   - Press **Ctrl+H**.
2. Click the browse button ( ) next to **Find what,** which launches the **Tag Browser**.
   - Navigate to the correct folder in the left pane, and then select the tag in the right pane.
   - Click **OK** to save the selection and close the browser.
3. Click the browse button next to **Replace with.** Select the replacement tag from the **Tag Browser** and click **OK** to save the selection and close the browser.
4. Click on the browse button next to **Find within** and clear all the checkboxes except Graphic Objects. Click **OK** to save the selection and close the dialog box.
5. Identify the word search limitations in **Find Options**.
6. Select the **Direction**, in which to search:
   - Select **Up** to search for the tag or text string in all components before the shown one.
   - Select **Down** to search for the tag or text string in all components after the shown one.
7. To replace items, select **Confirm replacement** if needed, then:
   - To select individual items to replace, click **Find Next** and then click **Replace**, when the correct item is shown or
   - To replace all tags, Click **Replace All** to replace all found items.

If needed, click the **View Log File** button to show a text file of the results of the Replace, Replace All or Undo Replace functions.

To undo replacements, click **Undo Replace**. Only the last replace or replace all action can be undone. Undo will not be successful if a component has been manually edited after the replace action.

Tag placeholders

Tag placeholders provide a way to use one graphic display to represent a number of similar operations.

For example, suppose you are creating displays for a plant that cans corn and peas. The machinery used in both processes is identical. Instead of creating two displays and specifying corn-related tags in one display and pea-related tags in another, you can create one display and not specify any tag names. Where a tag name is required, type a tag placeholder.

You can use tag placeholders wherever you would normally assign a tag to an object, including in expressions and embedded variables. You can also use tag placeholders in the expressions you create to animate objects.

You can also use tag placeholders with global objects.

You can use tag placeholders in:

- The graphic display that opens when the application is first run.
- Graphic displays that are opened using a Goto display button.
- Graphic displays that are opened using a display list selector.
- The graphic display that opens when the logout button is pressed to log the logged-on user out.
- The graphic display specified on the **Project Settings Runtime** tab when auto logout is enabled.
- Graphic displays that are opened using the **Remote Display Number** global connection.

Use parameter files or parameter lists to specify which tags or folders to substitute for which placeholders. For global objects, you can specify the tags or folders of tags using global object parameters.

For more information about using parameter files, parameter lists, and global object parameters, see Use parameters and global objects on page 374.

Create tag placeholders

A tag placeholder is the cross-hatch character (#) followed by a number from 1 to 500.

The tag placeholder can replace any part of a tag name, including folder names. For example, you could create a parameter file specifying that the tag placeholder #1=Folder1. You could assign the folder and a tag name to a graphic object's connection: *#1\Tag1*.

You can assign tag placeholders:

- On the **Connections** tab of an object's **Properties** dialog box.
- On the **Connections** tab of the **Property Panel**.
- In the **Expression** box in the **Animation** dialog box.
- Anywhere that you can insert an embedded variable. For information about embedded variables, see Use embedded variables on page 367.
- A global object's **Parameter Definitions** dialog box.

To create a tag placeholder

- Type the cross-hatch character followed by a number (no space in between). For example, #1.

Perform basic operations on objects

Once you have drawn an object, you can select the object and work with it. You can:

- Move objects
- Copy objects
- Duplicate objects
- Resize objects
- Reshape drawing objects
- Delete objects

Move objects

You can move objects using the mouse or the keyboard. The keys give you fine positioning, allowing you to move objects in small increments. You can also use the grid to position objects precisely.

Another option is to position an object using the object's Top and Left properties in the **Property Panel**. For information about using the **Property Panel**, see Use the Property Panel on page 296.

You can also specify an object's position using the **Common** tab in the object's **Properties** dialog box. For more information, see Set up objects' spatial properties, names, and visibility on page 327.

Once you have moved objects into position, you might want to align other objects with them, or lock them into place.

For information about aligning objects, see Align objects on page 314.

For information about locking objects into position, see Lock objects into position on page 318.

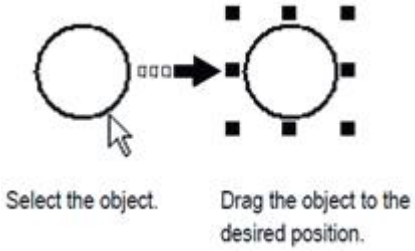
To automatically align objects to the grid as you move them

- From the **View** menu, select **Snap On**. A check mark appears beside the menu item when the option is selected.

For information about setting up the grid, see Use the grid on page 256.

To move objects by dragging with the mouse

1. Select one or more objects.
2. Place the pointer on an object (not on the edge or on the handles).
3. Drag the objects to the desired position.



4. If you selected several objects, dragging one of the objects moves all the selected objects. The objects maintain their position relative to each other.

Copy 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.

To copy objects, use one of these methods:

- 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.

Copy 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 Set up language switching on page 159.

Copy objects on the same display

You may need to copy objects on the same display.

To copy objects on 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.

Drag objects between displays

You may need to drag objects between displays.

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.
   For more information, see Resize displays on page 260.
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.

Cut or copy objects

You can cut or copy objects using the menu items on the **Edit** menu or the buttons on the toolbar.

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, select or click **Cut**.
   - To retain the original object, select or click **Copy**.

Paste objects

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 paste objects

1. Click the display or library to paste to.
2. From the **Edit** menu, select **Paste**, or on the **Graphics** toolbar, click  .

Duplicate objects

Use the **Duplicate** option on the **Edit** menu to make a copy of the selected object or objects. The duplicated object is placed slightly offset from the original.

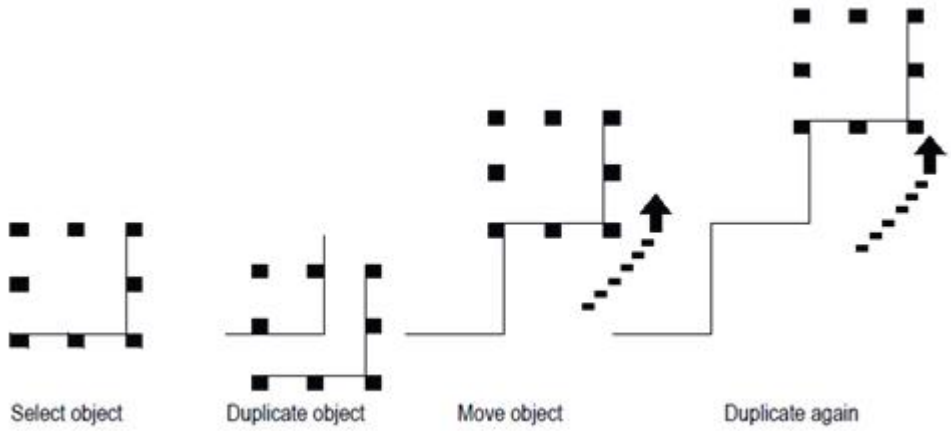
To duplicate objects

1. Select one or more objects.
2. Do one of the following:
   - From the **Edit** menu, select **Duplicate**.
   - Click  on the toolbar.

- Right-click the selected objects and select **Duplicate**.

Tip:

- If you duplicate an object, move it, and then select **Duplicate** again, the object will be duplicated and moved in one step. This is useful for creating a series of objects with an equal distance between them.
- Any animation attached to the object is also duplicated. If a group is duplicated, the new copy of the group can be ungrouped to individual objects.
- Duplicate works until you cancel the object's selection.
- Duplicate does not use the clipboard, so the clipboard's contents are not changed when you use this command.



Resize objects

You can resize objects using the mouse or using the keyboard. The keys let you resize objects in small increments. You can also use the grid to resize objects precisely.

Another option is to size an object using the object's Height and Width properties in the **Property Panel**. This method is especially useful for quickly resizing multiple objects to the same size. For information about using the **Property Panel**, see Use the Property Panel on page 296.

You can also specify an object's size using the **Common** tab in the object's **Properties** dialog box. For more information, see Set up objects' spatial properties, names, and visibility on page 327.

When you resize text objects, if you have selected the **Size to fit** option, the font size is adjusted to fit the new object size as closely as possible.

Automatically align objects to the grid as you resize them

To automatically align objects to the grid as you resize them

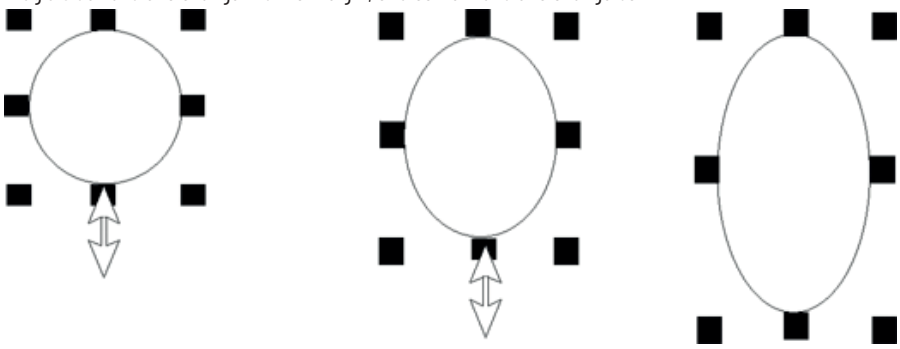
- From the **View** menu, select **Snap On**. A check mark appears beside the menu item when the option is selected.

For information about setting up the grid, see Use the grid on page 256.

Resize an object using the mouse

To resize an object using the mouse

1. Select the object.
2. Place the pointer on one of the handles.
   A double arrow appears.
3. Drag the handle until the object is the desired size or shape.
   Drag a side handle to change width or height, or a corner handle to change both.



For perfect circles and squares, press **Ctrl** and hold the key down while you drag a corner handle.

To maintain the object's original proportions (width to height), press **Shift** and hold the key down while you drag a corner handle.

Resize an object in small increments using the keyboard

To resize an object in small increments using the keyboard

1. Select the object.
2. Place the pointer on one of the handles.
   A double arrow appears.
3. Hold down **Shift** and press an arrow key until the object is the desired size.
   To adjust the amount of the increment, first hold down the **Shift** key and press the + or - keys on the keyboard's numeric keypad.

Reshape drawing objects

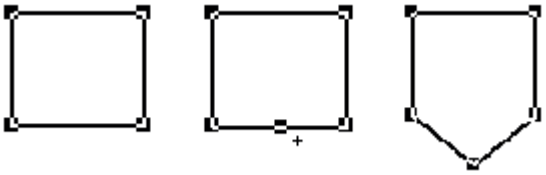
You can reshape arcs, lines, polygons, polylines, rectangles, rounded rectangles, and wedges.

Reshape lines, rectangles, polylines, and polygons

To reshape lines, rectangles, polylines, and polygons

1. Select the object that you want to reshape.
2. Select in the toolbox, or right-click the object and then select **Edit**.
   The cursor changes to the **Polygon** tool, and handles appear on the object.
3. Move the cursor over any line or corner of the object.
   A handle with a cross hair appears.

4. Drag the handle until the object is the desired shape.



Dragging from a point along the line (between corners) creates a new angle between the two corners.

5. To delete an angle, position the pointer at the tip of the angle, and then press **Delete**.

Reshape arcs and wedges

To reshape arcs and wedges

1. Select the object that you want to reshape.
2. Select or in the toolbox, or right-click the object and then select **Edit**.

   The cursor changes to the **Arc** or **Wedge** tool, and handles appear on the object.
3. Place the pointer on one of the handles.

   A cross hair appears.
4. Drag the handle until the object is the desired shape.

Reshape rounded rectangles

To reshape rounded rectangles

1. Select the rounded rectangle.
2. Select in the toolbox, or right-click the object and then select **Edit**.

   A handle appears inside the rounded rectangle.
3. Click the handle and drag inward to increase roundedness, or outward to decrease roundedness.

Delete objects

You can delete one or more objects from a graphic display.

To delete objects

1. Select one or more objects.
2. Do one of the following:
   - From the **Edit** menu, select **Delete**.
   - Right-click the object and select **Delete**.
   - Press the **Delete** key on the keyboard.

Tip:

- To delete all the objects on a graphic display, from the **Edit** menu, select **Clear All**.
- Delete does not use the clipboard. Use **Undo** to reverse an accidental deletion.

Group and ungroup objects

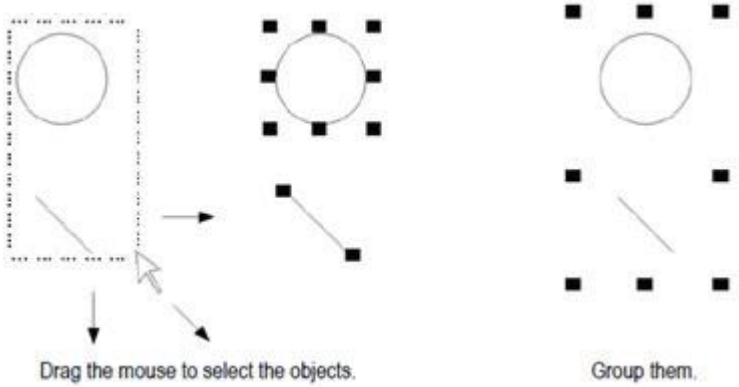
Grouping combines several objects into a single object so you can manipulate them as a single object. Grouping is useful for keeping objects in the same position relative to each other. You can cut, copy, and paste groups, arrange the group as a single object relative to other objects, and apply the same properties to all the members of the group at once.

You can attach animation to a group, and any animation attached to individual objects in the group remains active. The group animation generally takes precedence over the animation of individual objects within the group. For more information, see Apply animation to groups on page 349.

Deleting a group deletes all individual objects in the group.

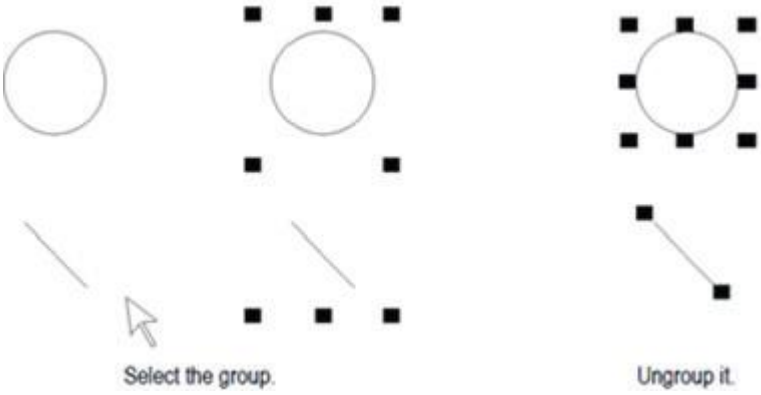
To group objects

1. Select all the objects you want grouped.
2. From the **Arrange** menu, select **Group**, or on the **Graphics** toolbar, click .
3. The handles around each object are replaced by a set of handles around the group.



To ungroup objects

1. Select the group of objects to ungroup.
2. From the **Arrange** menu, select **Ungroup**, or on the **Graphics** toolbar, click .
   The handles around the group are replaced with handles around each object.



Ungrouping deletes any animation attached to the group, because the group no longer exists. However, animation attached to the individual objects that were in the group remains active.

Edit groups of objects

Edit a group the same way that you would edit an individual object. You can:

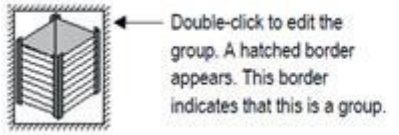
- Use the **Property Panel** to apply the same properties to all the members of the group at once. For example, change the line width of all objects in the group to two pixels.
  For information about using the **Property Panel**, see Use the Property Panel on page 296.
- Use the toolbars to apply the same pattern style, background style, foreground colors, and background colors to all the members of the group.
- For objects with states, use the **States** toolbar to cycle through the states and apply the same properties to the states for each object in the group at once. When you select a group containing objects with states, only the states that are common to all objects in the group appear on the toolbar.

Edit objects within a group

You can also edit individual objects within the group without breaking the group, which is particularly useful when you have animation attached to the group.

To edit objects within a group

1. Double-click the grouped object, or right-click it and then click **Edit**. A hatched border appears around the group.
   When the hatched border is around the group, you are in group edit mode. In this mode, you can select individual objects in the group and modify them.
   You can also add new objects to the group.



2. To select an individual object (or a group) in the group, click it. You can also use the **Object Explorer** to select objects within the group.
   The status bar and **Object Explorer** indicate which object or group is selected.
3. Make your changes to the object.
   You can change the selected object's shape, size, or position, or use the object's **Properties** dialog box or the **Property Panel** to edit the object's properties.
4. If desired, create new objects inside the hatched border.
5. To stop editing, click outside the group.

Arrange objects

You can arrange objects (or groups of objects) in a number of ways. You can:

- Layer objects by moving them in front of or behind other objects.
- Align objects with each other and with the grid.
- Space objects horizontally or vertically.
- Flip drawing objects horizontally or vertically.
- Rotate drawing objects.
- Lock objects into position.

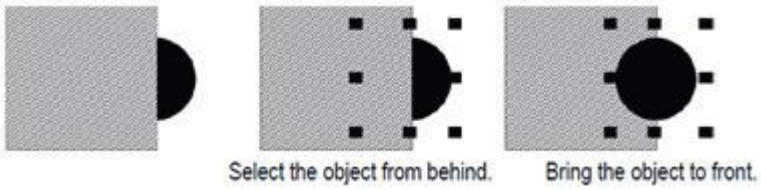
Layer objects

You can layer objects (or groups of objects) on top of each other. Objects are layered in the order they are created, with the most recently created object on top. Change the layer order with **Send to Back** and **Bring to Front**.

- **Send to Back** moves the selected object to the bottom layer.
- **Bring to Front** moves the selected object to the top layer.

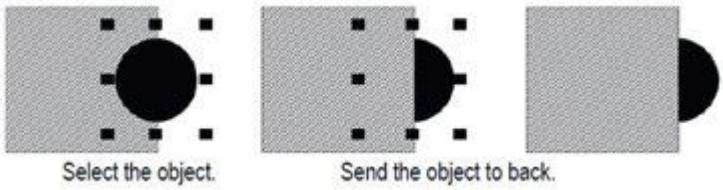
To bring an object to the front

1. Select the object.
   To select an object that's behind another object, place your pointer on the front object, click once, pause, and then click again. Do not double-click and do not move the mouse.
   You can also select a concealed object easily by clicking the object in the **Object Explorer**.
2. From the **Arrange** menu, select **Bring to Front**, or click the **Bring to Front** tool .



To send an object to the back

1. Select an object.
2. From the **Arrange** menu, select **Send to Back**, or click the **Send to Back** tool .



Align objects

You can align objects (or groups of objects) with each other and with the grid.

To align objects

1. Select the objects you want to align.
2. From the **Arrange** menu, select the appropriate menu item, or click a tool on the **Alignment** toolbar:

This button or menu item	Aligns selected objects with the
Align Left	Left-most selected object
Align Center	Horizontal center of all selected objects
Align Right	Right-most selected object
Align Top	Top-most selected object
Align Middle	Vertical center of all selected objects
Align Bottom	Bottom-most selected object

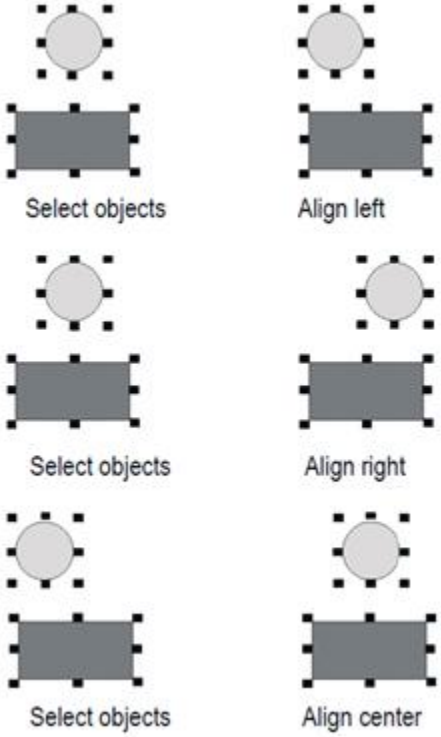
This button or menu item	Aligns selected objects with the
Align Center Points	Center of all selected objects
Align to Grid	Grid

To automatically align objects to the grid as you create or move them

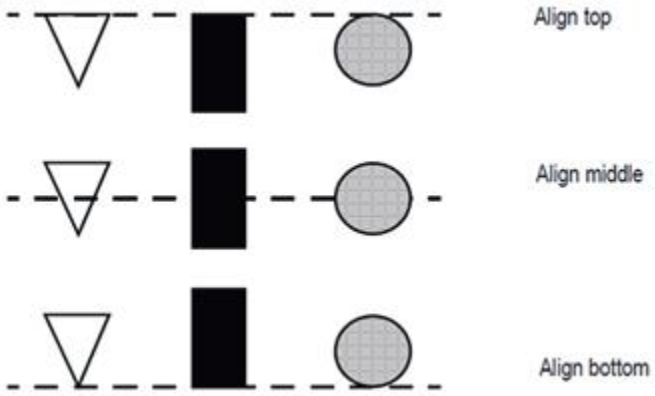
- From the **View** menu, select **Snap On**. A check mark appears beside the menu item when the option is selected.

For information about setting up the grid, see Use the grid on page 256.

Examples: Align objects left, right, and center



Examples: Align objects top, middle, and bottom



Space objects

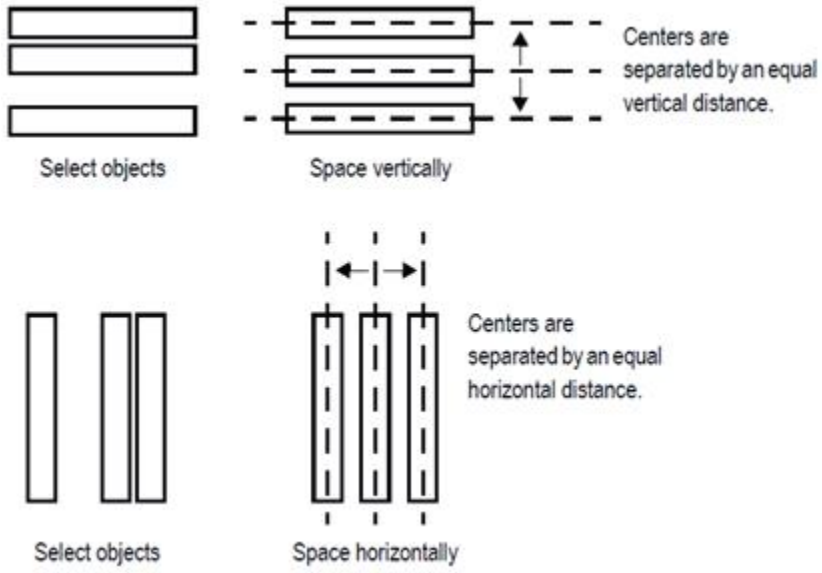
With Space Vertical and Space Horizontal, objects (or groups of objects) are moved vertically or horizontally to have the same amount of space from the center point of each object.

To space objects

1. Select the objects you want to space.
2. From the **Arrange** menu, select a **Space** menu item, or click a tool on the **Graphics** toolbar:

This tool or menu item	Does this
Space Horizontal	Places the centers of the selected objects an equal distance apart horizontally.
Space Vertical	Places the centers of the selected objects an equal distance apart vertically.

Examples: Space objects vertically and horizontally



Flip drawing objects

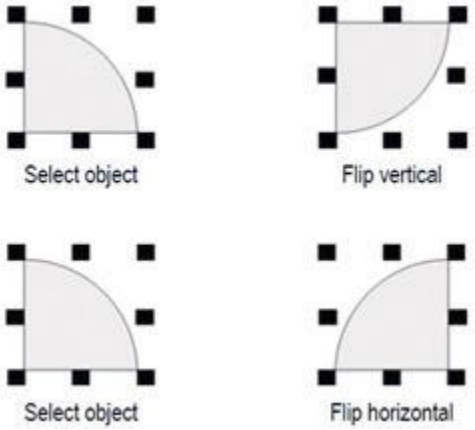
Flipping an object creates a mirror image of the object. You can flip all the drawing objects (or groups of drawing objects) except text, images, and panels.

To flip a drawing object

1. Click the object.
2. From the **Arrange** menu, select a **Flip** menu item, or click a tool on the **Graphics** toolbar:

This tool or menu item	Flips selected objects
Flip Vertical	Top to bottom (upside-down)
Flip Horizontal	Left to right

Examples: Flip drawing objects vertically and horizontally



Rotate drawing objects

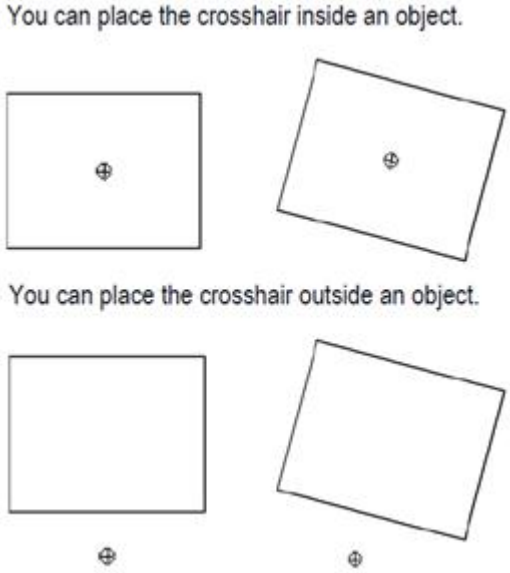
You can rotate all the drawing objects (or groups of drawing objects) except images, panels, and rounded rectangles.

You can attach rotation animation to the same drawing objects. With rotation animation, the object rotates around an anchor point to indicate a tag's value at runtime. For details about rotation animation, see Set up rotation animation on page 348.

When you rotate text, it rotates around the anchor point but the text itself remains upright.

To rotate a drawing object

1. From the **Objects** menu, select **Rotate**, or from the toolbox, select .
2. Click the object you want to rotate. A small crosshair circle appears in the middle of the object. This is the anchor point that is used as the center of rotation.



3. To move the center of rotation, click the cross-hair and drag it to a new anchor position. The anchor can be inside or outside the object.
4. Click an edge of the object and drag in the direction you want to rotate it.
   To rotate the object in five-degree increments, press **Ctrl** while you drag.
5. When the object is in the desired position, release the mouse button.

Lock objects into position

You can lock graphic objects (or groups of objects) into position by converting them to wallpaper. Once you convert objects to wallpaper, you cannot select or edit them unless you unlock the wallpaper. Wallpaper objects cannot be animated at runtime.

If the grid is on, wallpaper objects are positioned behind the grid. If you just want to lock the objects into place while you're working on the display, unlock the wallpaper when you're finished. If you want to use the wallpaper objects as a background for your display, leave the wallpaper locked. For more information about creating a background for your display, see Create a background for your display on page 261.

Set up graphic objects

This chapter describes how to set up a graphic object, including:

- Set up a graphic object properties
- Set up how a graphic object is used at runtime
- Time, date, and number formats for a graphic object

Set up a graphic object

To set up a graphic object

1. Double-click the graphic object.
2. In the **Properties** dialog box, specify the options on each tab. See Graphic object properties on page 319.
3. When you are finished, click **OK**.

For information on how to set up each graphic object, see FactoryTalk View ME Help.

For information on how to use each graphic object, see Use graphic objects on page 274.

Graphic object properties

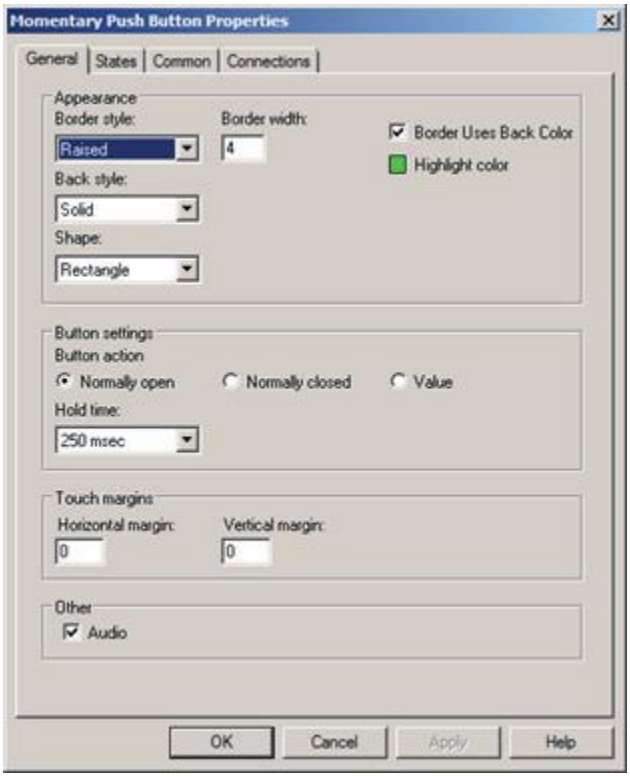
A graphic object may have several of the following tabs in the **Properties** dialog box:

- General tab
- Alarm tab
- Audit tab
- Display tab
- E-Signature tab
- States tab
- Label tab
- Print tab
- String tab
- Numeric tab
- Timing tab
- Common tab
- Connections tab

General tab: Set up an object's runtime appearance

On the **General** tab, specify what the object looks like at runtime.

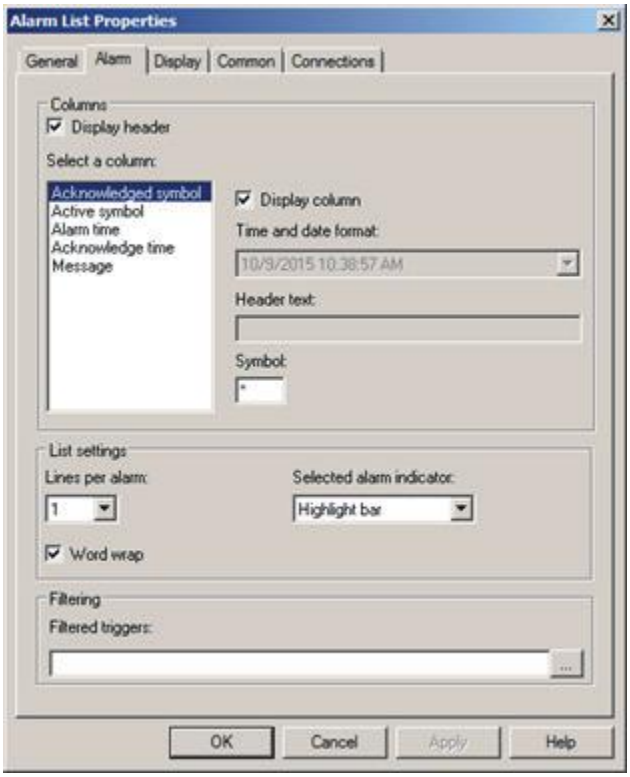
Every graphic object has a **General** tab in its **Properties** dialog box.



Alarm tab: Set up the alarm properties

On the **Alarm** tab, set up the following:

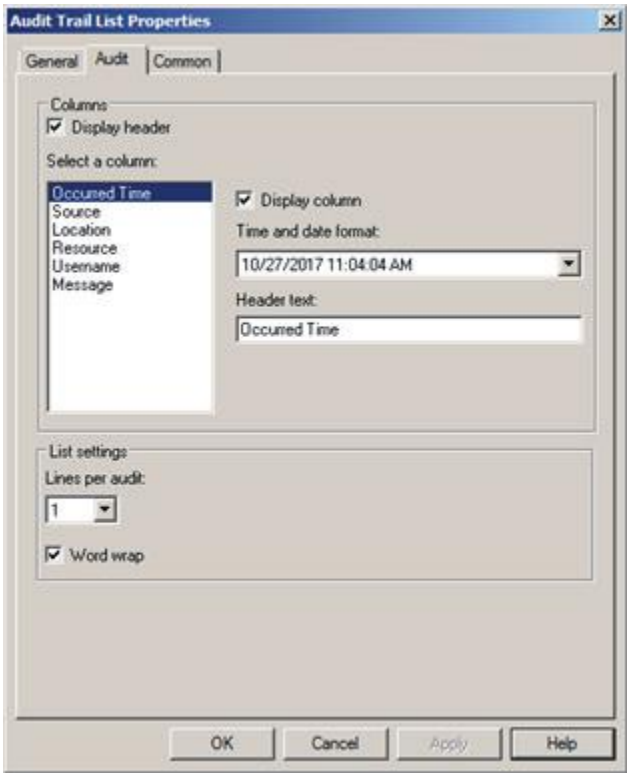
- the columns to include in the list
- the number of lines per alarm
- the triggers to filter by, if any
- whether to queue alarms
- whether to show all alarms or active alarms only
- which alarms to include in a filtered trigger list



Audit tab: Set up appearance for an audit object

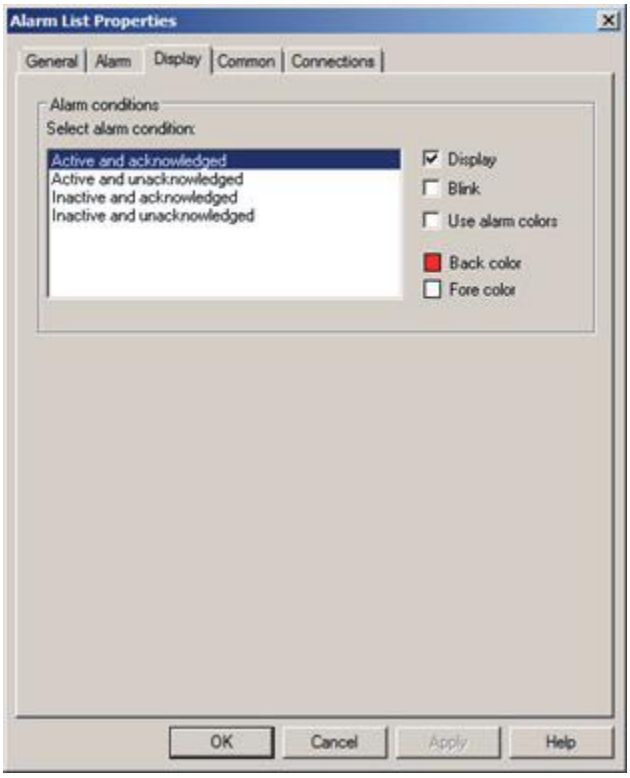
On the **Audit** tab, set up the following:

- The columns or rows to include in the object
- The number of lines per audit in the list
- Whether to use word wrap



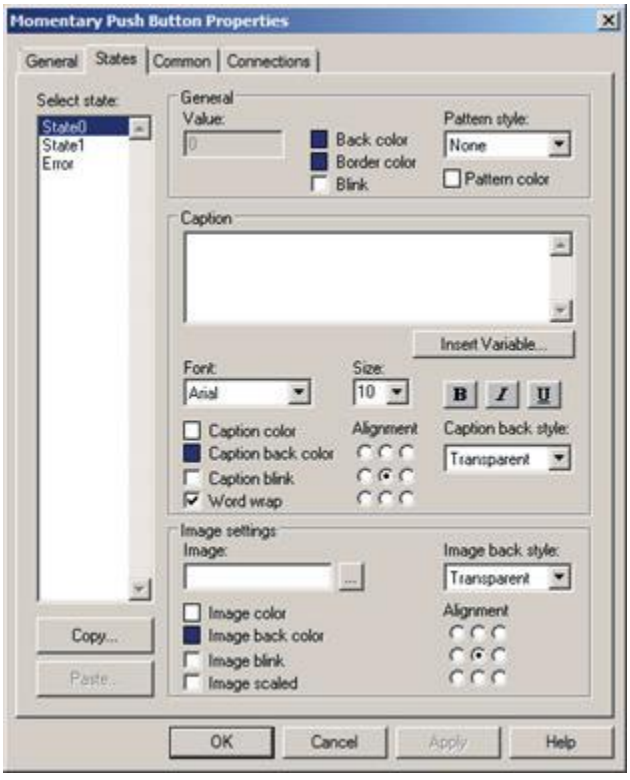
Display tab: Set up how the object shows at runtime

On the **Display** tab, specify what to show at runtime.



States tab: Set up an object with states

On the **States** tab, set up the value and caption for each state.

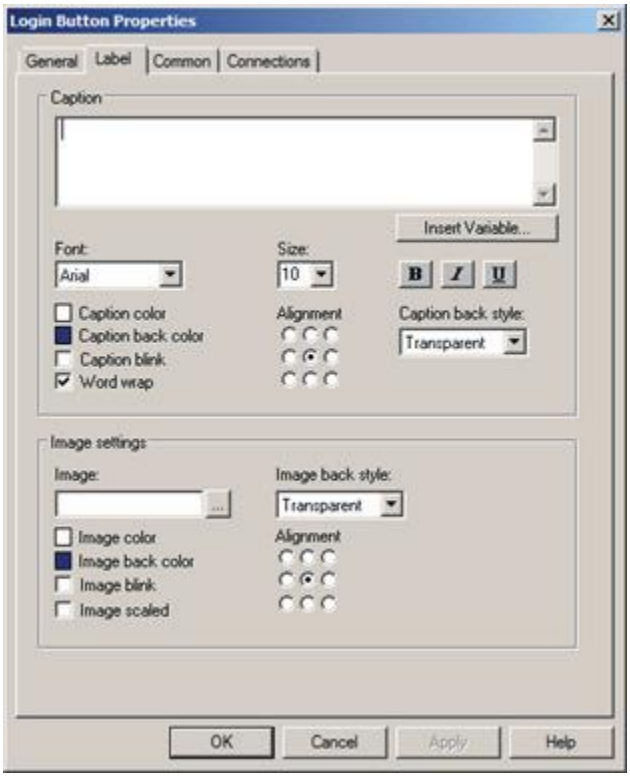


To simplify setting up states when many of the properties are the same, you can copy and paste settings from one state to another. For details, see Help.

Some graphic objects have a configurable number of states. For these objects, you can use the **Insert State** and **Delete State** buttons on the **States** tab, to add and remove states without returning to the **General** tab. The **Number of States** setting on the **General** tab is automatically updated. For more information about the **Insert State** and **Delete State** buttons, see Help.

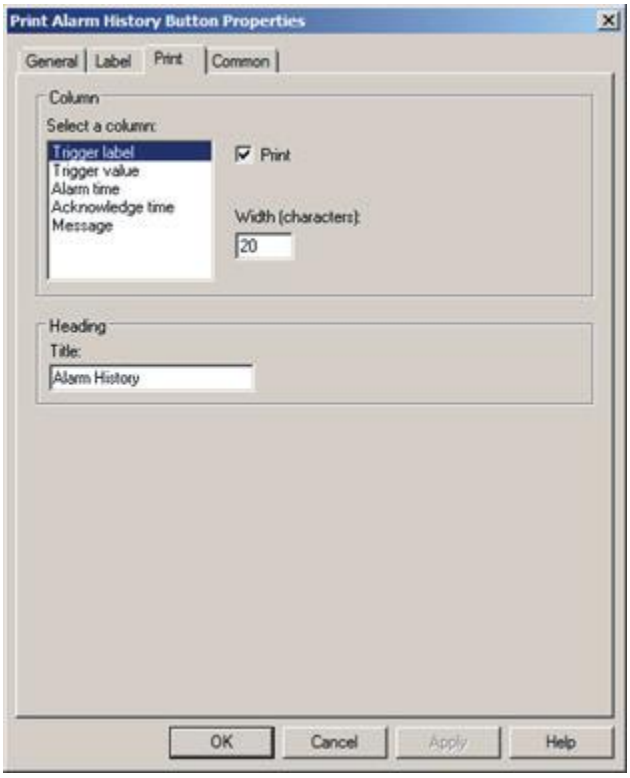
Label tab: Set up what text or image shows on a button object

On the **Label** tab, specify what text or image to show on the button.



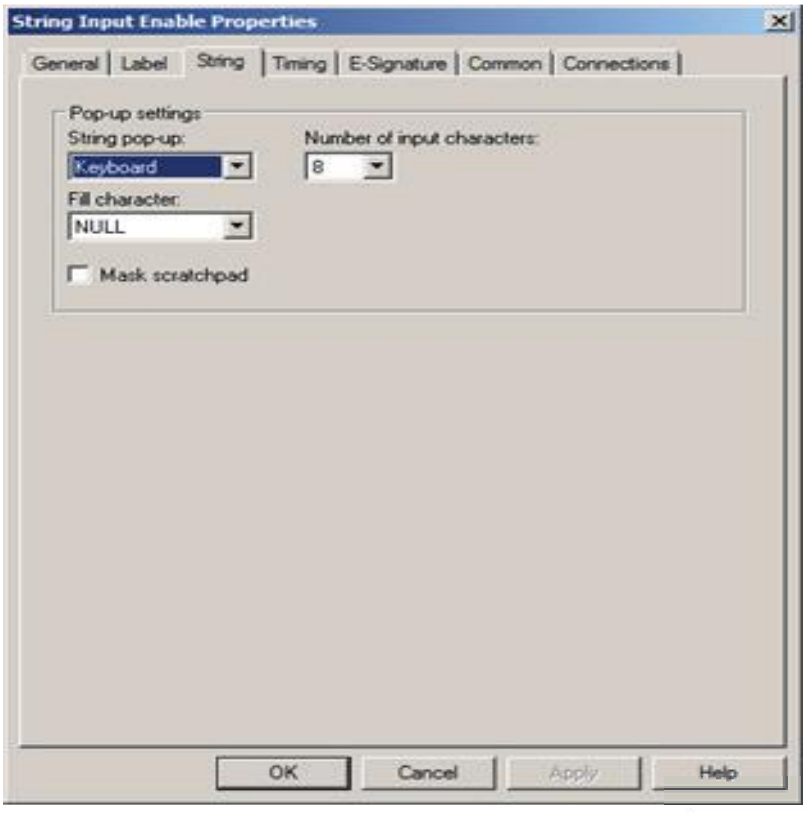
Print tab: Set up what information to print

On the **Print** tab, specify what information to print on the report.



String tab: Set up the popup windows to open and number of characters to accept

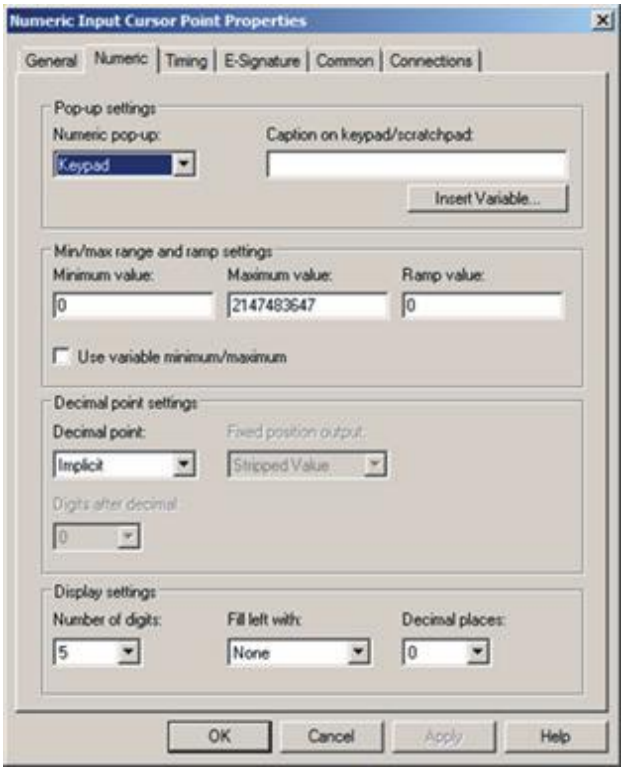
On the **String** tab, specify the window to open and the number of characters to accept.



Numeric tab: Set up open windows, ramp values, minimum and maximum values, and decimal point settings

On the **Numeric** tab, set up the following:

- which window to open, if any
- which caption, if any, will be shown in the open window
- the ramp value, if any
- minimum and maximum values to send to the data source
- whether the minimum and maximum values will be variable (from tags or expressions) or constant
- decimal point settings
- display settings



Timing tab: Set up a button object's auto repeat

On the **Timing** tab, set up the following:

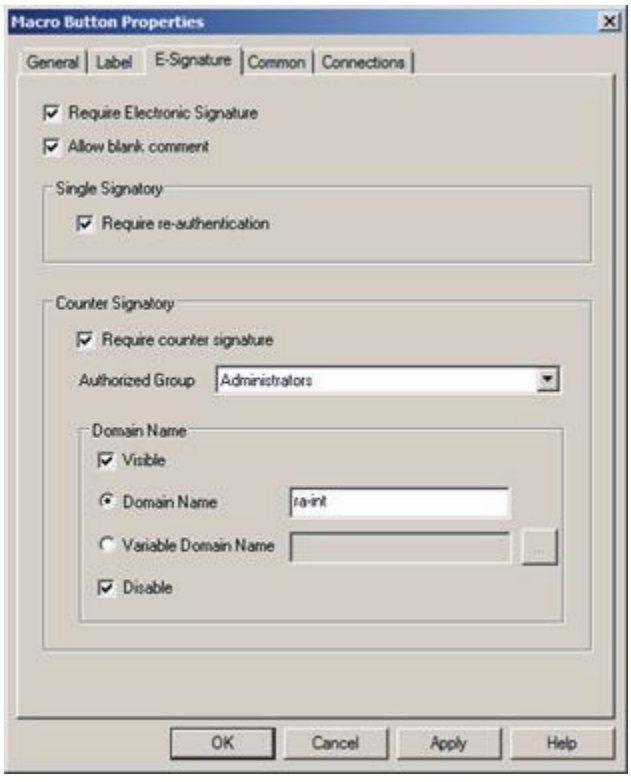
- Whether or not the button press repeats automatically when the operator presses and holds the button down. You can also set up the rate at which the button press repeats.
- The timing and handshake settings for the **Enter** key. These settings do not apply when you ramp a value.

For more information about auto repeat, see Repeat a button's action by holding down the button on page 334 and Set up auto repeat for a button on page 335.

For information about using Enter key handshaking, see Methods of Enter key handshaking on page 335.

E-Signature tab: Set up an object's electronic signature properties

Use the **E-Signature** tab of the **Properties** dialog box to configure electronic signature capability to control and record operator actions.



For information about the options on the **E-Signature** tab, see Help.

For information on how to use electronic signature, see Use electronic signature on page 234.

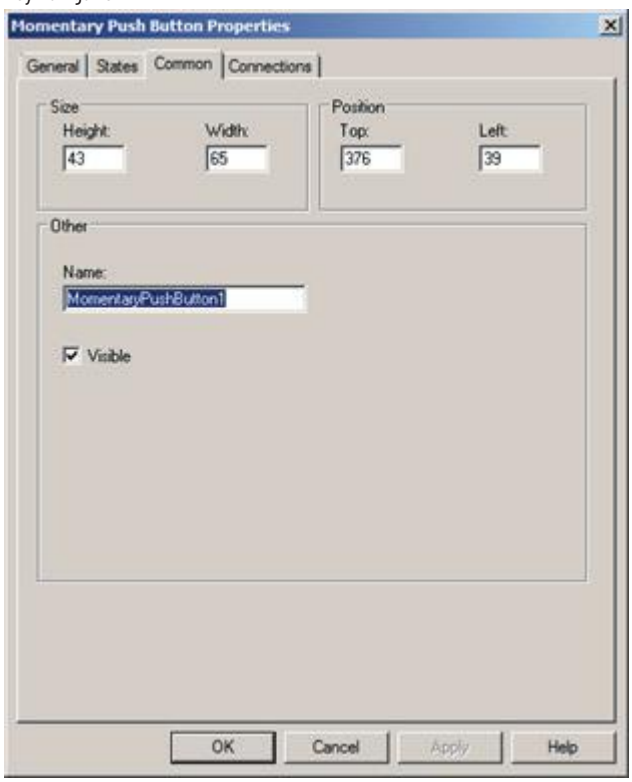
Common tab: Set up an object's spatial properties, names, and visibility

Use the **Common** tab to set up the following properties for the object:

- Height and width
- Top and left position
- Name
- Visibility

Every graphic object has a **Common** tab in its **Properties** dialog box. For ActiveX and trend objects, you can also set up the following properties:

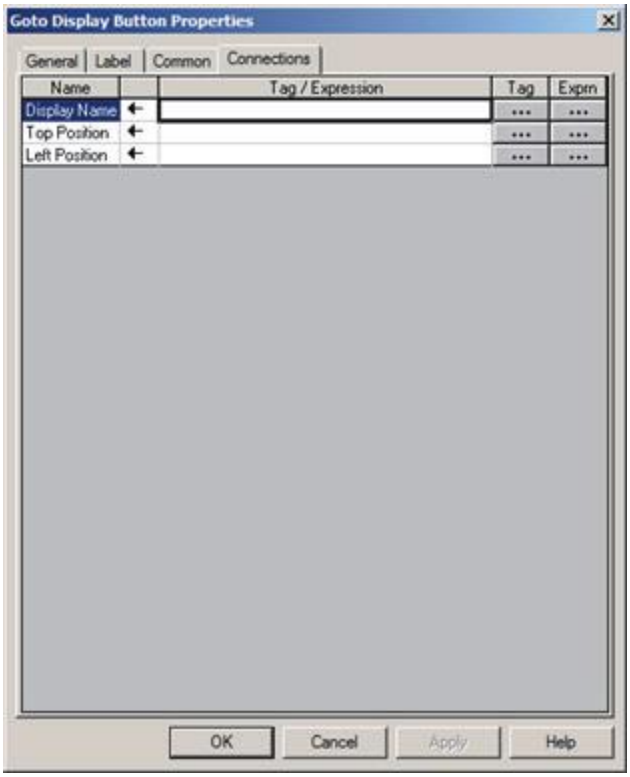
- Focus highlight
- Key navigation



Connections tab: Set up tags or expression for an object

On the **Connections** tab, specify the tags or expression whose data is shown or with which the object exchange data.

For more information, see Assign tags and expressions to graphic objects on page 301.



Set up how a graphic object is used at runtime

Just as you must provide operators with a way to navigate between displays at runtime, you must also make sure that operators can use the objects on the displays.

This section describes how to set up your objects so the operator can:

- Press objects using a touch screen.
- Use function keys to press buttons when a mouse or touch screen is not available on the runtime computer.
- Use the keyboard or keypad to navigate to and select lists, trends, and ActiveX input objects.
- Use buttons to work with lists and trends.
- Repeat a button's action by pressing and holding it.
- Ensure that tag values are read by the data source before sending new values.

Position objects for touch screens

If the operator will be using a touch screen at runtime, keep these tips in mind when positioning graphic objects in your displays:

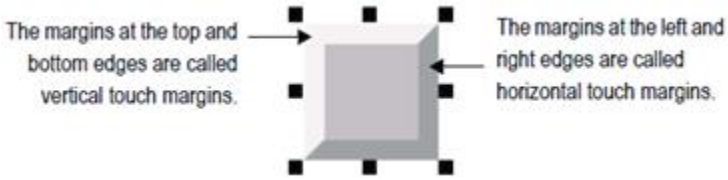
- Don't place important buttons where they'll be blocked by an On Top display. The user can't press a covered button.
- Ensure that buttons are large enough for users to touch easily.
- Use touch margins for buttons that are positioned close together, to ensure that the adjacent button is not pressed by mistake.

Use touch margins

Touch margins are touch-insensitive borders inside the button's margin. If the operator presses on the touch margin, the button press is not registered. Touch margins are useful when buttons are positioned close to each other and you want to make sure the operator does not press the wrong button by mistake.

You can create touch margins at the top and bottom of the button, at the sides, or on all four sides.

In the illustration below, the button's border and touch margins are the same size, 12 pixels. A button press would be registered only when the darker square in the middle of the button is pressed.



If the object's shape is a circle or ellipse, the touch margin applies to the object's bounding box, not the object's border. The bounding box is an invisible square or rectangle that surrounds the object. When you select the object, the selection handles show the location of the bounding box.

To create touch margins, use one of these methods

- On the **General** tab of the button's **Properties** dialog box, type the number of pixels for the touch margins in the **Horizontal margin** and **Vertical margin** boxes.
- On the **Properties** tab of the **Property Panel**, type the number of pixels for the **HorizontalMargin** and **VerticalMargin** properties.

Assign function keys to buttons

You can assign function keys to the buttons in your displays to allow the operator to press the buttons using the function keys on the runtime terminal (or the function keys on a keyboard, if one is available). You can also assign a function key to the numeric input cursor point.

You can assign up to 34 function keys to each graphic display.

Here are some tips for assigning function keys:

- Where possible, use the same function keys for the same operations in all your graphic displays. For example, if every display contains a Goto display button that returns the operator to a graphic display called Main Menu, assign the same function key to this button in each display.
- Include the name of the function key assigned to a button in a caption on the button.

For buttons with multiple states, include the function key name in the caption for each state, or create a text object to use as a label for the button (so that you don't have to set up the caption for each state), and then group the text and button together.

- If your application will run on a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, assign keys that are supported by the runtime terminal. Different sizes of terminals have different function keys.

Function key equivalents

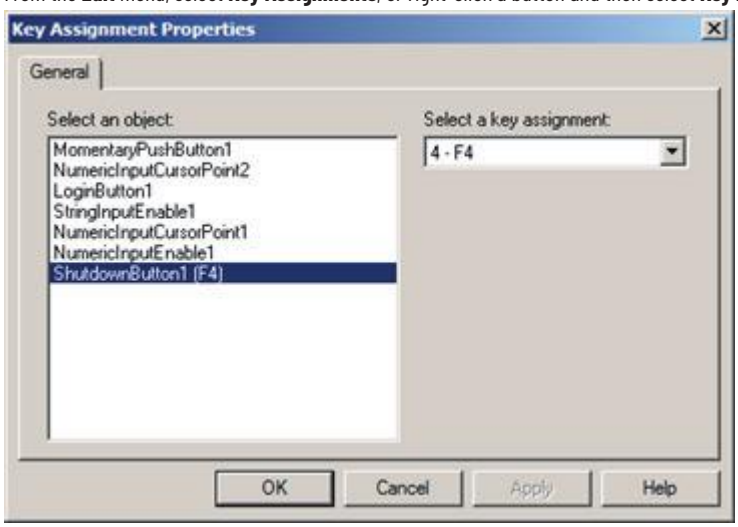
If your application will run on a personal computer, the keyboard contains the function keys F1 to F12 only. The remaining function keys are associated with these key combinations:

For this function key	Use this key combination
F13	Left **Shift+F1**
F14	Left **Shift+F2**
F15	Left **Shift+F3**
F16	Left **Shift+F4**
F17	Left **Shift+F5**
F18	Left **Shift+F6**
F19	Left **Shift+F7**
F20	Left **Shift+F8**
K1 - K12	Right **Alt+F1** - right **Alt+F12**
K13	Right **Shift+F1**
K14	Right **Shift+F2**
K15	Right **Shift+F3**
K16	Right **Shift+F4**
K17	Right **Shift+F5**
K18	Right **Shift+F6**
K19	Right **Shift+F7**
K20	Right **Shift+F8**

At runtime, the operator presses the key combination to activate the object to which the function key is assigned.

To assign function keys to buttons

- From the **Edit** menu, select **Key Assignments**, or right-click a button and then select **Key Assignments.**



For details about using the **Key Assignment Properties** dialog box to assign function keys and change function key assignments, see Help.

Use the keyboard to navigate to and select objects

If a mouse or touch screen is not available on the runtime computer, the operator can use the keys on a keyboard or keypad to select (give focus to) these objects:

- Lists: control list selector, piloted control list selector, display list selector, diagnostics list, alarm list, and alarm status list
- Alarm banners
- Trends
- Numeric input enable buttons and String Input Enable buttons
- Numeric input cursor points
- RecipePlus table and selector
- Third-party ActiveX input objects

What input focus looks like

The object with focus is surrounded by a highlight box, unless the Disable Highlight When Object has Focus box is selected (in the **Display Settings** dialog box). You can specify the color of the highlight in this dialog box as well.

For ActiveX and trend objects, use the **Common** tab in the object's **Properties** dialog box to specify whether to show a highlight box.

If the **Disable Highlight When Object has Focus** checkbox is selected in the **Display Settings** dialog box, that setting overrides the setting you specify on the **Common** tab.

For more information about setting up options on the **Common** tab, see Help.

Use the keys on the keyboard or keypad

When a graphic display opens, the keyboard-navigable object that is closest to the top left corner of the display is selected. The operator can use these keys to move to and select a different object:

Use this key	To do this
Tab	Move from the upper left to the lower right.
Shift+Tab	Move from the lower right to the upper left.
Ctrl+arrow key	Move left, right, up, or down.

Remove objects from and adding objects to the tab sequence

By default, you can use the keys to navigate to all lists, alarm banners, numeric input cursor points, trends, and ActiveX input objects in a display. However, you can turn off key navigation for these objects if desired.

When an object's key navigation is turned off, the operator can still select the object using a mouse or touch screen.

By default, key navigation is turned off for the Numeric Input Enable buttons and **String Input Enable** buttons. You can turn on keyboard navigation if you want the operator to use the keyboard to navigate to these objects.

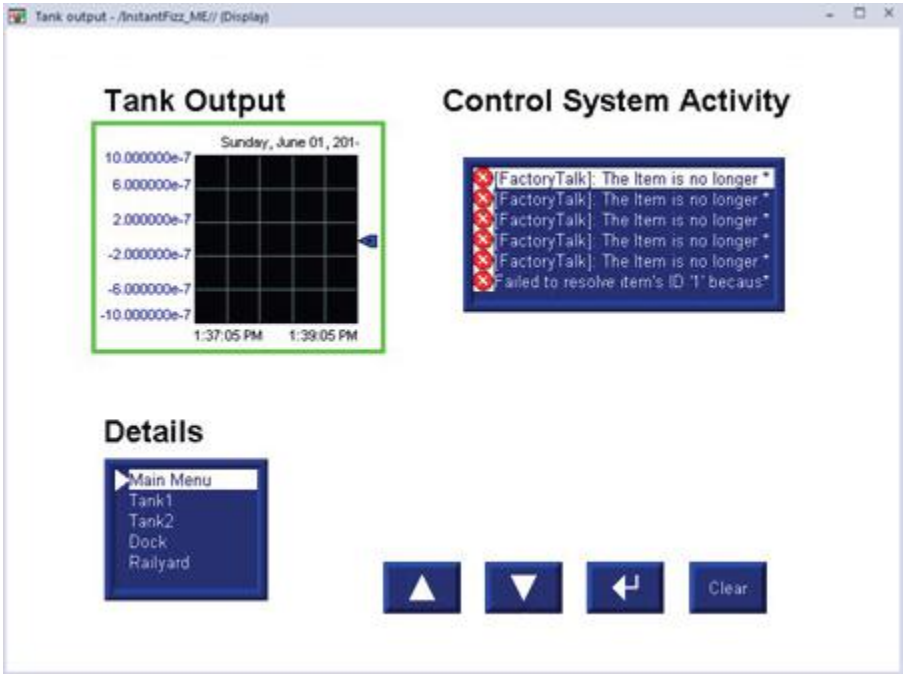
To turn key navigation on or off, use one of these methods:

- On the **Properties** tab of the **Property Panel**, set the **KeyNavigation** property to **False** for off, or **True** for on.
- For list objects, alarm banners, numeric input cursor points, **Numeric Input Enable** buttons, and **String Input Enable** buttons, on the **General** tab of the object's **Properties** dialog box, clear the **Key navigation** box for off, or select the box for on.
- For ActiveX and trend objects, on the **Common** tab of the object's **Properties** dialog box, clear the **Key** navigation box for off, or select the box for on.

Link buttons to objects

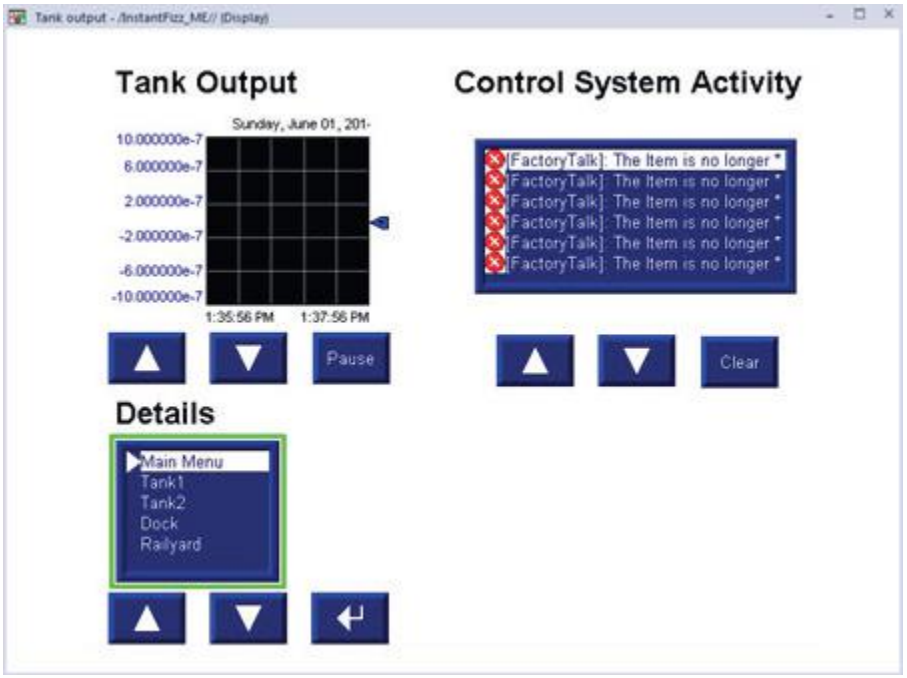
Some FactoryTalk View buttons can be linked to specific trend, list, recipe, and alarm banner objects, or you can set them up to work with whichever object has focus on the display. By default, the buttons are set up to work with whichever object has input focus on the display.

In the following example, the Tank Output object has focus, and the move up and move down buttons work with this object.



If you are creating small graphic displays, you might prefer to use one set of buttons to work with all the display's graphic objects that accept input focus.

If space is sufficient, you can create multiple copies of the buttons and link them to specific objects. One benefit of linking a button to a specific object is that the operator does not have to select the object before pressing the button. Another benefit is that you can position the buttons close to the specified object, making it easier for the operator to understand which buttons work with which objects.



Link a button to a specific object using the button's Properties dialog box

To link a button to a specific object using the button's Properties dialog box

1. Double-click the button to open its **Properties** dialog box.
2. On the **General** tab, in the **Send press to** box, select **Linked Object**.
3. To select from a list of all the objects on the display that you can link the button to, click the **Browse** button next to the **Linked object** box.
4. Click the name of the object to link the button to, and then click **OK**.
5. Click **OK** to close the button's **Properties** dialog box.

Link a button to a specific object using the Property Panel

To link a button to a specific object using the Property Panel

- On the **Properties** tab, specify the **SendPressTo** and **LinkedObject** properties.

Once you have linked buttons to an object, you might want to turn off the object's key navigation, since this option is no longer needed. For details, see Remove objects from and adding objects to the tab sequence on page 332.

Repeat a button's action by holding down the button

To repeat a button's action by pressing and holding it, set up auto repeat for the button. If you set up auto repeat, when the operator presses and holds down the button, repeated button presses are registered until the operator releases the button.

A button press occurs when the operator clicks an object with the mouse, presses it on a touch screen, or presses the function key associated with the object. Auto repeat works with all these methods of pressing buttons.

You can use auto repeat with these buttons:

For this button	Each button press does this
Multistate push	Sends the value for the next state to the data source. External changes to the **Value** connection are not recognized when the button is in auto repeat mode.
Ramp	Sends the new ramped value to the data source. External changes to the **Value** connection are not recognized when the button is in auto repeat mode.
Move up	Moves the highlight up an item in the list, recipe selector, or recipe table, scrolls up in the trend, or ramps the numeric input cursor point or Numeric Input Enable button value.
Move down	Moves the highlight down an item in the list, recipe selector, or recipe table, scrolls down in the trend, or ramps the numeric input cursor point or Numeric Input Enable button value.
Page up	Moves the highlight up a page in the list, recipe selector, or recipe table.
Page down	Moves the highlight down a page in the list, recipe selector, or recipe table.
Move left	Scrolls the trend to the left.
Move right	Scrolls the trend to the right.

For each button that uses auto repeat, you can specify these properties:

- **Auto repeat rate** – the number of times per second a button press is registered when the button goes into auto repeat mode. The default rate is 0, which means that auto repeat is turned off.
- **Auto repeat delay** – the length of time the button has to be pressed and held down before auto repeat starts.

Set up auto repeat for a button

To set up auto repeat for a button, use one of these methods:

- On the **Timing** tab of the button's **Properties** dialog box, specify the **Auto repeat rate** and **Auto repeat delay** properties.
- On the **Properties** tab of the **Property Panel**, specify the **AutoRepeatRate** and **AutoRepeatDelay** properties.

Ensure that values are read by the data source before sending new values

To ensure that a value is read by the data source before the operator sends a new value, use **Enter key handshaking**. While **Enter key handshaking** is in effect for an object, the operator cannot send a new value to the object's **Value** connection.

You can use **Enter key handshaking** with these graphic objects:

- Control list selector
- Piloted control list selector
- Numeric input enable button
- String Input Enable button
- Numeric input cursor point

If **Enter key handshaking** is in effect for one of these objects, the operator can still provide input to other objects in the graphic display.

If the graphic display is closed while **Enter key handshaking** is in effect, the handshaking is canceled.

Methods of Enter key handshaking

Enter key handshaking works by setting the object's **Enter** connection to 1. As long as the **Enter** connection is set to 1, new values cannot be sent to the **Value** connection. How the **Enter** connection is reset to 0 depends on how you set up **Enter key handshaking**.

There are two ways you can use **Enter key handshaking**:

- To hold the value at the data source for a specific period of time.
- To hold the value at the data source until the data source notifies FactoryTalk View that the value has been read.

Choose the method that best suits your application needs and communication system.

Hold the value for a specific period of time

To set up an object's **Enter key handshaking** so that the value at the **Value** connection is held for a specific period of time, assign a tag to the **Enter** connection and specify the **Enter** key hold time. You can also specify an **Enter** key control delay, if desired.

This method of **Enter key handshaking** works as follows:

1. When the operator presses the **Enter** button, the value is sent to the **Value** connection and the **Enter key control delay** timer begins timing. (The use of a delay is optional.)
2. If you specify an **Enter** key control delay, when the time is up, the **Enter** connection is set to 1. If you don't use the delay, the **Enter** connection is set to 1 as soon as the operator presses **Enter**.
   As long as the **Enter** connection is set to 1, the operator cannot send new values to the data source.
3. When the **Enter** connection is set to 1, the **Enter key hold time** timer begins timing.
4. When the **Enter** key hold time has expired, the **Enter** connection is reset to 0 and the operator can send a new value to the **Value** connection.

To set up Enter key handshaking to hold the value for a specific period of time

1. On **Timing** tab of the object's **Properties** dialog box, specify the **Enter key control dela**y (optional) and **Enter key hold time** properties.
2. On the **Connections** tab, assign a tag to the **Enter** connection. A digital tag is recommended (either an HMI tag or a data server tag).

You can also use the **Property Panel** to specify the properties and assign a tag to the **Enter** connection.

Hold the value until it is acknowledged

To set up an object's **Enter key handshaking** so that the value at the **Value** connection is held until the data source notifies FactoryTalk View that it has read the value, use two connections: the **Enter** connection and the **Enter handshake** connection.

Instead of using an **Enter key hold time**, specify an **Enter key handshake time**. You must also specify the Handshake reset type. You can use an **Enter key control delay**, if desired.

How the **Enter handshake** connection resets the **Enter** connection depends on which **Handshake reset type** you select:

With this handshake reset type	The Enter connection is set to 0 when
Non-zero Value	The **Enter handshake** connection has a non-zero value. If the **Enter handshake** connection already has a non-zero value when the value is sent to the **Value** connection (or when the **Enter** key control delay has expired, if the delay is used), then the **Enter** connection is not set to 1, and **Enter key handshaking** does not take place.
Zero to Non-zero transition	The **Enter handshake** connection changes from 0 to a non-zero value.

Set up the data source to send a non-zero value to the **Enter handshake** connection when it has read the new value at the **Value** connection, and then to reset the **Enter handshake** connection to 0.

If the **Enter key handshake time** expires before the **Enter handshake** connection resets the **Enter** connection, an error message is sent to FactoryTalk Diagnostics.

If you use the **Enter handshake** connection, handshaking works like this:

1. When the operator presses the **Enter** button, the value is sent to the **Value** connection and the **Enter key control delay** timer begins timing. (The use of a delay is optional.)
2. If you specify an Enter key control delay, when the time is up, the **Enter** connection is set to 1. If you don't use the delay, the **Enter** connection is set to 1 as soon as the operator presses **Enter**.
   As long as the **Enter** connection is set to 1, the operator cannot send new values to the data source.
   If the Handshake reset type is Non-zero Value, the **Enter handshake** connection must be 0 when the delay expires in order to set the **Enter** connection to 1.
3. When the **Enter** connection is set to 1, the **Enter key handshake time** timer begins timing.
4. The **Enter** connection remains set until the **Enter key handshake time** expires or until reset by the **Enter handshake** connection, whichever happens first.
5. The **Enter** connection is reset to 0 and the operator can send a new value to the **Value** connection.

To set up Enter key handshaking to hold the value until the data source has read it

1. On the **Timing** tab of the object's **Properties** dialog box, specify these properties:
   - **Enter key control delay (optional)**
   - **Enter key handshake time**
   - **Handshake reset type**
2. On the **Connections** tab, assign these connections:
   - **Enter** — assign a tag. A digital tag is recommended (either an HMI tag or a data server tag).
   - **Enter handshake** — assign a tag or expression.
3. Set up the data source to send a non-zero value to the **Enter handshake** connection when it has read the new value at the **Value** connection, and then to reset the **Enter handshake** connection to 0.

You can also use the **Property Panel** to specify the properties and assign tags to the connections.

Time, date, and number formats for graphic objects

Graphic objects use the time, date, and number format of the current application language. For example, if the application language uses a comma for the decimal symbol, floating-point values shown in graphic objects uses a comma for the decimal symbol.

For information about using multiple languages, see Set up language switching on page 159.

Animate graphic objects

This chapter describes:

- The types of animation and which objects support which types of animation
- How to use the **Animation** dialog box
- How to test animation
- How to use tag names, tag placeholders, and expressions when attaching animation
- How to set the minimum and maximum values for animation that uses a range of motion
- How to use **Object Smart Path** to define an object's range of motion
- How to set up each type of animation
- How to apply animation to groups
- How to check what kind of animation is attached to an object
- How to copy and paste animation
- How to set up animation for global objects

Types of animation

Animation associates graphic objects with tags so the appearance or position of an object changes to reflect changes to the tag's value. For example, an object's color could change from yellow to orange to red as the tag's value increases. Or a slider could move from left to right as a tag's value increases.

You can use these types of animation:

- Color
- Fill
- Height
- Horizontal position
- Horizontal slider
- Hypelrink
- Rotation
- Vertical position
- Vertical slider
- Visibility
- Width

Which objects can have which types of animation?

This table summarizes which types of objects support which types of animation. For information about the different types of graphic objects, see Types of graphic objects on page 274.

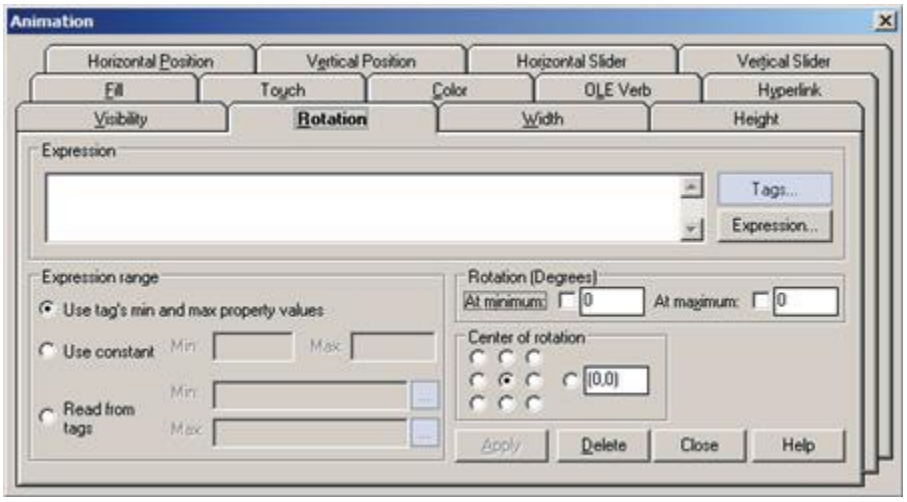
These objects	Support these types of animation
Drawing objects, except images, panels, and rounded rectangles	All types
Rounded rectangles	All types except rotation
All other objects	Visibility

You can also attach animation to groups of drawing objects. For more information, see Apply animation to groups on page 349.

You can attach as many types of animation to a drawing object (except images and panels) as you like. For example, apply width, height, horizontal position, and vertical position animation to an object to give it the appearance of moving into or out of the display as it shrinks and grows.

Use the Animation dialog box

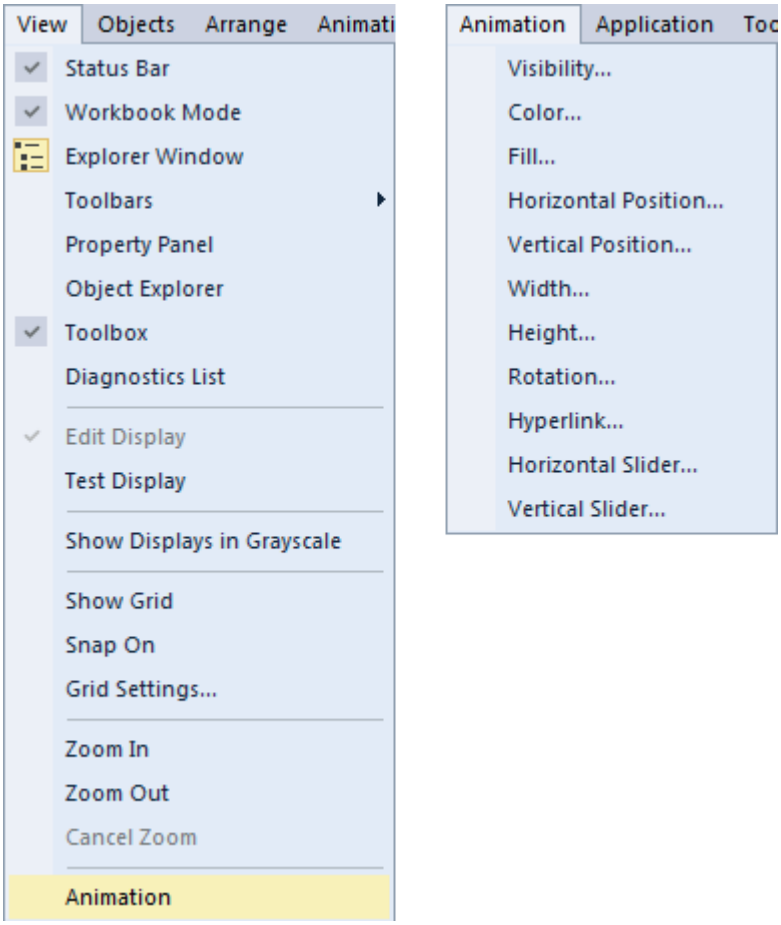
The **Animation** dialog box is a floating dialog box, which means you can keep it open all the time and move it around the screen. While it's open, you can select other objects and open other dialog boxes.



For details about the parts of the **Animation** dialog box, see Help.

To open the Animation dialog box, use one of these methods:

- Select an object, and then from the **View** menu, select **Animation**.
- Select an object, and then from the **Animation** menu, select an animation type. Animation types that are not supported for the selected object are unavailable.
- Right-click an object, select **Animation**, and then select an animation type. Animation types that are not supported for the selected object are unavailable.



Use Object Smart Path to visually set animation

Because the **Animation** dialog box stays open, you can go back and forth between the dialog box and the graphic display. This makes it easy to set the range of motion for an object because you do not have to know how many pixels you want an object to move. Instead, you can set the range of motion visually using the **Object Smart Path** feature. For details, see Use Object Smart Path on page 342.

Test animation

To test the animation you have set up on a graphic display, use **Test Display** to switch to test mode. When you finish testing, switch back to edit mode to continue editing.

To switch between test and edit modes, do one of the following:

- From the **View** menu, select **Test Display** or **Edit Display**
- On the toolbar, click ▶ or ■ .

Tip: Test mode is not the same as running the display. Test mode does not change the appearance or position of the display as set up in the **Display Settings** dialog box. Also, data logging is not turned on in test mode.

Use tag names and tag placeholders for animation

When setting up animation for objects, you are linking objects to tags. You can specify a tag name or use tag placeholder.

You can use HMI tags or existing data server tags, or you can use a new tag name.

Tag placeholders allow you to create displays that can be used with different tags.

You can use tag placeholders in:

- The graphic display that opens when the application is first run.
- Graphic displays that are opened using a Goto display button.
- Graphic displays that are opened using a display list selector.
- The graphic display that opens when the logout button is pressed.
- The graphic display that opens when auto logout is used.
- Graphic displays that are opened using the **Remote Display Number** global connection.

Use parameter files or parameter lists to specify which tags or folders to substitute for which placeholders. For global objects, you can specify the tags or folders of tags using global object parameters. For more information about using parameter files, parameter lists, and global object parameters, see Use parameters and global objects on page 374.

The tag placeholder can replace any part of a tag name, including folder names. For example, you could create a parameter file specifying that the tag placeholder #1=Folder1. You could assign the folder and a tag name to a graphic object's connection: #1\Tag1.

To create a tag placeholder

- In the **Expression** box, type the cross-hatch character followed by a number (no space in between). For example, #1.

Use expressions for animation

Many types of animation can be achieved using expressions. You can use expressions containing 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 it can stand alone as the entire expression.

For more information about expressions, see Use expressions on page 352.

Set minimum and maximum values

Many types of animation require a minimum and maximum range for the tag or expression. These values determine the start and endpoints for a range of motion.

For example, if you specify a minimum of 0 and a maximum of 100, the object will not react to values outside of this range. So, even if the expression has a value of 200, the object does not change from its **At maximum** position.

When setting up animation, select one of these methods for calculating the minimum and maximum values:

- **Use tag's min and max property values** — select this method to use the minimum and maximum values of the first HMI tag in the expression. If more than one HMI tag is used in the expression, the first HMI tag's minimum and maximum values are used.
  For analog HMI tags, the values are taken from the **Minimum** and **Maximum** boxes in the **Tags** editor. For digital tags, the minimum is 0 and the maximum is 1.
- **Use constant** — select this method to use numeric constants. Type the minimum and maximum values in the boxes.
- **Read from tags** — select this method to read two tags' values to determine the minimum and maximum values. Type the tag names in the boxes, or click the **Browse** buttons  to open the **Tag Browser** and select the tags.
  If you use this method, the tags are read when the graphic display opens. Their values at that time are used for the minimum and maximum values. The tags are not read again after this.

Define a range of motion

To define a range of motion for an object, use one of these methods:

- Use the mouse to move the object on the display. This uses the **Object Smart Path** feature to visually define the range of motion.
- Type values in the **At minimum** and **At maximum** boxes.

Motion can be defined in pixels, percentages, or degrees.

Animation that does not use a range of motion

Visibility and color animation do not use a range of motion, because these types of animation represent a change of state, not a range of values.

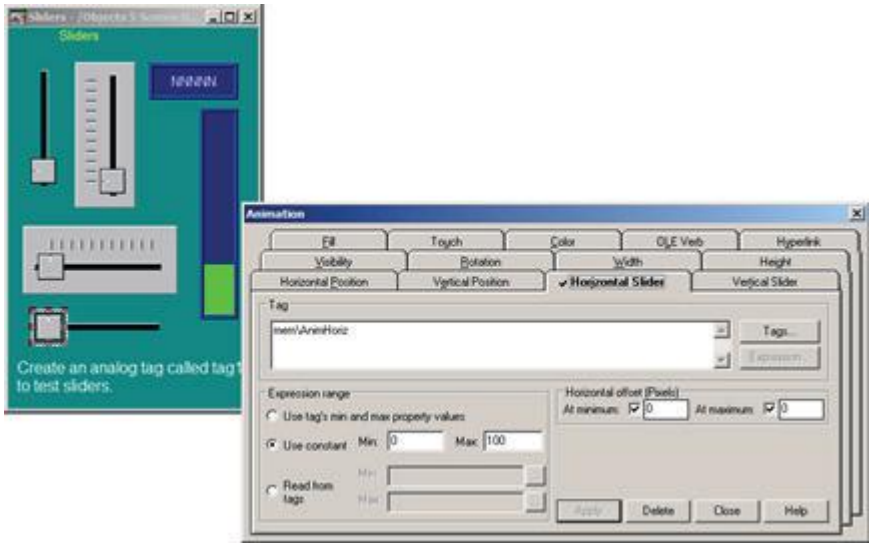
Use Object Smart Path

With **Object Smart Path**, you can easily set the range of motion for an object. The following example shows how **Object Smart Path** works.

Example: Use Object Smart Path to define the range of motion for horizontal slider animation

To define a range of motion for a slider object

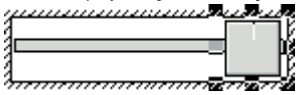
1. In the **Graphics** editor, create a slider object using a line and a rectangle, or copy a slider object from the Sliders graphic library.
2. Open the **Animation** dialog box and click the **Horizontal Slider** tab.
3. On the display, select the rectangle in the slider object.
4. In the **Tag** box of the **Animation** dialog box, specify a tag name.



5. On the display, drag the rectangle to the position that will indicate the lowest number in the range.

6. In the **Animation** dialog box, set this position by clicking the **At minimum** checkbox.

7. On the display, drag the rectangle to the position that will indicate the highest number in the range.



8. In the **Animation** dialog box, set this position by selecting the **At maximum** checkbox.

9. To save the settings, click **Apply**.

When you finish setting up the animation, the rectangle returns to its original position.

Set up the different types of animation

This section describes the different types of animation and provides tips and examples for setting up animation.

Set up visibility animation

With visibility animation, an object becomes visible or invisible based on a tag value or the result of an expression. When an object is invisible, it cannot be selected. Mouse clicks pass through it to whatever object is underneath.

Visibility animation is available for all objects and overrides an object's **Visible** property.

Tip: If you use a tag's value to control visibility animation as well as in an expression to control some other aspect of animation, when you set up visibility, set the Expression true state to **Invisible**. If you do not do this, the object could appear briefly in its design-time location and orientation before animating properly.

For more information about setting up visibility animation, see Help.

Example: Use visibility animation to set up security for a graphic display

This example shows how to use visibility animation to control what operators can see. On a graphic display that all users have access to, only the Admin user can see the graphic object to which this animation is attached.

This example uses the security function CurrentUserName( ). The function returns the string value of the Account ID (user name) for the user who is currently logged in.

The CurrentUserName( ) function is case sensitive. All user names use uppercase letters, so make sure that you use uppercase letters in your expression.

To specify which user can view an object in a display

1. Select the object to limit visibility to.
2. Open the **Animation** dialog box and click the **Visibility** tab.
3. In the **Expression** box, type this:

   ```
   CurrentUserName( ) == "ADMIN"
   ```
4. For the **Expression true state**, select **Visible**.
5. Click **Apply**, and then click **Close**.

At runtime, the object is visible only if the Admin user is logged in.

Set up color animation

With color animation, an object changes 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. Colors can be solid or blinking. For each color change, specify the value or threshold at which the color is to change and specify the colors to change to. At runtime, when the value reaches or crosses the threshold, the color changes.

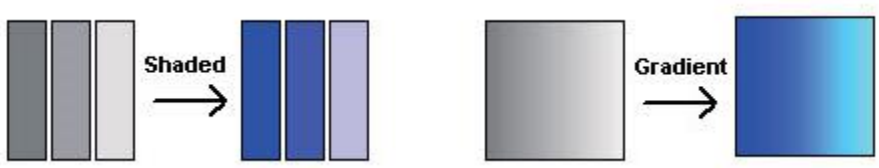
Color animation is available for all drawing objects only except image and panel objects, which only support visibility animation.

Use the **Fill Style** option to control the look of the fill color.

Use this fill style	To
Solid	Change the colors, and they will have a solid look to them.
Gradient	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	Prevent the colors of the object to be changed from what was originally imported.
Shaded	Use a tight dot pattern to soften or give a shaded appearance.

Use this fill style	To
	The Shaded fill style takes effect only on the group object that is composed of individual objects with dark to light colors, such as the Symbol Factory objects.

The following example shows the effects of Shaded and Gradient fill styles.



Keep 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.

Control 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 more information about setting up color animation, see Help.

Example: Create a text object that blinks

This example describes how to create a text object that constantly blinks between two colors. Since the blinking is not based on changes in tag values, the expression is simply a constant value that matches the value for the selected threshold.

For details about creating text, see Create text on page 283.

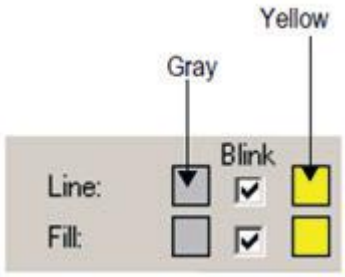
1. Select the text object.
2. Open the **Animation** dialog box, and then click the **Color** tab.
3. In the **Expression** box, type 0.
   Zero is the default value for threshold A.
4. In the list box, click threshold A. (Leave the value in the Value box as 0.)
5. For the line color, click **Blink**. (If desired, click Blink for the fill color too.)
6. For each color, click the color box, and then click the color to use.
7. Click **Apply**.

Example: Create an object that changes color as the fill level changes

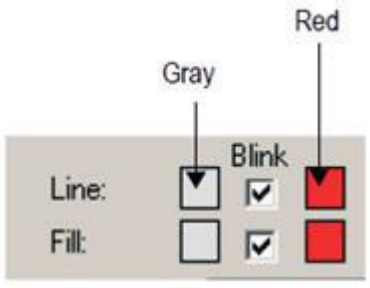
This example describes how to create a rectangle object that changes color as the object's fill level increases. This example uses a tag called Hopper1\FlourLevel. The tag has a range of 1 to 100. When the flour level reaches 80, the rectangle blinks between gray and yellow to warn the operator that the hopper is nearly full. When the flour level reaches 95, the rectangle blinks between gray and red.

You could use a bar graph object (without animation) to achieve a similar result.

1. Double-click the rectangle to open the **Polygon Properties** dialog box. Assign these properties to the rectangle:
   - In the **Back style** box, select **Solid**.
   - For the **Fore color** and **Back color**, select **gray**.
2. Click **OK** to close the **Polygon Properties** dialog box.
3. With the rectangle selected, open the **Animation** dialog box, and then click the **Fill** tab.
4. Attach fill animation as follows:
   - In the **Expression** box, type *Hopper1\FlourLevel* (this is the tag that monitors the fill level).
   - For **Fill Direction**, click **Up**.
5. Click the **Color** tab, and then attach color animation as follows:
   - In the **Expression** box, type *Hopper1\FlourLevel* (the same tag that was used in the **Fill** tab).
6. Set up the color for the normal state
   - In the **List** box, select **A**. (In the **Value** box, leave the value as 0.)
   - For **Fill Style**, select **Solid**.
   - For **Line** and **Fill**, select the color box, and then select gray (the same gray used for the rectangle).
7. Set up the color for the first warning
   - In the **List** box, select **B**.
   - In the **Value** box, type *80*.
   - For **Line** and **Fill** colors, select **Blink**. A palette box will be shown for the **Blink** color.



   - For the line colors, select gray for the first color and yellow for the blink color. Repeat for the fill colors.
8. Set up the **color** for the second warning
   - In the **List** box, select **C**.
   - In the **Value** box, type *95*.
   - For the **Line** and **Fill**, select **Blink**. A palette box will be shown for the **Blink** color.



   - For the line colors, select gray for the first color and red for the blink color. Repeat for the fill colors.
9. Click **Apply**.

Set up fill animation

Use the **Fill** option on the **Animation** menu to change the fill level in an object when an expression or tag value changes. The object's fill level is proportional to the minimum and maximum values specified for the expression. For example, if the value of the expression is halfway between its minimum and maximum values, the object will be half full.

You can attach fill animation to drawing objects only. The exceptions are the image and panel objects, which only support visibility animation.

If you select the **Inside Only** checkbox, fill animation does not affect objects' borders, line objects, or objects with transparent backgrounds.

For more information about setting up fill animation, see Help.

Set up horizontal position animation

Use the **Horizontal Position** option on the **Animation** menu to move an object horizontally as a tag value or the results of an expression change. The horizontal position of the object 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 horizontal position of the object will be halfway between its minimum and maximum horizontal offset positions.

You can attach horizontal position animation to drawing objects only. The exceptions are the image and panel objects, which only support visibility animation.

For more information about setting up horizontal position animation, see Help.

Set up vertical position animation

Use the **Vertical Position** option on the **Animation** menu to move an object vertically as a tag value or the results of an expression change. The vertical position of the object 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 vertical position of the object will be halfway between its minimum and maximum vertical offset positions.

You can attach vertical position animation to drawing objects only. The exceptions are the image and panel objects, which only support visibility animation.

To move an object on the display "up", a negative offset number is required. The top-left corner of the display has the coordinates of 0,0.

For more information about setting up vertical position animation, see Help.

Set up width animation

With width animation, an object's width changes based on a tag value or the result of an expression. The width of the object 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 width of the object will be halfway between its minimum and maximum horizontal change percents.

Width animation is available for all drawing objects except image and panel objects, which only support visibility animation.

For more information about setting up width animation, see Help.

Set up height animation

With height animation, an object's height changes based on a tag value or the result of an expression. The height of the object 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's height will be halfway between its minimum and maximum vertical change percents.

Height animation is available for all drawing objects except image and panel objects, which only support visibility animation.

For more information about setting up height animation, see Help.

Set up rotation animation

With rotation animation, an object rotates around an anchor point based on a tag value or the result of an expression, with the angle of rotation of the object being 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 rotate half the specified amount.

Rotation animation is available for all drawing objects except images, panels, and rounded rectangles. If you apply rotation animation to text, the text rotates around the anchor point but remains in the upright position.

For more information about setting up rotation animation, see Help.

Set up hyperlink animation

With hyperlink animation, an object links to a destination. When the object is clicked at runtime, in Test application mode, or in Test Display mode, the linked destination opens. The destination can be a file, a web page, or a program.

Hyperlink animation is available for all drawing objects except image and panel objects which only support visibility animation.

For more information about setting up hyperlink animation, see Help.

Set up horizontal slider animation

Use the **Horizontal Slider** option on the **Animation** menu to create a graphic object that can control the value of a specified tag. With horizontal slider animation, you define a path for the object, then use the mouse to move the object horizontally along its path. The pixel position of the object is translated into a value that is written to the tag.

A tag that has values input by slider animation can be used in an expression to attach animation to another object or group of objects. If the tag value is changed externally, the position of the slider will change as well.

Both vertical and horizontal slider animation can be attached to the same object.

You can attach horizontal slider animation to drawing objects only. The exceptions are the image and panel objects, which only support visibility animation.

For more information about setting up horizontal slider animation, see Help.

Tip: The Sliders graphic library contains several slider objects that you can drag and drop into your graphic displays. Attach slider animation to the button portion of the slider object.

If you create your own slider object, it's useful to draw an object (for example, a line) to represent the path the slider will move along.

Set up vertical slider animation

Use the **Vertical Slider** option on the **Animation** menu to create a graphic object that can control the value of a specified tag. With vertical slider animation, you define a path for the object, then use the mouse to move the object vertically along its path. The pixel position of the object is translated into values that are written to the tag.

A tag that has values input by slider animation can be used in an expression to attach animation to another object or group of objects. If the tag value is changed externally, the position of the slider will change as well.

Both vertical and horizontal slider animation can be attached to the same object.

You can attach vertical slider animation to drawing objects only. The exceptions are the image and panel objects, which only support visibility animation.

For more information about setting up vertical slider animation, see Help.

Apply animation to groups

You can apply animation to objects and then group those objects and apply animation to the group. When the display is running, animation is applied as follows:

These types of animation	Are applied like this
Color	Animation attached to individual objects within the group overrides group animation.
Fill	Animation results for both the individual objects and the group are applied.
Horizontal slider, hyperlink, vertical slider	Group animation overrides animation attached to individual objects within the group.
Height, width, horizontal position, vertical position, and rotation	Animation results for individual objects and the group are combined. For example, if an individual's horizontal position animation result is to offset the object by 100 pixels, and the group's result is to offset the group by 200 pixels, the individual object is offset by 300 pixels.
Visibility	When the group's animation visibility is **False** (the group is not visible), then no objects in the group are visible, regardless of the animation status of the individual objects. When the group's animation visibility is **True** (the group is visible), the visibility of an object within the group is determined by the individual object animation.

Test your animation to ensure you achieve the intended results.

To apply animation to objects within groups, use the group edit feature. For details, see Edit groups of objects on page 313.

Check the animation on objects

To see what type of animation has been set up for an object or group of objects, use the **Object Explorer**, the **Animation** menu, or the **Animation** dialog box.

For information about using the **Object Explorer** to highlight objects that have animation attached, see Highlight objects in the Object Explorer on page 294.

To see what type of animation has been set up for objects within a group, use the group edit feature or the **Object Explorer**. For information about using the group edit feature, see Edit groups of objects on page 313.

View the animation on an object using the Animation menu

To view the animation on an object using the Animation menu

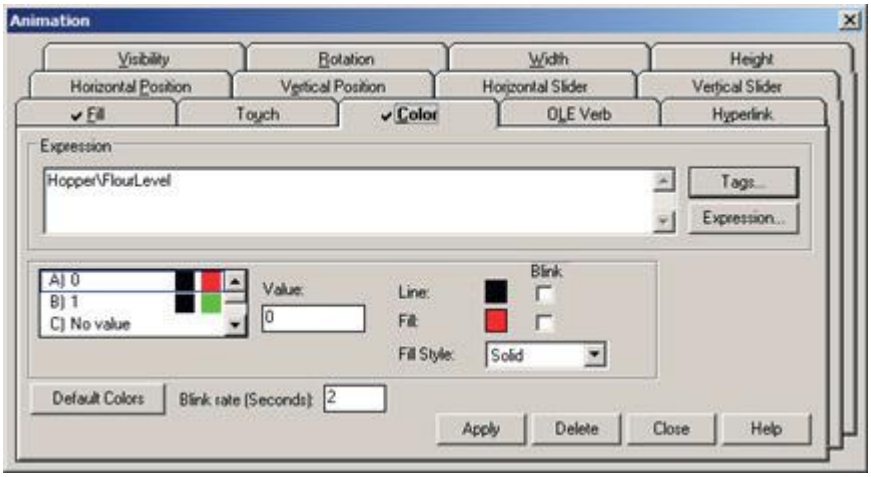
1. Select an object.
2. View the animation by doing one of the following:
   - Click the **Animation** menu and see which items have a check mark.
   - Right-click the object and then select **Animation** to see which items have a check mark.

View the animation on an object using the Animation dialog box

To view the animation on an object using the Animation dialog box

1. Select an object.
2. From the **View** menu, select **Animation**.

   When the **Animation** dialog box opens, look at which tabs have check marks on them to see which types of animation have been set up.



Copy or duplicate objects with animation

You can copy or duplicate objects that have animation attached to them. When you do, the animation attached to the objects is also copied or duplicated. If you copy or duplicate a group, the copy of the group can be ungrouped to individual objects, just like the original.

For information about copying and duplicating objects, see pages Copy objects on page 307 and Duplicate objects on page 308.

Copy animation without copying objects

If you have attached animation to an object, you can copy the animation and paste it onto another object. If the object has more than one type of animation, all animation is copied and pasted. You can only copy animation to an object that supports the same type of animation.

To copy and paste animation

1. Select the object that has the animation you want to copy.
2. From the **Edit** menu, select **Copy Animation**, or right-click the object and then select **Copy Animation**.
3. Select the objects to copy the animation to.
4. From the **Edit** menu, select **Paste Animation**.
   To paste to a single object, you can right-click the object and then select **Paste Animation.**

Set up animation for global objects

FactoryTalk View global objects allow you to link the appearance and behavior of a graphic object to multiple copies of that object. When the original base object is updated, the changes are automatically applied to all the copies of the object. For information about creating base objects, see Use global objects on page 379.

The copies of the object are called reference objects, and they have the same properties as the original base object. If desired, you can assign unique animation to the reference object. For information about creating reference objects, see Create reference objects on page 381.

The **LinkAnimation** property determines whether the reference object uses the base object's animation.

To set up animation for a reference object

1. Double-click the reference object to open the **Property Panel**.
2. Select the **LinkAnimation** property setting to use:
   - **Do not link** — allows you to set up separate animation for the reference object.
   - **Link with expressions** — the reference object uses the animation and tags or expressions assigned to the base object.
   - **Link without expressions** — allows you to use only the types of animation set up for the base object, but assign different tags or expressions to the reference object.
3. If you selected **Do not link** or **Link without** expressions, set up animation for the object using the methods described in this chapter.

For more information about setting up link properties and working with reference objects, see Set up reference objects' link properties on page 382.

Use expressions

This chapter describes:

- The types of expression components
- How to use the **Expression** editor
- Which editors use expressions
- How to format expressions
- How to use tag names and tag placeholders with expressions
- How to use constants
- How to use operators
- How to use math, security, and language functions
- How to use if-then-else logic in expressions
- The evaluation order of operators
- How to use write expressions

About expressions

Sometimes the data you gather from devices is meaningful only when you:

- Compare it to other values.
- Combine it with other values.
- Create a cause-effect relationship with other values.

Expressions allow you to create mathematical or logical combinations of data that return more meaningful values.

An expression can be a tag name, a tag placeholder, a mathematical or logical equation, or a function that returns a numeric, string, or true or false value. For expressions that return true and false values, 1 and other non-zero values signify True, and zero signifies False. Expressions let you manipulate the data gathered from devices and make it more meaningful by comparing it to other values, combining it with other values, creating cause and effect relationships between values, or displaying it.

Expressions can contain:

- Tags, tag placeholders, arithmetic, and bitwise operators, and mathematical functions such as SQRT (square root), that return numeric values.
- Relational and logical operators, and functions such as ALM_FAULT (alarm fault), that return true/false values.
- Non-mathematical functions, such as the CurrentLanguage function that returns the RFC17766 code for the current language of an application.
- If-Then-Else logic that returns numeric or true and false values, depending on how it's structured. These are called conditional expressions because the result of the expression depends on whether the If statement is true or false. When the If statement evaluates to True the result is defined by the Then statement. When the If statement is False the result is defined by the Else statement.

Use expressions in:

- Graphic displays
- Alarm setup
- Information setup

- Macros
- Global connections

Expressions that result in floating-point values

If an expression results in a floating-point value but an integer value is required, the floating-point value is rounded.

For information about how values are rounded, see How values are rounded on page 80.

Use the Expression editor

To create an expression, use one of these methods:

- Type it directly in the **Tag or expression** column, for any connection that accepts expressions, or in the **Expression** box (for animation).
- Open the **Expression** editor, and then create the expression in the editor.

Use the Expression editor versus typing expressions directly

Once you are familiar with expression syntax, you might find it quicker to create short expressions by typing them directly in the **Tag or expression** column.

The **Expression** editor allows you to see more text at once, which is useful for longer, more complicated expressions. Also, you can click buttons to enter tag names, operators, and functions, thus avoiding typing mistakes. Another advantage of using the **Expression** editor is that you can check whether the syntax of the expression you have created is valid.

Create an expression by typing it directly

To create an expression by typing it directly

- Type an expression up to 16,000 characters long.

Expressions that you type directly are not checked for syntax.

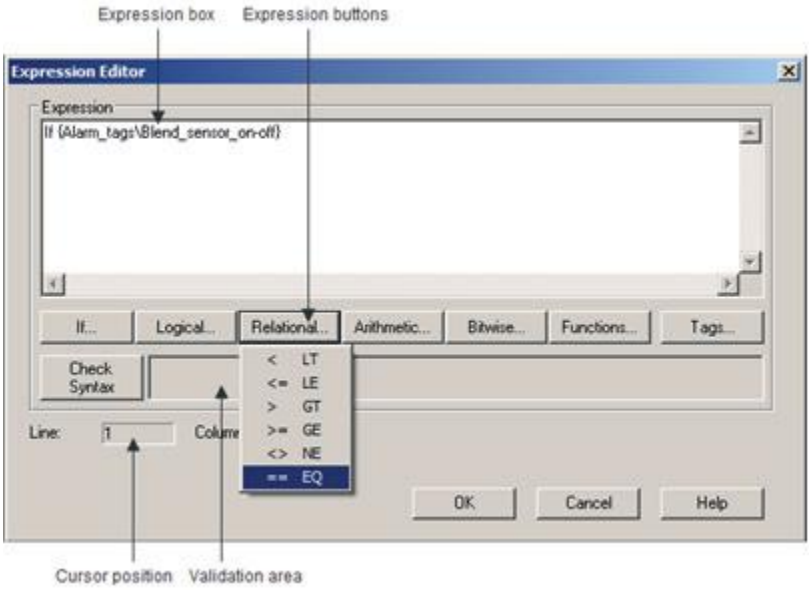
Open the Expression editor

To open the Expression editor, use one of these methods:

- Click the **Browse** button in the **Exprn** column 

for a connection that accepts expressions.
The **Browse** button is not available for connections to which you can assign only tags.
- In the **Animation** dialog box, click the **Expression** button.

The **Expression** editor has these parts:



For details about using the options in the **Expression** editor, see Help.

Where you can use expressions

You can use expressions in these editors:

- **Graphics** — You can define an expression to control various aspects of a graphic object's appearance. For more information about assigning expressions to graphic objects, see Use expressions to manipulate tag values on page 302.

  You can also use expressions to attach animation to graphic objects. For more information, see Animate graphic objects on page 338.
- **Global Connections** — You can use expressions to remotely control when to open and print displays, as well as the date and time to show. For more information, see Set up global connections on page 89.
- **Alarm Setup** — When setting up alarms, you can use expressions for alarm triggers, and with some of the connections that silence and acknowledge alarms. For more information, see Set up alarms on page 95.
- **Information Setup** — You can use expressions to determine when to show information messages. For more information, see Information messages on page 392.
- **Macros** — You can use expressions in macros to assign values to tags. For more information, see Use macros to assign values to tags on page 421.

Format expressions

You can format expressions so they are easier to read. However, do not let tag names, function names, or function arguments span more than one line.

When formatting expressions, you can use line returns and multiple spaces.

Enclose strings in quotes. The string can contain any character, and can include spaces.

Example: Format 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 it to the following:

if (tag1 > tag2) then 0 else if (tag1 > tag3) then 2 else 4

Use tag names and tag placeholders

A tag name can be included as part of an expression or can stand alone as the entire expression.

To supply a tag name, use one of these methods

- Type a tag name.
  You can type a tag name that does not exist in the tag database. When you click **OK**, you are prompted to create the tag. You can create it now, or write down the name and create it later.
- Click the **Tags** button and select a tag from the **Tag Browser**.

Enclose tag names that contain dashes or start with a number in braces { } when you use them in an expression. This distinguishes the characters in the tag name from the characters in the expression.

You can use string tags as operands with the plus (+) arithmetic operator and with the relational operators.

Use tag placeholders instead of tag names

The **Graphics** editor accepts tag placeholders instead of tag names. Placeholders allow you to use the same display with different sets of tags.

You can use tag placeholders in:

- The graphic display that opens when the application is first run.
- Graphic displays that are opened using a Goto display button.
- Graphic displays that are opened using a display list selector.
- The graphic display that is opened by a logout button.
- The graphic that is opened when auto logout is used.
- Graphic displays that are opened using the **Remote Display Number** global connection.

Use parameter files to specify which tags or folders to substitute for which placeholders. For global objects, you can specify the tags or folders of tags using global object parameters. For more information about using parameter files and global object parameters, see Use parameters and global objects on page 374.

The tag placeholder can replace any part of a tag name, including folder names. For example, you could create a parameter file specifying that the tag placeholder #1=Folder1. You could assign the folder and a tag name to a graphic object's connection: #1\Tag1.

To create a tag placeholder in an expression

- Type the cross-hatch character followed by a number (no space in between). For example, #1.

Constants

A constant can have any of the following formats:

- Integer (123)
- Floating-point (123.45)
- String constant ("character string")

Arithmetic operators

Arithmetic operators perform math on two or more numeric values and calculate the result. The arithmetic operators are:

Symbol	Operator	Example (For these examples, tag1 = 5 and tag2 = 7)
+	addition	tag1 + tag2 returns a value of 12 You can also use this operator with string operands. See String operands on page 356.
-	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. In the example, the remainder of 7 divided by 5 is 2; so 7 % 5 = 2 **Important**: This operator is for integers only, not floating-point numbers.
**	exponent	tag1 ** tag2 returns a value of 78125

Be sure that any tag value you use as a divisor cannot at some point have a value of zero. Expressions that attempt to divide a number by zero produce an error at runtime.

String operands

The + operator can be used to join string operands. For example, the expression "hello" + "world" returns: helloworld.

You cannot join string tags to analog tags, whether they are HMI or data server tags.

Relational operators

Relational operators compare two numeric or string values to provide a true or false result. If the statement is true, a value of 1 is returned. If false, 0 is returned.

The relational operators are:

Symbols	Operator	Numeric Example	String Example
For the numeric examples, tag1 = 5 and tag2 = 7 For the string examples, serial_no = "ST009"			
EQ, ==	equal	tag1 == tag2 false	serial_no == "ST009" true
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 letters 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. The operators return a nonzero value if the expression is true, or a 0 if the expression is false.

Any statement that evaluates to a nonzero value is regarded as true. For example, the statement tag1 is false if the value of tag1 is 0, and true if tag1 has any other value.

The logical operators are:

Symbols	Operator	Action	Example (For these examples, tag1 = 5 and tag2 = 7)
AND, &&	and	Returns a 1 if the statements to the right and left of the operator are both true.	(tag1 < tag2) AND (tag1 == 5) Both statements are true; returns a 1.
OR, ||	or	Returns a 1 if either the statement to the left or right of the operator is true.	(tag1 > tag2) OR (tag1 == 5) tag1 == 5 is true; returns a 1.
NOT	negation	Reverses the logical value of the statement that 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 previous expressions. They determine the evaluation order of the operators. For more information, see Evaluation order of operators on page 360.

Bitwise operators

Bitwise operators examine and manipulate individual bits within a value.

IMPORTANT: These operators are for integers only, not floating-point numbers. Do not use them with tags or expressions that return floating-point values.

Symbol	Operator	Action
&	And	Compares two integers or tags on a bit-by-bit basis. Returns an integer with a bit set to 1 if both the corresponding bits in the original numbers are 1. Otherwise, the resulting bit is 0.
|	inclusive OR	Compares two integers or tags on a bit-by-bit basis. Returns an integer with a bit set to 1 if either or both of the corresponding bits in the original numbers are 1. If both bits are 0, the resulting bit is 0.
^	exclusive OR (XOR)	Compares two integers or tags on a bit-by-bit basis. Returns an integer with a bit set to 1 if the corresponding bits in the original numbers differ. 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 in the right operand. The bit on the right disappears.

Symbol	Operator	Action
		Either a 0 or a 1 is shifted in on the left, depending on whether the leftmost bit is a 0 or a 1, and whether the operand consists of a signed or unsigned data type. For signed data types, if the leftmost bit is 0, a 0 is shifted in. If the leftmost bit is 1, a 1 is shifted in. In other words, the sign of the number is preserved. For unsigned data types, a 0 is always shifted in.
<<	left shift	Shifts the bits within an integer or tag to the left. Shifts the bits within the left operand by the amount specified in the right operand. The bit on the left disappears and 0 always shifts in on the right. See Use the left shift operator on page 359.
~	complement	Returns one's complement; that is, it toggles the bits within an integer or tag. Reverses every bit within the number so every 1 bit becomes a 0 and vice versa.

Use the left shift operator

If the left bit is a 1 an overflow occurs, and an error message is generated. To prevent this, use the bitwise AND operator with the left shift operator in an expression. For example:

```
(dev << 1) & 65535
```

where dev is a tag whose value is being shifted left, and 65535 is 1111 1111 1111 1111 in binary form.

Examples: 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

Expressions with more than one operator are evaluated in this order:

- Operators in parentheses are evaluated first.
1. Therefore, to change the order of precedence, use parentheses.
- The operator with the highest precedence is evaluated next.
- When two operators have equal precedence, they are evaluated from left to right.

Operators are evaluated in this order:

Evaluation order	Symbols
1 (highest)	( )
2	NOT ~
3	* / MOD, % ** AND, && & >> <<
4	+ - OR, || | ^
5 (lowest)	EQ, == NE, <> LT, < GT, > LE, <= GE, >=

Examples: Evaluation order

For these 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).

Mathematical functions

Use math functions to calculate the square root, log (natural or base 10), or trigonometry ratios (in radians or degrees) of a tag.

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

Security functions

Use security functions to control access to your application. These functions allow you to determine a user's identity or security rights to limit access to the application based on these criteria.

This function	Returns this value
CurrentUserName( )	A string containing the name of the current user. This function is case sensitive. All RSView 3.20 and earlier user names use uppercase letters.
CurrentUserHasCode(*SecurityCode*)	1 (true) if any of the specified security codes have been assigned to the user; 0 (false) if not. The argument can be a security code, tag name, or tag placeholder. If you use a tag name or tag placeholder, it must be enclosed in curly brackets ({}), for example, CurrentUserHasCode({#1}). If checking multiple security codes, do not type a space between the security code letters. For example, CurrentUserHasCode(ABP) returns the value 1 if the user has been assigned one or more of the specified codes.
CurrentUserHasGroup(*"Group Name"*)	1 (true) if the current user is assigned to the specified group configured in FactoryTalk Diagnostics; 0 (false) if not. The argument can be a literal string, string tag, or tag placeholder. If you use a string, it must be enclosed in double quotes, for example, CurrentUserHasGroup ("*UserGroupName1*"). The argument can also be a combination of a literal string and string tag in the format of *"Liter string" + String tag*, for example, CurrentUserHasGroup ("*UserGroupName1*"+{*[Controller]Main.UserGroupName*}). If CurrentUserHasGroup("Group C") returns the value 1, the user has been assigned to Group C. This function is only supported in 8.0 and later versions of FactoryTalk ME applications.

If you use a tag parameter for the CurrentUserHasCode( ) or the CurrentUserHasGroup ( ) security function and create an 8.20 or earlier version runtime application (.mer) using current version FactoryTalk View:

- The runtime application functions well.
- When you restore the runtime application to 8.20 or earlier version design time application (.med) and create runtime application from the design time application in FactoryTalk View 8.20 or earlier, the security function does not work.

For more information about setting up security for your application, see Set up security on page 143.

For an example of using the CurrentUserHasCode(*x*) function, see Example: Assign visibility animation to the shutdown button on page 154.

For examples of using the CurrentUserName( ) function, see Example: Assign visibility animation to the Goto display button on page 154.

Language function

The language function shows you which language your application is using.

You can show the current language in a string display, or use it in expressions to generate language-specific messages for your users.

This function	Returns this value
CurrentLanguage( )	RFC1766 name of the current runtime language.

The RFC1766 name is a standard way of representing a language using the format: *languagecode-Country/RegionCode*

where *languagecode* is a lowercase two-letter code and *Country/RegionCode* is an uppercase two-letter code.

For example, U.S. English is en-US.

For more information about setting up languages for your application, see Set up language switching on page 159. For a list of RFC1766 names, see RFC1766 names on page 465.

Language switching alarm, information, and local messages in FactoryTalk View ME Station 4.00

FactoryTalk View 5.00 and later provides direct support for switching message languages at runtime.

For applications that will run in FactoryTalk View ME Station version 4.00, use the CurrentLanguage( ) expression function to specify message offsets in the information message file. In the file, divide your messages into sections for each language. To do this, set up trigger value offsets for a series of messages in the **Alarm Setup** editor, information message files, and local message files.

Export the text strings in the **Alarm Setup** editor and message files for translation. Then paste the translated strings into the editor and original message files, and assign to each string the correct trigger value. For information about exporting text for translation, see Set up language switching on page 159.

Example: Set up information messages in multiple languages

This example shows how to generate English, French, or German information messages at runtime in a FactoryTalk View ME Station 4.00 application, depending on which language the application is using.

1. Create a tag called *Information_messages* that will generate trigger values of 11 to 20 for different conditions that require information messages.
2. Create an information message file.
3. Create English messages for trigger values 11 to 20.

   Tip: Messages are sorted alphanumerically in the Excel spreadsheet or Unicode text file created for translation. Therefore, numbers 2 through 9 would appear after 10, 11, 12, and so on. To keep your messages in order in the translation file, begin the first series of numbers at 11.

4. Create French messages for trigger values 21 to 30.

5. Create German messages for trigger values 31 to 40.
6. Assign this expression to the **Value** connection in the **Information Setup** editor:

```
If CurrentLanguage( )="en-US" then Information_messages
Else If CurrentLanguage( )="fr-FR" then Information_messages + 10
Else Information_messages + 20
```

If-then-else

If-then-else expressions carry out an action conditionally or branch actions depending on the statements in the expression. The if-then-else statements enable the expression to perform different actions in different situations and to repeat activities until a condition changes.

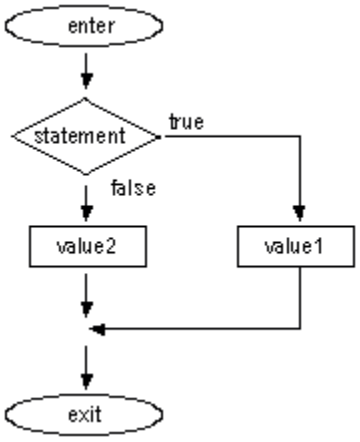
To build conditional expressions, use the relational operators and the logical operators for the statement and values.

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. If the result of the statement is a nonzero value, the statement is true (and returns value1); if the result is 0, the statement is false (and returns value2).

The if-then-else structure is illustrated here.



Nested if-then-else

You can nest an if-then-else structure inside the Then or Else part of an if-then-else structure.

Example 1: Nested if-then-else

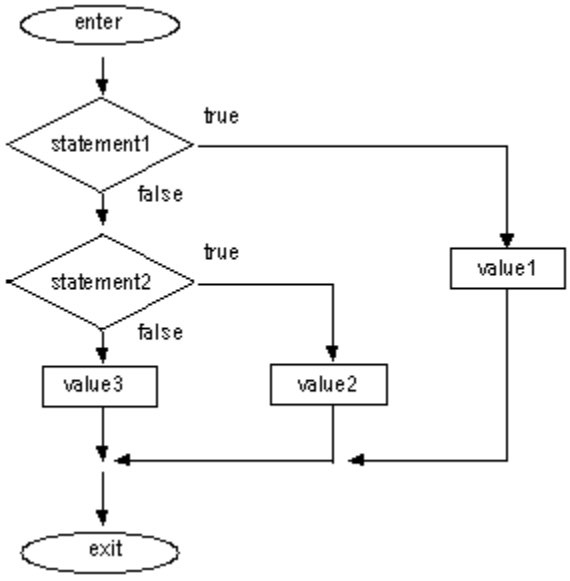
This expression:

if *statement1* then *value1*

else if *statement2* then *value2*

else *value3*

has this interpretation:



Example 2: Nested if-then-else

This expression:

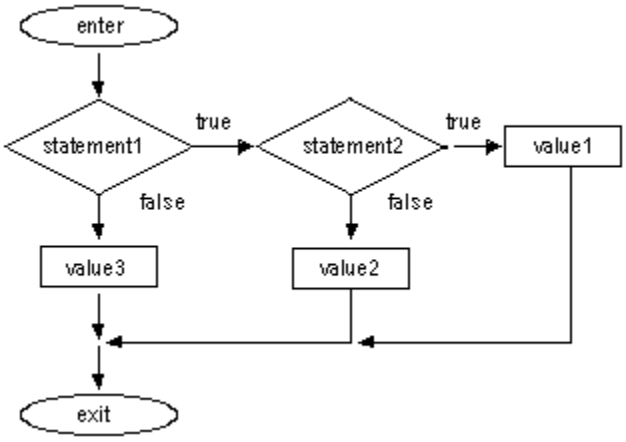
if *statement1* then

if *statement2* then value1

else *value2*

else *value3*

has this interpretation:



Use write expressions

Write expressions allow the operator to enter a value that is manipulated by an expression before being sent to the data source. FactoryTalk View substitutes the value the operator enters for the placeholder in the expression,

calculates the value of the expression, and writes the result to the **Value** connection. All write expressions must contain a question mark (?) as a placeholder for the value the operator enters.

You can use write expressions with the Numeric Input Enable button and the numeric input cursor point. When the operator presses the button or cursor point, a keypad or scratchpad opens. The operator enters a value in the keypad or scratchpad, and this value is substituted for the ? placeholder in the write expression.

Example: Use write expressions

In this example, the operator regulates the speed of a conveyor belt by entering a value in feet or meters per second. When the operator enters the value in meters per second, the value is converted to feet per second before being passed to the data source.

The operator first indicates whether the value is in feet or meters by pushing a Maintained push button. The push button has one state corresponding to feet per second, and the other state to meters per second.

Then the operator presses the Numeric Input Enable button and enters the value for the conveyor speed in a numeric pop-up keypad. The ? character in the write expression is the placeholder for the value that the operator enters.

Set up the Maintained push button

To set up the Maintained push button

1. In the **Maintained Push Button Properties** dialog box, on the **States** tab, set up these states:
   - **State 0** — Value: 0, Caption: Feet/S
   - **State 1** — Value: 1, Caption: Meters/S
2. On the **Connections** tab, assign a digital tag called Feet_or_meters to the **Value** connection (either an HMI tag or a data server tag).

Set up the Numeric Input Enable button

To set up the Numeric Input Enable button

1. On the **Label** tab of the **Numeric Input Enable Properties** dialog box, type the caption *Enter conveyor speed*.
2. On the **Connections** tab, assign a tag called *Conveyor_speed* to the **Value** connection.
3. Assign this expression to the **Optional Exp** connection:

   ```
   if Feet_or_meters == 0 then
   ?
   else
   ? * 3.281
   ```

FactoryTalk View writes the result of the expression to the Conveyor_speed tag at the data source.

Use embedded variables

This chapter describes:

- The types of embedded variables
- Where you can create embedded variables
- How to create embedded variables
- Embedded variable syntax
- How embedded variables are updated at runtime
- How embedded variables are shown at runtime

About embedded variables

Embedded variables allow you to show values that change dynamically at runtime. Put placeholders in strings where the embedded variable will be shown. At runtime, the placeholder is updated with the real-time values of the variables.

Embedded variables are of two kinds:

- A tag is read to provide the runtime value. Use tags, to embed tag values (Numeric or String), date values, and time values. The tag is read at runtime and the current value is shown.
- A tag is not read. Use this type to embed literal strings and numbers that will be shown at runtime as they were entered.

Use one or more embedded variables in:

- The text captions on graphic objects
- The title bar of On Top displays
- Message text.

For example, you could embed a tag value and the time variable in a local message. At runtime when the local message is shown, it is updated to reflect the tag's current value as the value changes. The time is also updated as the time changes.

You can also use literal strings and constants in embedded variables, or a combination of both variable and literal strings and numbers.

Embedded variables can consist of:

- Numeric (analog or digital) tags, including both HMI and data server tags
- Literal numbers (constants)
- String tags, including both HMI and data server tags
- Literal strings (static text)
- Tag placeholders.
- The time
- The date

For information about tag placeholders, see Use tag placeholders on page 374.

Where you can create embedded variables

You can create embedded variables in these editors:

- *Graphics*. Use this editor to insert embedded variables in the captions for graphic objects or in any text objects. For graphic objects with multiple states, you can insert different embedded variables in each state's caption.
- *Local Messages*. Use this editor to insert embedded variables in local messages.
- *Information Messages*. Use this editor to insert embedded variables in information messages.
- *Alarm Setup*. Use this editor to insert embedded variables in alarm messages.
- *Display Settings*. Use this editor to insert embedded variables in the title bars of displays.

For more information about	See
Graphic objects	Set up graphic objects on page 319
On Top displays	On Top displays on page 259
Local messages	Local messages on page 269
Information messages	Information messages on page 392
Alarms	Set up alarms on page 95

Create an embedded variable in a graphic object's caption

To create an embedded variable in a graphic object's caption

1. Open the graphic object's **Properties** dialog box.
2. Click the tab containing the **Caption** box.

   The **Caption** box is on the **Label** tab or the **States** tab, depending on the type of object.

   For text objects, use the **Text** box on the **General** tab.
3. Click **Insert Variable**.
4. Click the type of variable to insert.
5. Fill in the options in the dialog box that opens. For details about the options, see Help.

Create an embedded variable in an On Top display's title bar

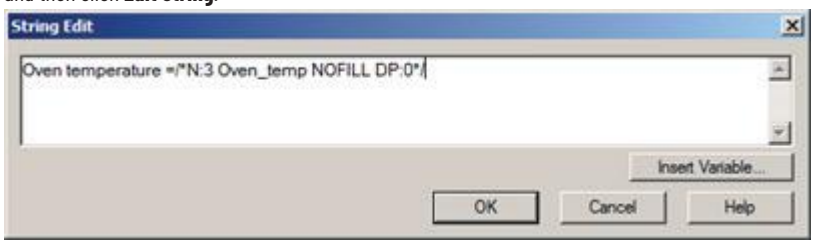
To create an embedded variable in an On Top display's title bar

1. Open the **Display Settings** dialog box.
2. Select the **Title Bar** checkbox if it is not already selected.
3. Click **Insert Variable**.
4. Click the type of variable to insert.
5. Fill in the options in the dialog box that opens. For details about the options, see Help.

Create an embedded variable in a message

To create an embedded variable in a message

1. In the **Message** column of the **Local Messages**, **Information Messages**, or **Alarm Setup** editor, right click and then click **Edit String**.



2. Click **Insert Variable**.
3. Click the type of variable to insert.
4. Fill in the options in the dialog box that opens. For details about the options, see Help.

Embedded variable syntax

Embedded variables are case-sensitive, and must use specific syntax to work. Otherwise, the embedded variable is treated as a piece of text. Therefore, we do not recommend creating and editing embedded variables manually. Instead, use the **Insert Variable** and **Edit Variable** dialog boxes.

Numeric embedded variable syntax

Use numeric embedded variables to insert analog or digital tag values into captions, title bars, and messages. You can use both HMI and data server tags.

You can also insert "literal" numbers to show a constant or to specify a tag placeholder in the caption or message.

Numeric embedded variables use this syntax:

/*LN:# *Tag_name Fill_character* DP:#*/

where

- *L* (optional) indicates it is a literal (constant) number. This symbol prevents a tag read. If you type a tag placeholder for the *Tag_name*, the value of the placeholder is substituted from the parameter file or global object parameter definition.
- *N* indicates it's a numeric embedded variable.
- *#* indicates the number of digits.
- *Tag_name* is the tag to show; you can also type a literal number or tag placeholder here.
- *Fill_character* is the fill character to use: NOFILL, ZEROFILL, or SPACEFILL.
- *#* indicates the number of decimal places.

Examples: Numeric embedded variable syntax

To show the current value of a tag called Oven_temp, with 3 digits, no decimal places, and no fill character, type this:

/*N:3 Oven_temp NOFILL DP:0*/

To show the constant 48, with 3 decimal places and 2 zeroes to the left of the number (for a total length of 8 digits including the decimal), type this:

```
/*LN:8 48 ZEROFILL DP:3*/
```

At runtime the numeric embedded variable would look like this: 0048.000.

String embedded variable syntax

Use string embedded variables to insert string tag values into captions, title bars, and messages.

You can also insert "literal" strings of static text. For example, you can type a word or phrase, a tag placeholder, or a number. To control how constant numbers are shown, use a literal numeric variable rather than a string variable.

String embedded variables use this syntax:

```
/*LS:-# Tag_name SHOWSTAR*/
```

where

- *L* (optional) indicates it is a literal (static) string. This symbol prevents a tag read. If you type a tag placeholder for the *Tag_name*, the value of the placeholder is substituted from the parameter file or global object parameter definition. The value cannot contain spaces.
- *S* indicates it's a string embedded variable.
- - (optional). A minus sign (-) before the # indicates that if the string is longer than the fixed number of characters, the right-most characters will be shown.
- # indicates the number of characters if you select a fixed number of characters; type 0 if you don't want to use a fixed number.
- *Tag_name* is the tag to show; you can also type a literal string or tag placeholder here. The string cannot contain spaces.
- *SHOWSTAR* (optional) specifies that if the string is longer than the fixed number of characters, an asterisk (*) will be shown to indicate the string is truncated. If you use the minus sign (-), the asterisk will be shown as the first character at the left end of the string. Otherwise, the asterisk will be shown as the last character at the right end of the string.

Examples: String embedded variable syntax

To show the current value of a string tag called Blower_status, with a fixed length of 20 characters, you would type this:

```
/*S:20 Blower_status*/
```

To show the literal string Oven temperature, type this:

```
/*LS:20 "Oven temperature"*/
```

To show the literal string 36.5, type this:

```
/*LS:3 36.5*/
```

To assign the tag placeholder #1, without a fixed string length, type this:

/*S:0 #1*/

To show the value of the string tag Conveyor_message, with a fixed length of 40, showing the right-most characters, with an asterisk to indicate if the message is truncated, type this:

/*S:-40 Conveyor_message SHOWSTAR*/

Time and date embedded variable syntax

Use time and date embedded variables to insert the current time or date into captions, title bars, and messages

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 syntax

To show the time followed by the short date, you would type this:

/*TSD*/

A space is placed between the time and date when the embedded variable is shown at runtime.

How embedded variables are updated at runtime

At runtime, this is how embedded variables are shown and updated:

- **Graphic objects and title bars** — When a display containing a graphic object or title bar that uses an embedded variable is open, the value of the embedded variable is updated whenever a new tag value is read from the data source. For time and date embedded variables, the time and date are updated as the system time and date change.
- **Local messages** — When a display containing a local message display object is open, and the message the object is showing contains an embedded variable, the value of the embedded variable is updated whenever a new tag value is read from the data source. For time and date embedded variables, the time and date are updated as the system time and date change.

- **Information messages** — The value of the embedded variable is read when the information message is first shown. It is not updated after that.
  If the message is printed, it is printed using the value the variable had when the message was first shown. This value is retained if you shut down and restart the application.
- **Alarm messages** — The value of the embedded variable is read when the alarm occurs, and is shown in the message associated with the alarm. It is not updated after that.
  If the message is printed, it is printed using the value the variable had when the alarm first occurred. This value is retained if you shut down and restart the application.

How embedded variables are shown at runtime

If there is no valid data available for the embedded variable, the variable is replaced with question marks (?). This could 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.

If a string or numeric embedded variable has been set up but no tag has been assigned, the embedded variable is replaced with asterisks (*).

Numeric embedded variables

The value shown for a numeric embedded variable depends on whether the tag value is a floating-point number or an integer. Integer values are shown as is. Floating-point values are rounded to fit the specified number of digits for the variable.

For example, if the variable is set up to show 6 digits, 1234.56 is rounded to 1234.6. 1234.44 is rounded to 1234.4. The decimal counts as one of the digits.

For more information about how values are rounded, see How values are rounded on page 80.

If the tag value, including the decimal point and minus sign, contains more digits than specified for the variable, the numeric variable is replaced with asterisks.

For example, if the variable is set up to show 6 digits, and the tag value is -123456, the variable will be replaced with asterisks.

Literal numbers are shown using the same rules as numeric tag values.

Number formats

The numeric variable uses the number format of the current application language. For example, if the application language uses a comma for the decimal symbol, the numeric variable uses a comma for the decimal symbol.

For information about using multiple languages, see Set up language switching on page 159.

String embedded variables

For string embedded variables that do not use a fixed number of characters, the entire string tag value is shown, unless a null character is read. Nothing after a null character is shown.

If a fixed number of characters is used, the variable shows the value of the tag up to the number of characters specified, unless a null character is encountered before the specified length. Nothing is shown after a null character. If necessary, spaces are used to make up the required number of characters.

Null characters have a hex value of 0. The null character indicates the end of string input. It does not add to the actual string length.

If the string is longer than the specified number of characters, it is truncated to fit the number of characters. If the SHOWSTAR option is used, an asterisk (*) replaces the first or last character shown. If the embedded variable is set up to show right-most characters, excess characters are truncated at the left end of the string and the asterisk (if used) appears at the left. Otherwise, the right-most characters are truncated and the asterisk (if used) appears at the right.

Literal strings are shown using the same rules as string tag values.

Time and date embedded variables

For embedded variables that show both the time and the date, a space is placed between the time and date when the embedded variable is shown at runtime.

Time and date formats

Time and date embedded variables use the time and date formats for the current application language. For example, if you specify the short date format, at runtime, the display uses the short date format that the application language uses.

For information about using multiple languages, see Set up language switching on page 159.

Use parameters and global objects

This chapter describes how to:

- Use tag placeholders and parameter files or lists
- Create tag placeholders
- Create parameter files and parameter lists
- Use global objects
- Create global object displays and base objects
- Create, set up, and delete reference objects
- Use global object parameters
- Use controller instruction faceplates

The topics in this chapter describe features of FactoryTalk View Studio that can help you set up your applications more quickly by reusing similar groups of objects and graphic displays.

- Tag placeholders, parameter lists, and parameter files allow you to use the same graphic display with different sets of tags.
- Global objects allow you to use multiple instances of the same graphic object and make changes to all of the objects at once.
- Global object parameters allow you to assign different sets of tags to different copies of the object without breaking the link to the base object, thus preserving the ability to update all copies of the object at once.
- Controller instruction faceplates provide pre-configured graphic displays that interact with the instructions in Studio 5000 processors. For the latest controller instruction faceplates, search "process library" on Product Compatibility and Download Center.

Use tag placeholders

Use a tag placeholder on page 305 to mark where you want to insert a tag name or some part of a tag name at runtime. A tag placeholder is a crosshatch character (#) followed by a number from 1 to 500.

FactoryTalk View ME uses three methods for replacing a tag placeholder with a tag name:

- Parameter files
- Parameter lists
- Global object parameters

To use the same graphic display with different sets of tags, use tag placeholders with parameter files or parameter lists. Using tag placeholders can be quicker than duplicating a display and setting up the objects in it to use a different set of tags, especially when the display uses a lot of tags. Using parameter files or parameter lists also reduces the size of the runtime application file.

You can use tag placeholders wherever you would normally assign a tag to an object, including in expressions and embedded variables. You can also use tag placeholders in the expressions you create to animate objects.

Tag placeholders can provide a way to use one graphic display to represent a number of similar operations. For example, suppose you are creating displays for a plant that cans corn and peas. The machinery used in both processes is identical. Instead of creating two displays and specifying corn-related tags in one display and pea-related tags in another, you can create one display and not specify any tag names. Where a tag name is required, type a tag placeholder.

Use parameter files or parameter lists to specify which tags to substitute for which placeholders when a display opens at runtime. For information about using parameter files, see Create parameter files on page 375. For information about using parameter lists, see Create parameter lists on page 377.

You can also use tag placeholders with global objects. You can assign tag placeholders to the connections for base objects, and to connections for reference objects with the **LinkConnections** property set to **False**. For more information about global objects, see Use global objects on page 379.

You can use tag placeholders in:

- The graphic displays that open when the application is first run. Specify the graphic displays to open, and the parameter file or parameter list to use with the graphic displays in the **Startup** editor.
- Graphic displays that are opened using Goto display buttons. Specify the graphic display to open, and the parameter file or parameter list to use with it, when you set up the button.
- Graphic displays that are opened using display list selectors. Specify the graphic displays to open, and the parameter files or parameter lists to use with them, when you set up the display list selector.
- The graphic display that opens when a logout button is pressed.
- The graphic display specified on the **Project Settings Runtime** tab when auto logout is enabled.
- Displays specified on the **Display** tab of the **Global Connections** editor.

Use tag placeholders with parameter files or parameter lists

To use tag placeholders with parameter files or parameter lists

1. In the **Graphics** editor, create graphic objects and assign tag placeholders to the objects.
2. If you are using parameter files, create parameter files in the **Parameters** editor for each set of tags that the display will use. In the parameter files, specify which tags to substitute for which placeholders.
3. In the **Graphics** editor, create Goto display buttons, logout buttons, or display list selectors for opening the display containing tag placeholders. Specify the appropriate parameter files in the graphic object's dialog box.

   To use parameter lists, specify the tags for the lists in the dialog box.
4. If a startup display uses tag placeholders, in the **Startup** editor, specify the parameter file to use with the startup display. To use a parameter list, specify the tags for the list in the **Startup** editor.

   For information about the **Startup** editor, see Create runtime applications on page 177.
5. If auto logout is enabled, you have the option of opening a display on logout. Specify the appropriate parameter file in the Runtime tab of the **Project Settings** dialog box. To use a parameter list, specify the tags for the list in the **Project Settings** dialog box.

   For information about the **Project Settings** editor, see Help.
6. If the displays that will be changed remotely by global connections use parameters, specify the parameter file or parameter list in the **Display** tab of the **Global Connections** editor.

   For information on global connections, see Set up global connections on page 89.

Create parameter files

The parameter file specifies which tags to substitute for the placeholders on the display, by assigning one tag to each unique placeholder on the display. Create a parameter file for each set of tags that you want to use with the same graphic display.

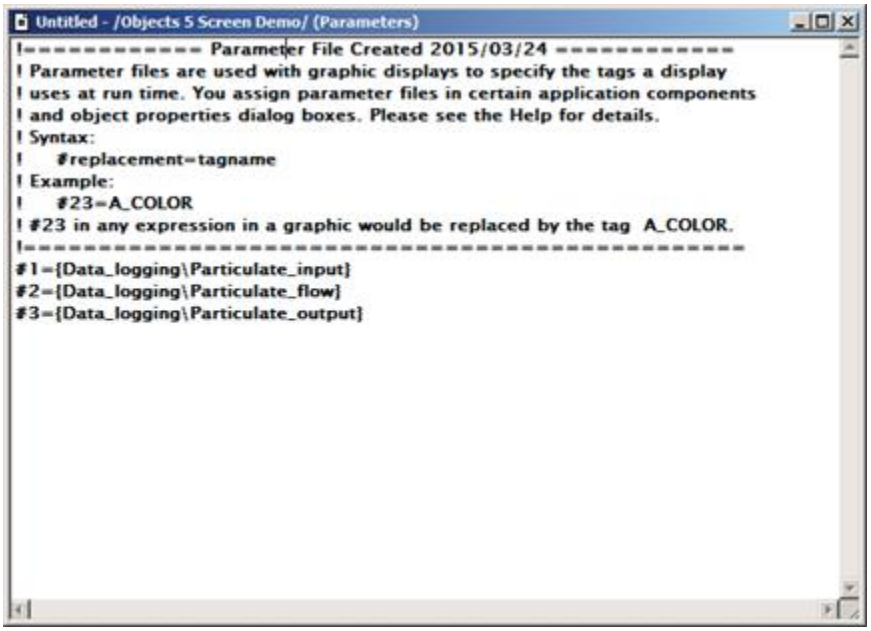
At runtime, the tag values that are shown depend on which parameter file is used when the display opens. When you open the display, the tags specified in the parameter file replace the tag placeholders.

Parameters can replace any portion of a tag address. For example, you can use parameters to replace folder names.

Parameter files can list up to 500 tag placeholders. Use them when a large number of tag placeholders are needed for a display. The text format of a parameter file allows you to see and edit the placeholders easily. When copying displays from one application to another, the related parameter files must also be copied.

Use the Parameters editor

Use the **Parameters** editor to create one or more files of tag placeholder replacements. Each file is stored in the editor's folder. You can open and work on multiple parameter files at the same time.



For details about using the **Parameters** editor, see Help. To open the Help, press **F1** on your keyboard.

Example: Use a parameter file to replace tag placeholders

This example shows how to use a graphic display called Canning Overview with two sets of tags, one for canning corn, and one for canning peas.

The Canning Overview display is opened from a graphic display called Main Menu.

1. Create these sets of tags. You can use both HMI and data server tags.

Tag type	Tag name	Tag name
String	Corn_Name	Pea_Name
Analog	Corn_Temp	Pea_Temp
Analog	Corn_Weight	Pea_Weight
Analog	Corn_Level	Pea_Level

2. Create two parameter files, called *Corn and Peas*, containing these parameters:

Corn	Peas
#1=Corn_Name	#1=Pea_Name
#2=Corn_Temp	#2=Pea_Temp
#3=Corn_Weight	#3=Pea_Weight
#4=Corn_Level	#4=Pea_Level

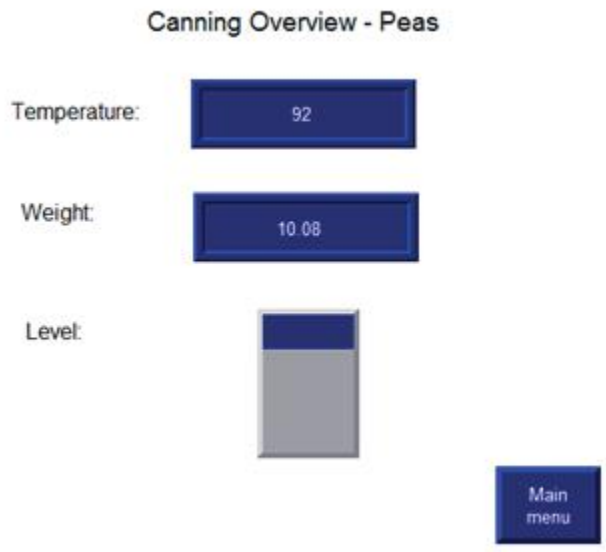
3. In the Canning Overview display, assign tag placeholders to the Value connections for these graphic objects:

This graphic object	Uses this tag placeholder
String display	#1
Numeric display 1	#2
Numeric display 2	#3
Bar graph	#4

4. Use descriptive text to illustrate the objects on the display.
5. In the **Main Menu** display, create a display list selector for opening the Canning display, with two states.
6. Assign the Corn parameter file to one state, and the Peas parameter file to the other.
7. Call the captions for the states Corn and Peas, respectively.

At runtime, when the operator selects the Peas state on the display list selector in the Main Menu, and presses the **Enter** key, the Canning Overview display opens and shows the values of the Pea_ tags. When the operator selects the Corn state, the values of the Corn_ tags are shown.

The Canning Overview display looks like this when the operator selects the Peas state:



Create parameter lists

You create parameter lists by selecting the tags as you set up the objects that will open the displays that use them. The first tag in the list will replace placeholder #1, the second tag in the list will replace #2, and so on. A parameter

list stays with the object when it is copied or duplicated, and with the graphic display in which it is created, when the display is copied or moved.

Parameter lists can be used anywhere that parameter files can be used. The parameter list option appears in:

- The **States** tab of the **Display Selector** object
- The **General** tab of the **Logout** button and Goto display button objects
- The **Initial graphic displays** option in the **Startup** editor
- The **Runtime** tab of the **Project Settings** editor
- The **Display** tab of the **Global Connections** editor

Parameter lists are defined in the object on the display. If the display or the object is copied or imported to another application, the parameter list stays with the object.

To use tag placeholders and parameter lists

1. In the **Graphics** editor, create graphic objects and assign tag placeholders to the objects.
2. In the dialog box where the parameter list option appears, click on the Browse button to open the **Tag Browser**.
3. Select the first tag or folder to add to the list and click **OK**. The tag or folder name appears in the parameter list text box. At runtime, this tag replaces placeholder #1.
4. To add another tag to the list, repeat steps 2 and 3. Each tag in the parameter list box is separated from the previous tag by a comma. At runtime, the second tag replaces placeholder #2, the third tag replaces placeholder #3, and so on.

Example: Use a parameter list to replace tag placeholders

This example shows how to use a graphic display called Canning Overview with two sets of tags, one for canning corn, and one for canning peas.

The Canning Overview display is opened from a graphic display called Main Menu.

1. Create these sets of tags. You can use both HMI and data server tags.

Tag type	Tag name	Tag name
String	Corn_Name	Pea_Name
Analog	Corn_Temp	Pea_Temp
Analog	Corn_Weight	Pea_Weight
Analog	Corn_Level	Pea_Level

2. In the Canning Overview display, assign tag placeholders to the Value connections for these graphic objects:

This graphic object	Uses this tag placeholder
String display	#1
Numeric display 1	#2
Numeric display 2	#3
Bar graph	#4

3. Use descriptive text to illustrate the objects on the display.

4. In the Main Menu display, create a display list selector for opening the Canning display, with two states.
5. In the first state, choose the parameter list option and select the Corn parameter tags from the **Tag Browser** in this order: Corn_Name, Corn_Temp, Corn_Weight, Corn_Level. In the second state, do the same for the Peas tags.
6. Call the captions for the states Corn and Peas, respectively.

At runtime, when the operator selects the Peas state on the display list selector in the Main Menu, and presses the **Enter** key, the Canning Overview display opens and shows the values of the Pea_ tags. When the operator selects the Corn state, the values of the Corn_ tags are shown.

Parameter passing

You can use parameter files or parameter lists to pass (carry forward) parameters from one display to other related displays that open from it. To carry the same parameter forward to subsequent displays, assign the same tag placeholder (for example, #1), instead of a tag name in the parameter list or parameter file field of the object in the related display.

Example: Use a parameter list to replace tag placeholders in linked displays

This example uses PIDE faceplate displays to demonstrate parameter passing.

The main, or first level display, is called MAIN. It has a Goto display button that opens a Logix_PIDE faceplate display. The faceplate has a Goto display button that opens a Trend faceplate.

On the Main display, the Goto display button has the parameter list option on the **General** tab selected, and a structure tag name, with a shortcut, selected from the **Tag Browser** to represent #1: {::[PLC1]Program:Fermenter_Temp.Tank_PIDE01}, for example.

When the button is pressed, the Logix_PIDE faceplate opens. The objects on the faceplate have been set up to read the tags in the structure: #1.CV, #1.SP, and so on. The objects replace the parameter #1 with the structure tag name and then derive their values from the structure elements they have been set up to read.

To carry the same parameters forward another level, the Goto display button on the Logix_PIDE faceplate display has the parameter list option on its **General** tab selected, and the parameter {#1} typed in. When this button is pressed, the Trend faceplate opens, and its objects in turn replace their #1 parameters with the same structure tag name, and read their values from the appropriate elements.

Use global objects

FactoryTalk View global objects allow you to link the appearance and behavior of a graphic object to multiple copies of that object. When you update the original object, the changes are automatically applied to all the copies of the object.

You create global objects in global object displays, in the Global Objects folder of the **Explorer** window. The objects you create in a global object display are called base objects.

You can copy or drag base objects into standard graphic displays. The copied object is called a reference object. You can copy a global object into any number of graphic displays, and multiple times into the same graphic display.

By default, global object displays (.ggfx) are stored in **\Users\Public\Public Documents\RSView Enterprise\ME\HMI projects***Application Name***Global Objects**

where *Application Name* is the name of your application.

Set up global objects

To set up global objects

1. In the **Graphics** editor, create a global object display. For more information, see the next section.
2. Create graphic objects in the global object display. They are the base objects. Set up their appearance, animation, and connections.
3. Copy or drag base objects into a standard graphic display. The copied objects are reference objects. For more information, see Create reference objects on page 381.
4. Edit the link properties of the reference objects as desired. For more information, see Set up reference objects' link properties on page 382.

Create global object displays and base objects

You can create a global object display in the Global Objects folder, or create a graphic display in the Displays folder and then add the display to the Global Objects folder. You can also add library displays to the Global Objects folder.

The objects you create in the global object displays are called base objects.

Create a global object display

To create a global object display

1. In the **Graphics** folder, right-click **Global Objects**, and then select **New**.
2. Create graphic objects on the display. These are the base objects.

   Tip: You cannot use ActiveX objects as base objects. You cannot convert objects to wallpaper in the global object display.

   For information about creating graphic objects, see Use graphic objects on page 275.
3. From the **File** menu, select **Save**, or click the Save tool .
4. In the **Component name** box, type a name for the display, and then click **OK**.
   The display is added to the list in the **Global Objects** folder.

Add a display or library to the Global Objects folder

To add a display or library to the Global Objects folder

1. In the **Explorer** window, in the **Graphics** folder, right-click the **Global Objects** icon.
2. Select **Add Component into Project.**
3. Browse to and select the display or library to add, and then click **Open**.
   The display is copied into the **Global Objects** folder, and given the file extension .ggfx.

Any ActiveX objects are deleted. All other objects are converted to base objects.

Tip: If you delete, remove, or rename a global object display, you break the links between the base objects on the display and their reference objects. For more information about breaking links, see Help.

Create reference objects

FactoryTalk View global objects allow you to link the appearance and behavior of a graphic object to multiple copies of that object. When the original base object is updated, the changes are automatically applied to all the copies of the object. The copies of the base object are called reference objects.

You can copy or drag base objects into standard graphic displays. Each copied object becomes a reference object. You can also copy, drag, and duplicate reference objects that you have already created to create more copies of the base object.

Tip: Base objects can be group objects. This provides powerful template capabilities. When you add or remove objects from the base object group, all the reference objects are automatically updated.

If desired, you can assign unique connections, animation, and size to the reference object.

To create a reference object

1. Copy an object from a global object display to a standard graphic display.
2. To assign unique connections, animation, or size to the object, double-click the object to open the **Property Panel**.
3. For information about using the **Property Panel**, see Use the Property Panel on page 296.

You can also create reference objects by adding a global object display to the Displays folder. All the objects in the new display are reference objects.

Add a global object display to the Displays folder

To add a global object display to the Displays folder

1. In the **Explorer** window, in the **Graphics folder**, right-click the **Displays** icon.
2. Select **Add Component into Project**.
3. Browse to and select the global object display to add, and then click **Open**.

The display is copied into the **Displays** folder, and given the file extension .gfx. All the objects are converted to reference objects.

Edit a reference object's base object

To edit a reference object's base object

1. Right-click the reference object, and then select **Edit Base Object**.
2. The global object display containing the base object opens, with the object selected.

3. Make your changes to the base object.
4. All reference objects linked to the base object are updated.

To view the changes to the reference object, close the display containing the reference object, and then reopen it. Or, toggle the object's **LinkSize** property on and off.

Set up reference objects' link properties

FactoryTalk View global objects allow you to link the appearance and behavior of a graphic object to multiple copies of that object. When the original base object is updated, the changes are automatically applied to all the copies of the object.

The copies of the object are called reference objects, and they have the same properties as the original base object. If desired, you can assign unique size, connections, and animation to the reference object.

To assign unique properties to the reference object, use the **Property Panel** to edit the following properties for the reference object:

- **LinkSize** — select **False** to set up the height and width separately for the reference object. If **LinkSize** is set to true, when you try to resize the reference object, it will snap back to its linked size.
- **LinkConnections** — select False to set up connections separately for the reference object. Whether you use the base object's connections, or set up separate connections, the tags used count towards the tag limit for the display.
  You can assign tag placeholders to the connections for base objects, and to connections for reference objects with the **LinkConnections** property set to False.
- **LinkAnimation**—select **Do not link to set up separate animation** for the reference object. To use the animation set up for the base object but assign different expressions to the reference object, select **Link without expressions**.
  If you select **Link with expressions**, the reference object uses the animation and expressions assigned to the base object.

Reference objects also have a property called *LinkBaseObject*, which specifies the name and location of the base object to which the reference object is linked. You cannot edit the **LinkBaseObject** property.

Change a reference object's link properties

To change a reference object's link properties

1. Double-click the object to open the **Property Panel**.
2. Make your changes to the link properties.
3. Specify the new size, connections, or animation for the object.

For information about resizing objects, see Resize objects on page 309.

For information about assigning connections to objects, see Assign tags and expressions to an object's connections on page 297.

For information about assigning animation to objects, see Animate graphic objects on page 338.

Specify default link properties for reference objects

To specify default link properties for reference objects

1. From the **Edit** menu, select **Global Object Defaults**.
2. Specify the new default values.
3. Click **OK**.

The defaults are used for any new reference objects that you create. You can still edit the properties for individual objects separately.

Break the link to a base object

To break the link to a base object

- Right-click the reference object, and then click **Break Link**.

The reference object becomes a regular object. You can edit all of its properties. However, you cannot restore the connection to the base object.

Delete the base object

If you delete a base object or a global object display containing base objects, any reference objects that are linked to the deleted base objects are broken. A broken reference object is displayed as a red square with an X through it. You cannot edit broken reference objects.

Repair a broken reference object

To repair a broken reference object

- Recreate the base object with the same object type and name as before, on the same display as before.

Use global object parameters

Global object parameters are parameters that you can assign to global objects. You can use a global object parameter to assign different tags or sets of tags to each reference object without breaking the link to the base object. You can then make changes to the base object and all the associated reference objects at the same time.

Use global object parameters when more than one instance of a global object is used on a display. For example, the Logix_PIDE global object display contains a grouped object composed of other grouped objects. The objects' connections are set up with tags and expressions that use values from a set of Studio 5000 tags. When you create multiple reference objects from this base object, each reference object can use a different set of Studio 5000 tags. The global object parameters you assign to the base object allow you to do this, because you are using a placeholder instead of a specific backing tag (a backing tag is a path to a folder of tags; it is also known as a structured tag). If you then change, add, or remove a tag or expression in the base object, the same change is made to all the reference objects.

If you don't use global object parameters, you can still assign different tags to different reference objects by changing the reference object's LinkConnections property to False and the LinkAnimations property to Link without expressions.

The global object parameter takes the same form as a regular parameter. The parameter can be the placeholder for an individual tag or for a folder of tags. For example, #1 could be a placeholder for the path to the folder containing the PIDE tags assigned to the global object.

When you set up the base object, specify the global object parameters to use with the object. You can provide a description of each parameter to remind you or another application designer of the type of value to assign to the parameter on the reference object. Then assign specific values to each parameter for the reference object. You can assign numeric or string constants, tags, or backing tags.

You can create up to 1000 global object parameters.

Difference between global object parameters and regular parameters

Global object parameters allow you to assign different values to different instances of the same placeholder. For example, each reference object on the display might have the placeholder #1. Using regular parameter files, you could only assign one value to #1, and this would apply to all objects in the graphic display. With global object parameters, you can assign a different value to #1 for each object that uses the placeholder #1.

If a placeholder is defined in a global object parameter for an individual object and defined in a parameter file, the definition assigned to the object takes precedence.

Use global object parameters with group objects

To use global object parameters with a group object, the parameters are defined at the group level, not at the level of individual objects. Thus the definition applies to each object within the group. You can assign as many parameters as desired to the group. For example, you might assign #1 to some members of the group and #2 to other members of the group. If you create a global object parameter definition for an object and later group the object, the definition is deleted. In addition, if you create a global object parameter definition for a group and then ungroup the object, the definition is deleted.

Values are assigned to the global object parameters at the group level as well. You cannot assign separate values to individual objects in the group.

Use global object parameters

To use global object parameters

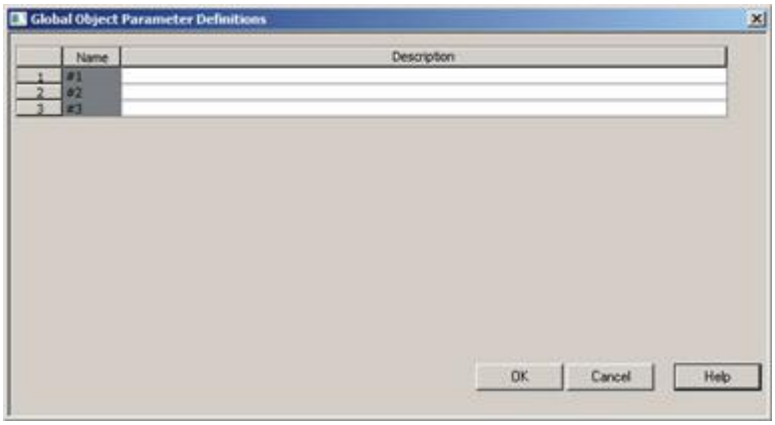
1. On the global object display, assign parameters to the base objects and add definitions to the parameters. For details, see the next section.
2. On the standard graphic display, select the reference object and specify the value to use for each parameter. For more information, see Create reference objects on page 381.

Specify global object parameters for the base object

To specify global object parameters for the base object

1. Create placeholders in each place where you want to use a global object parameter. You can create the placeholders anywhere a tag or expression is required.
   You can also type placeholders in embedded variables by using a literal string embedded variable. For information about embedded variables, see Use embedded variables on page 367.

2. Right-click the global object or grouped global object, and then select **Global Object Parameter Definitions**.



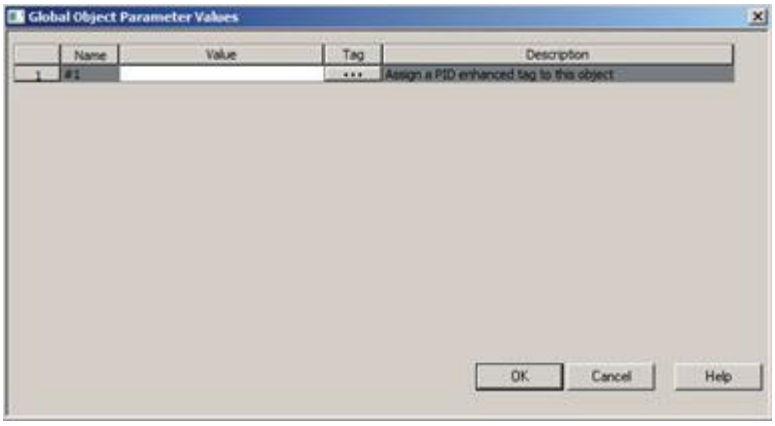
3. (Optional) Type a description for each parameter.

For details about using the **Global Object Parameter Definitions** dialog box, see Help.

Specify the parameter values for the reference object

To specify the parameter values for the reference object

1. Right-click the reference object or grouped reference object, and then select **Global Object Parameter Values.**



2. Specify a value for each parameter. The value can be a tag, backing tag, or numeric or string constant. Numeric constants are treated as strings of text.

About controller instruction faceplates

Controller instruction faceplates are pre-configured graphic displays and global object displays that interact with Studio 5000 processors. These displays contain graphic objects that display values from a Studio 5000 processor and allow operators to interact with the processor. You can use the faceplate graphic displays as standalone displays, or copy the faceplate objects onto other graphic displays. You can also edit the objects on the faceplates. For example, you might want to add your company logo or change the colors used on the faceplates.

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) does not add to the total license count.

For the latest controller instruction faceplates, search "process library" on Product Compatibility and Download Center.

Set up data logging

This chapter describes:

- Steps for setting up data logging
- Data log files
- Data log models
- How to change the data log model used at runtime
- How to show data logs using the trend graphic object
- Problems with data logging

Set up data logging

To set up data logging

1. In the **Data Log Models** editor, set up a data log model that specifies how many log values to store, the conditions that trigger data logging, where to log data, and which tags to monitor.
2. In the **Startup** editor, turn on data logging by selecting the **Data logging** checkbox. Also use this editor to specify the data log model to use at runtime. See Create runtime applications on page 177.
3. In the **Graphics** editor, create a graphic display containing the trend graphic object, and set up how the object looks, which tag values to show, the start time, and the time span for the data. Also specify the name of the data log model to use.

Data log files

As soon as the application starts running, FactoryTalk View begins logging tag values to the data log files. When the maximum number of data points have been logged, the oldest data is deleted to make room for the new data. FactoryTalk View supplies data from the log files to the trend object for the requested tags and time span.

The data log files are retained when you restart an application after a shutdown or power loss. You can delete the log files from the runtime computer at application startup.

Use data logging to keep a permanent record of tag data. You can record tag data as tag values change, or on a periodic basis (for example, every minute).

For information about deleting the log files, see Delete log files on the runtime computer on page 196.

File names

FactoryTalk View creates two data log files:

- *Data Log Model Name*.log
- *Data Log Model Name*.tag

You'll need this information if you log to a custom path and want to delete the log files manually.

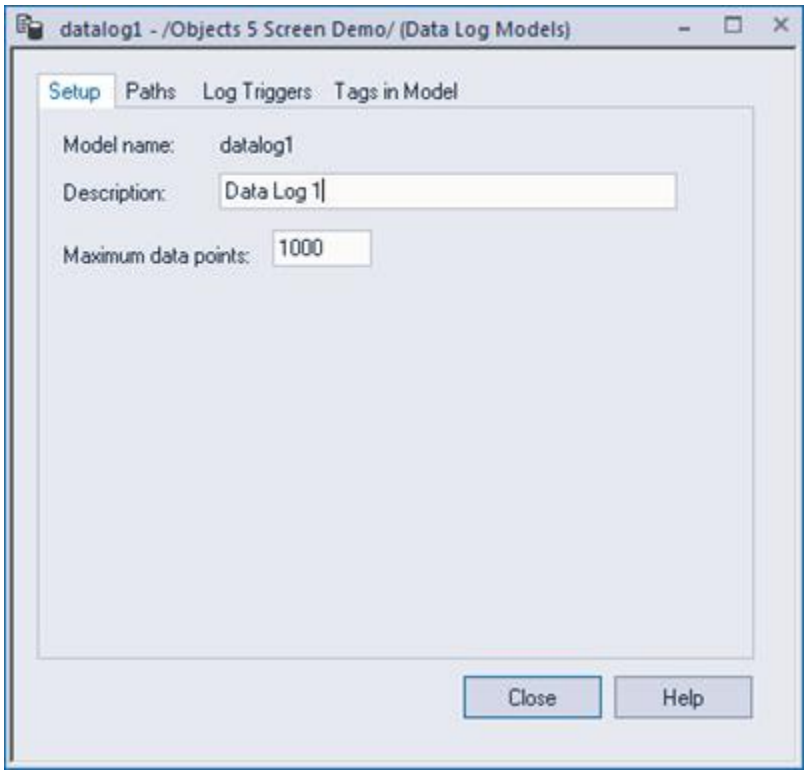
Data Log Models

A data log model defines which tags to log data for, as well as how and where the data is logged.

You can set up multiple data log models, but you can run only one data log model at a time. For information about running a different data log model, see Change the data log model used at runtime on page 390.

Create Data Log Models

Use the **Data Log Models** editor to create one or more data log models. Each data log model is a file stored in the editor's folder. You can open and work on multiple models at the same time.



For detailed information about the options in the **Data Log Models** editor, see Help.

Each data log model has a unique name, and an optional description.

You can log a maximum of 1,000,000 points for version 7.00.00 or later, and you can log 300,000 points for release versions earlier than 7.00.00. The default is 1000. When the maximum number of data points have been logged, the oldest data is deleted to make room for the new data.

Data storage locations

You can store data log files in any one of these locations:

- On the runtime computer
- On another computer on the network
- On a memory stick for a MobileView terminal
- On an SD card for a PanelView Plus 7 or PanelView Plus 6 terminal
- On a compact flash card for a PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal

If the logging destination is unavailable for any reason, for example, the card is removed or the network drive becomes unavailable, data logging stops. It does not restart until you restart the application.

Log to a network location

To log data to a network location, the network drive must be shared, and the runtime computer must be logged in to the same domain as the computer on the network. To do this, the user must have access rights for the domain.

The PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal cannot be part of a domain. However, you can verify that the user who is logged into the terminal is on a user list that is part of a domain.

For more information about logging in to a Windows domain or authenticating users, see Log in to the application on page 210.

Data logging methods

There are two methods for triggering data logging. You can set up logging so that tag values are logged:

- Periodically (periodic logging)
- Only when a tag value changes (on-change logging)

Log periodically

Periodic logging is used to take a snapshot of all tag values at a particular point in time. Tag values are logged even if there has been no change.

You cannot change the periodic log rate at runtime.

Log on change

On-change logging is used to log only tags whose values have changed.

For HMI tags, before logging occurs, the change must equal a specified percentage of change in the tag value. The percentage is based on HMI tags' minimum and maximum values as set up in the **Tags** editor. Only the tags that change by the specified percentage are logged. If you specify a percentage of 0, all changes are logged.

If a tag does not have a minimum or maximum attribute (for example, a data server tag in Studio 5000), when you specify on-change logging, all changes are logged for that tag. You don't need to specify a percentage.

Use a heartbeat rate

You can also specify a heartbeat rate, to log values at specified times even if no tag value changes have occurred. The heartbeat ensures that the data in the log file is current. The heartbeat is also a good way to ensure that data logging is working and acquiring valid data.

The heartbeat cannot be less than the maximum update rate, which is the rate at which data servers send tag values to FactoryTalk View.

If you specify a heartbeat of 0, the heartbeat is not used.

Tags in the data log model

The data log model can contain up to 100 analog or digital tags, including both HMI and data server tags.

You cannot use string tags, array tags, tag placeholders, parameters, or expressions in your data log model.

Delete tags from the model

If you delete a tag from the data log model, previously logged data for the tag is not accessible unless you add the tag back to the model.

Change the data log model used at runtime

Your application can run only one data log model at a time. Follow this procedure to switch data log models.

To run a different data log model

1. Shut down the application.
2. Start FactoryTalk View Studio and open the application.
3. In the **Startup** editor, specify the new data log model.
4. Create the runtime application. See Create runtime applications on page 177.
5. Transfer the runtime application to the runtime platform.
   For information about transferring applications to:
   - A personal computer, see Run applications on a personal computer on page 183.
   - A MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, see Transfer applications to a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal on page 199.
6. Run the new application.

Show data logs using the trend graphic object

You can use the trend graphic object to show the data that you have logged. At runtime, when the operator opens a graphic display containing a trend, the trend shows values from the data log file for the data log model that is running. The data log model specifies which data to collect in the data log file.

In addition to showing historical values from the data log file, trends can show current values for the tags in the model. Trends can also show current values for tags or expressions that are not in a data log model. For more information about trends, see Set up trends on page 398.

Problems with data logging

Problems with data logging occur under these circumstances:

- When your application starts at runtime, if any of the tags specified in the current data log model do not exist, an error message is sent to FactoryTalk Diagnostics.
- If the data log file is corrupted or invalid, the file is deleted and recreated, and an error message is sent to FactoryTalk Diagnostics.
- If logging to a folder on a networked computer, and the runtime computer is not logged in to the Windows domain of the network computer, the log folder cannot be created. An error message is sent to FactoryTalk Diagnostics.

- If logging to an invalid path, the log folder cannot be created. An error message is sent to FactoryTalk Diagnostics. One of the reasons that the path might be invalid is that the top-level folder of the path is not shared.
- When the application starts, FactoryTalk View checks whether there is disk space to store the data log model's data. If there is not enough space, an error message is sent to FactoryTalk Diagnostics and data logging does not start.

Use information messages

This chapter describes:

- Steps for setting up information messages
- How to use the **Information Messages** and **Information Setup** editors
- How to prepare to set up information messages
- How information messages work
- How to create information messages in multiple languages
- The [INFORMATION] display
- How to create your own information message display
- How to open and close the information message display
- How the information message display graphic object works
- How to change the information message file used at runtime

About information messages

Use information messages to give the operator messages about the process, prompts or instructions, and information about the current states.

Tip: To give the operator information on a specific graphic display while the display is open, use local messages.

For details about local messages, see Local messages on page 269.

Set up information messages

Follow the steps below to set up information messages.

To set up information messages

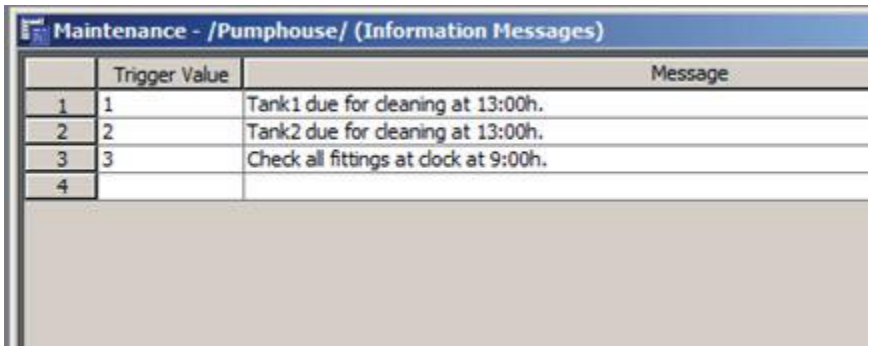
1. In the **Information Messages** editor, set up the messages and their trigger values.
2. In the **Information Setup** editor:
   a. Specify the graphic display to open when information messages occur, and the file of messages to show.
   b. Assign a tag or expression to the **Value** connection.
      If you want the operator to acknowledge messages, assign the **Ack** connection and specify the acknowledge hold time.
3. In the **Startup** editor, select the **Information messages** checkbox is selected.
4. (Optional) In the **Graphics** editor, modify the default [INFORMATION] display or create your own graphic display to use for information messages.
   For example, if you don't want the operator to acknowledge messages, edit the default display to remove the information acknowledge button.

   Tip: When renaming information message files, update the reference to the message file in the **Information Setup** editor to the new name.

For information about graphic displays, see Use graphic displays on page 252.

Use the Information Messages editor

Use the **Information Messages** editor to create one or more files of information messages. Each file is stored in the editor's folder. You can open and work on multiple message files at the same time.

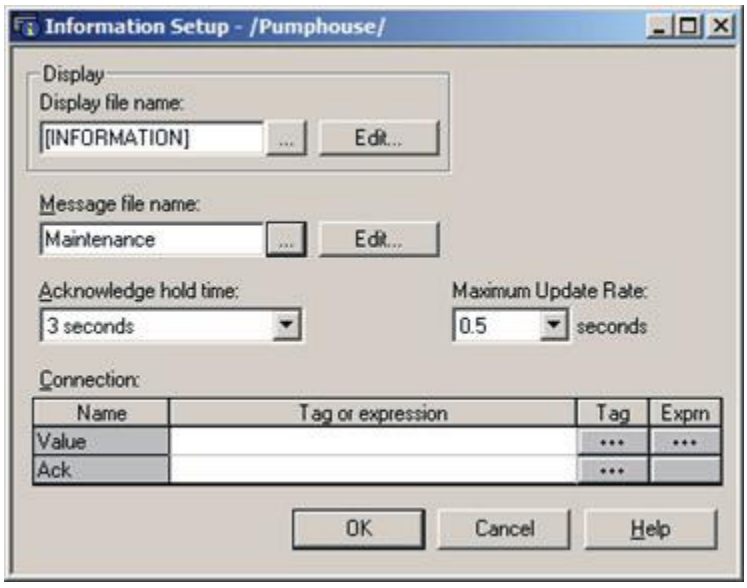


You can define up to 10,000 messages in each message file.

For detailed information about the options in the **Information Messages** editor, see Help.

Set up how information messages are shown

Use the **Information Setup** editor to set up how information messages are shown.



For detailed information about the options in the **Information Setup** editor, see Help.

Prepare to set up information messages

As your application is running, information is continually sent to the data source about the state of various processes. For example, your application might be monitoring whether a valve is open or closed, or the temperature in a boiler. Values representing the status of these processes are sent to the data source. For more information about data sources, see Data source on page 68.

Identify the tags and values to monitor

To set up information messages, determine which tags associated with machine processes to monitor, and identify the values for those tags that will trigger information messages.

For information about creating tags, see Use HMI tags on page 79.

How information messages work

These are the key parts of the information message system:

- **Information message file** — a text file containing a list of messages, with a numeric trigger value for each message
- **Information message display** — a graphic display that opens at runtime and shows information messages
- **Value connection** — a tag or expression. When the value of this connection matches a message's trigger value, the information message display opens with the associated message shown.

The following example shows how the key parts of the information message system work together.

Example: Set up the data source to show information messages

This example shows how to set up the data source to notify the operator that a conveyor belt has stopped. In this example, the status of two conveyor belts is being monitored. You can use both HMI and data server tags.

1. Create a digital tag called Conveyor_1_status. This tag points to an address in a programmable controller that is linked to a sensor on the first conveyor belt. When the belt is running, the tag's value is 0. When the belt stops running, the value changes to 1.
2. Create a second digital tag called Conveyor_2_status. This tag points to an address in a programmable controller that is linked to a sensor on the second conveyor belt. When the belt is running, the tag's value is 0. When the belt stops running, the value changes to 1.
3. Create an analog tag called Information_messages. Set up the data source to send a value of 1 to this tag when Conveyor_1_status has a value of 1, and to send a value of 2 to this tag when the Conveyor_2_status tag has a value of 1.
4. In the **Information Messages** editor, create these messages with trigger values matching the values that will be sent to the Information_messages tag:

Trigger value	Message
1	Conveyor belt 1 has stopped.
2	Conveyor belt 2 has stopped.

5. Save the message file with the name *Conveyor belts*.
6. In the **Information Setup** editor, assign the **Information_messages** tag to the **Value** connection, and select the **Conveyor belts message** file.

At runtime, when the value of Conveyor_1_status changes to 1, the first message is shown (in the default information message display). When the value of Conveyor_2_status changes to 1, the second message is shown.

Information messages and trigger values

Create messages associated with each tag value change that you want to inform the operator about. Assign each message a trigger value, and set up the data source to send the trigger value to the **Value** connection. You can use both HMI and data server tags.

The trigger value 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 values of 10, 20, and 30.

Trigger values cannot be 0. Digital tags have two possible values, 0 and 1. Therefore, if you use a digital tag you can only use the value 1 to trigger a message. If you want to use a digital tag to trigger two different messages, create an expression that adds 1 to the digital tag's value. That way, you can use the trigger values 1 and 2.

If you use an analog tag or an expression, you can use any non-zero integer or floating-point value to trigger an alarm. Floating-point values are rounded to the nearest integer. For information about how values are rounded, see How values are rounded on page 80.

You can create multiple information message files, but you can use only one message file at runtime. For information about using a different message file, see Change the message file used at runtime on page 397.

Create information messages in multiple languages

FactoryTalk View 5.00 and later supports information messages in multiple languages. When you create information messages, they are in the current application language. You can export the information messages for translation and then import them back into the application. For details, see Set up language switching on page 160.

Language switching information messages in FactoryTalk View ME Station 4.00

For applications that will run in FactoryTalk View ME Station version 4.00, use the CurrentLanguage( ) expression function to specify message offsets in the information message file. In the file, divide your messages into sections for each language. For information about the CurrentLanguage( ) function, see Language function on page 363.

The [INFORMATION] display

The Libraries folder contains a graphic display called [INFORMATION]. It contains an information message display graphic object and buttons for acknowledging the shown message and closing the display. By default, the [INFORMATION] display opens automatically at runtime when an information message is generated. You can also place the information message display object on your own graphic display to show information messages.

Item	Name	Description
1	Information message display graphic object	It lists one information message at a time.

Item	Name	Description
		For information about how the information message display object works at runtime, see How the information message display graphic object works on page 396.
❷	Ack (information acknowledge)	Press this button to acknowledge the information message. When the operator presses the button, if the **Ack connection** is assigned, the connection value is set to 1 at the data source. The value is held as long as the operator presses the button, or for the acknowledge hold time, whichever is longer. Then the connection is reset to 0.
❸	Close (close display)	Press this button to close the information message graphic display.

Use the [INFORMATION] display as is or modify the display. For example, you can change the color of the objects, add and remove buttons, or assign any caption you choose to the labels on the buttons. You can also create your own graphic display to show information messages. In the **Information Setup** editor, specify the display to use.

For information about creating an information message display, see Create your own information message display on page 396.

Create your own information message display

You can create your own graphic display for showing information messages, containing an information message display graphic object and the buttons you want to use on the display.

If you create your own graphic display, use an On Top display and select the Cannot Be Replaced option.

For more information about the information message display graphic object, see How the information message display graphic object works on page 396.

For information about creating graphic displays and graphic objects, see Use graphic displays on page 252 and Use graphic objects on page 274.

Open and close the information message display

The information message display that you specify in the **Information Setup** editor (either the default [INFORMATION] display or your own display) is automatically opened whenever the **Value** connection value matches a trigger value.

You can also create a **Goto display** button that the operator can press to open the information message display. For information about setting up a Goto display button and specifying the display to open, see Help.

The information message display closes when the **Value** connection value changes to 0.

To automatically close the display when the operator acknowledges a message, set up the data source to set the **Value** connection to 0 when the **Ack** connection is set to 1.

The operator can also close the display by pressing a close display button.

How the information message display graphic object works

At runtime, when the **Value** connection at the data source changes from 0 to a nonzero value, the assigned information message display opens. If the value matches a message's trigger value, the associated information

message appears in the information message display graphic object. The object can be in the default [INFORMATION] display, in an information message display you have created, or can be placed in any display in your application.

What is shown?

- If the **Value** connection is unassigned, the information message graphic display never opens automatically. If the operator opens a graphic display containing an information message display object, the object is blank.
- The **Value** connection is rounded up to the nearest integer. If the value does not match any of the trigger values in the specified message file, the display is filled with question marks (?).
  For information about how values are rounded, see How values are rounded on page 80.
- If the message is too long to fit in the information message display object, the last shown character is replaced with an asterisk (*).
- When the **Value** connection value is 0, the information message graphic display is closed.
- If you set up information messages in multiple languages, messages are shown in the current application language. When a language switch occurs, a message that was already in the information message display remains in the language that it originally appeared in. New messages are shown in the new language.

Change the message file used at runtime

Your application can use only one message file at a time.

To switch message files

1. Shut down the application.
2. Start FactoryTalk View Studio and open the application.
3. In the **Information Setup** editor, specify the new information message file.
4. Create the runtime application. See Create runtime applications on page 177.
5. Transfer the runtime application to the runtime platform.
   For information about transferring applications to:
   - A personal computer, see Run applications on a personal computer on page 183.
   - A MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, see Transfer applications to a MobileView, PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal on page 199.
6. Run the new application.

Set up trends

This chapter describes:

- What trends are
- Summary of steps for creating a trend graphic object
- How to create trends and the **Trend Object Properties** dialog box
- The parts of the trend graphic object
- The different chart types
- How to choose colors, fonts, lines, and markers for the trend
- How to test the trend
- How to use objects from the Trends graphic library
- How to use buttons to control the trend at runtime
- How to print trend data
- Runtime errors for trends

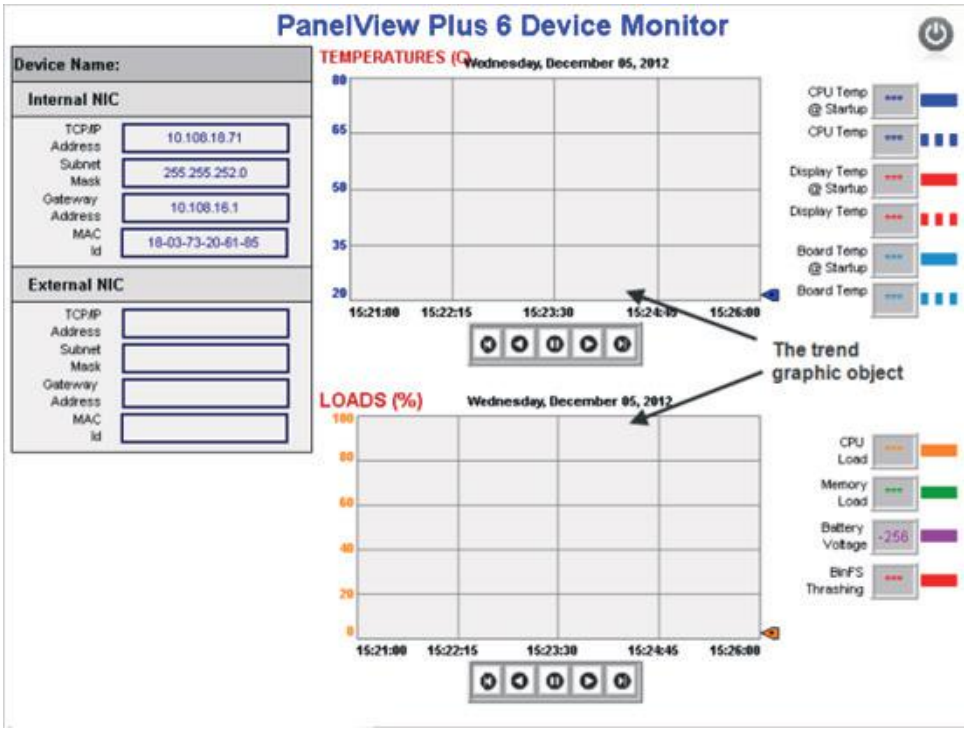
About trends

A trend is a visual representation of current or historical tag values. The trend provides operators with a way of tracking plant activity as it is happening.

You can:

- Plot data for as many as eight tags or expressions on one trend.
- Create a trend that is part of a graphic display or acts as the entire graphic display.
- Plot data over time, or plot one variable against another in an XY Plot chart to show the relationship between them.
- Show isolated or non-isolated graphs. Isolated graphing places each pen in a separate band of the chart. With non-isolated graphing, pen values can overlap.
- Create buttons to allow the operator to pause, scroll, and print the trend data.

The illustration below shows a trend that has been added to a graphic display. You can view the following display by opening the InstantFizz_ME sample application.



Current versus historical data

The data shown in a trend can come from two sources. For current values, data comes from the value table as it is collected. The value table is a record of the most recent values collected from the data source, and is stored in temporary memory while the application is running.

For historical values, data comes from a data log model's log file, if a model is assigned to the trend. You can show both current and historical data in the same trend.

For information about data log models, see Set up data logging on page 387.

Time, date, and number formats

The trend is shown using the time, date, and number formats of the current application language. For example, if the application language uses a comma for the decimal symbol, the scale on the y-axis uses commas for the decimal symbol.

For information about using multiple languages, see Set up language switching on page 159.

Create a trend

To create a trend

1. To plot historical data, create a data log model in the **Data Log Models** editor. For more information, see Set up data logging on page 387.
2. Create a trend graphic object in the **Graphics** editor, as described on Create a trend object on page 400.

3. Set up the trend in the **Trend Object Properties** dialog box. For details about the options in the dialog box, see Help.
4. If desired, create a next pen button, a pause button, or key buttons on the same graphic display, to allow the operator to switch between pens, pause the trend, or scroll the trend.
   For information about the buttons that you can use with trends, see Use buttons to control the trend at runtime on page 407.
5. To keep a printed record of the trend data, provide a way for the operator to print the graphic display. For more information see Print trend data on page 408.

Create a trend object

To use a trend object, create a trend on a graphic display and configure it to show data from a data log model.

To create a trend

1. In the **Graphics Display** editor, select **Object > Trending > Trend** or drag  from the toolbox.
2. Drag the mouse to position and draw a rectangle the general size and location you want the trend to be.
3. Double-click the trend object to open its **Properties** dialog box.
4. In the **Properties** dialog box, specify:
   - How the trend looks.
   - How the trend works at runtime.
   - The pens to use on the trend.
   - The horizontal axis and vertical axis.
   - The tags or expressions to plot.

NOTE:

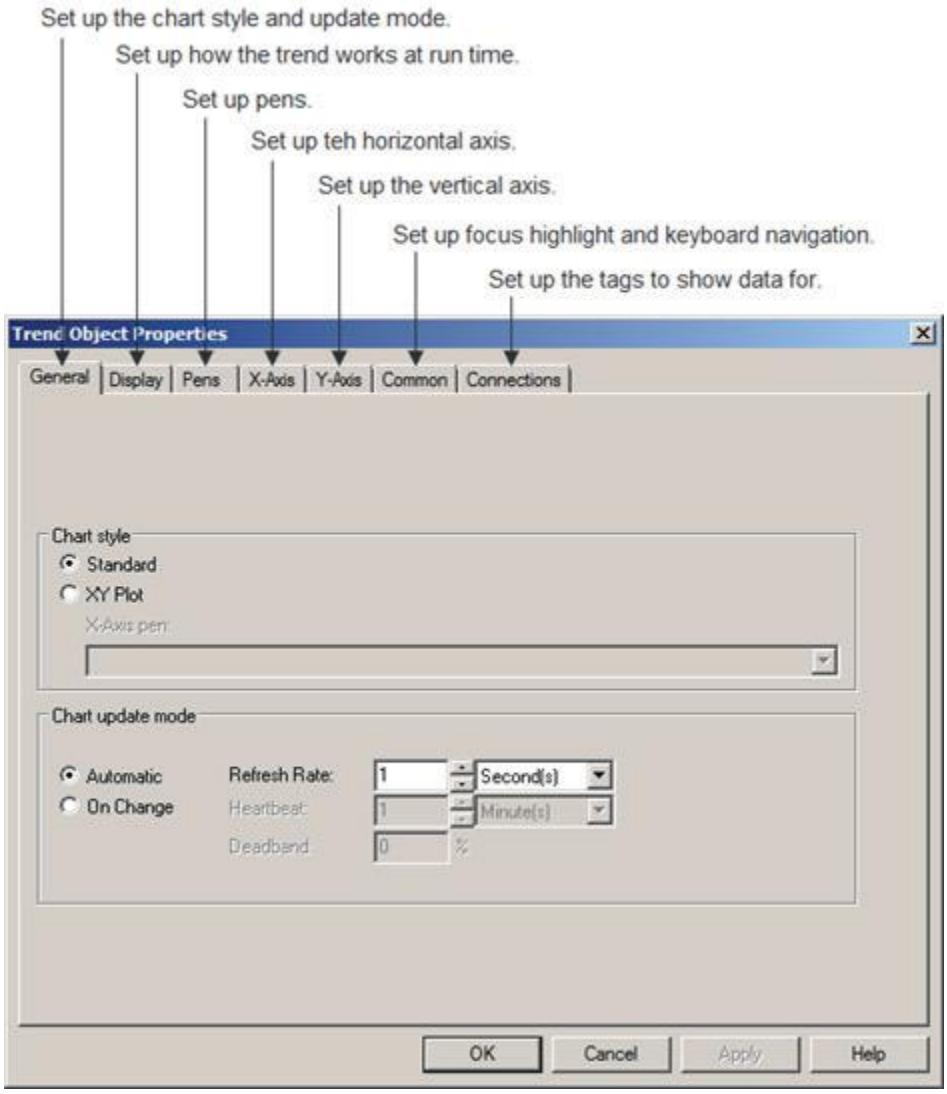
- If trend pens are configured with the maximum buffer size of 32,767 records, the entire allocated program memory on a CE terminal will be consumed in a few hours, resulting in an "out of memory" error. This requires a reboot of the terminal before the application can continue.
  To avoid this problem, set the buffer for extra data below the theoretical maximum.
- If you use two or more pens with the **Isolated graphing** display option selected, the colors of the pens will no longer match the legend and will appear offset.
  To avoid this problem, add and use a **Next Pen** button to align the legend with the pen.

Once you have set up the trend, you can move it, resize it, attach animation to it, and the like. You can also use this object on other graphic displays by dragging it from one display and dropping it onto another.

For more information about graphic objects, see Use graphic objects on page 274.

Set up trends in Trend Object Properties

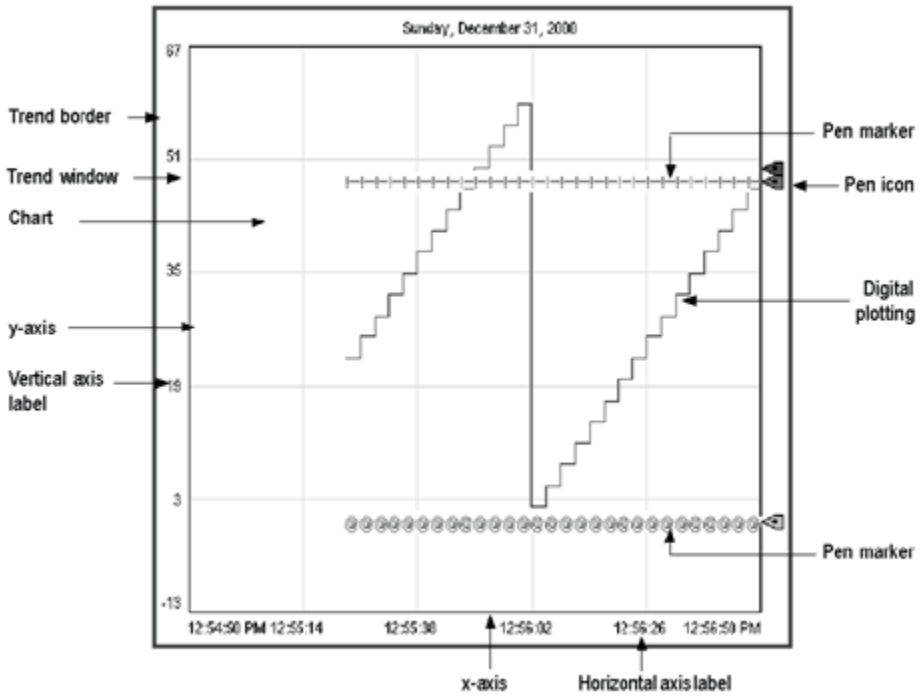
When you double-click a trend object, the **Trend Object Properties** dialog box opens. Use the dialog box to set up the trend.



For details about the options in the **Trend Object Properties** dialog box, see Help.

The parts of a trend

The illustration below shows a standard trend chart, with three pens and a two-minute time span. Two of the pens have markers. The third uses digital plotting. For more information about chart types, see Standard versus XY Plots on page 403.



Trend border

The border appears around the trend object at runtime when the trend is selected.

Trend window

The area around the chart, between the border and the chart, is the trend window.

Chart

The chart is the area of the trend in which values are plotted. It is bounded by the y-axis on the left and the x-axis on the bottom. It contains the plotted trend data (shown using pen lines and pen markers), as well as grid lines (if you choose to show them).

Y-axis

The y-axis is the left vertical edge of the chart. It is also known as the vertical axis.

Vertical axis labels

The vertical axis labels show the scale (range) of values for the pens. If desired, you can set up the trend to omit the vertical axis labels.

The minimum and maximum values for the scale can be determined automatically (using the best fit for the current data), be derived from a pen's minimum and maximum values, use a constant value, or be controlled by tags.

You can set up the trend so all pens use the same scale, or use individual ranges for each pen. If you choose the latter method, create a next pen button in the graphic display, to allow operators to view the range for each pen. When the operator presses the button, the vertical axis changes to the new pen's range.

For example, if Pen 1 has a minimum value of 10 and a maximum value of 100, the range 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 range on the vertical axis changes to -10 to 50 when the operator presses the next pen button.

X-axis

The x-axis is the bottom horizontal edge of the chart. It is also known as the horizontal axis.

Horizontal axis labels

For standard charts, the horizontal axis labels indicate the time span covered by the trend. For XY Plot charts, the horizontal axis labels show the scale (range) of values for the pen selected to serve as the x-axis pen.

If desired, you can set up the trend to omit the horizontal axis labels. The number of labels depends on the size of the trend object and the number of vertical grid lines.

Pens

Pens are the lines and symbols used to represent values. The values can be tags you are monitoring, expressions that manipulate tag values, or constants.

If there is no data for a pen, or if the data is outside the vertical axis range, the pen does not appear in the chart.

Pen icons

Pen icons appear at the right edge of the chart at runtime, if you choose to show them. The icon's position indicates the pen's most recent recorded value (from the value table), even if the trend is paused or if the most recent value has not been plotted yet.

Pen markers

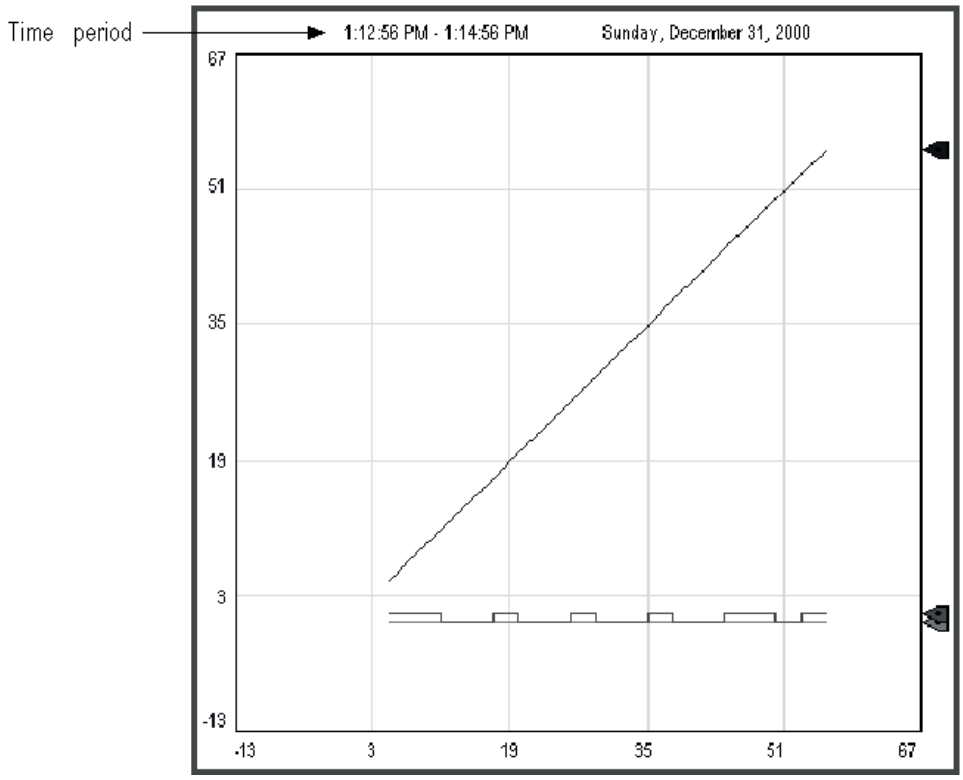
Pen markers are symbols that indicate data points. If data is plotted frequently, the markers might not appear as distinct, separate symbols. For example, see the lowest pen in the illustration on The parts of a trend on page 402.

Chart types

Standard versus XY Plots

You can create a standard chart, which plots tag values against time, or an XY Plot chart, which plots one (or more) tag's values against another's.

This illustration shows what an XY Plot chart could look like:

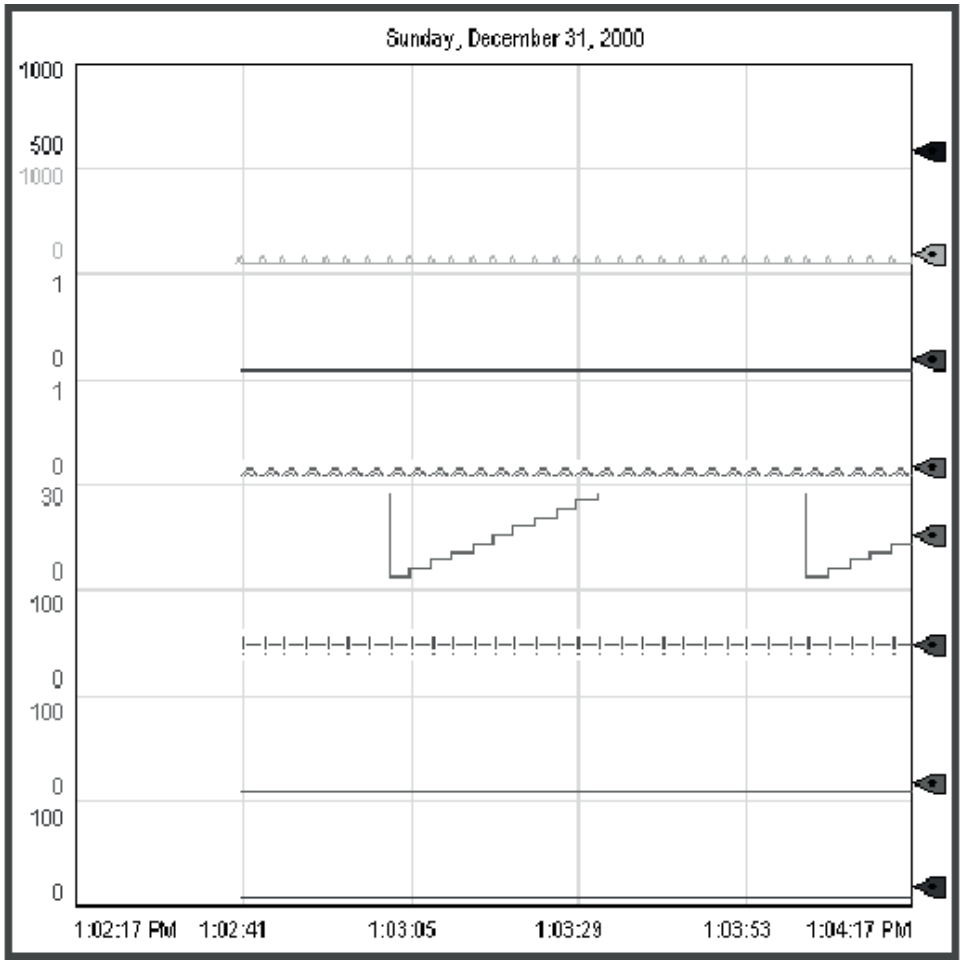


Notice that the horizontal axis labels show the range for the specified x-axis pen. The time period covered by the chart is at the upper left.

Isolated graphing

For charts with multiple pens, you can allow the pen values to overlap, or you can isolate each pen in its own horizontal band on the chart.

This is an example of isolated graphing, with a 10% buffer between each pen's band:



Notice that in this illustration each pen uses its own scale. If desired, you can use the same scale for all pens.

With isolated graphing, a grid line is automatically placed above each pen's band.

Plot a value across the full width of the chart

Use horizontal lines to provide a frame of reference for your tag data. For example, if you define values that are the limits within which a tag must operate, and show horizontal lines in your trend to indicate the limits, when a tag crosses one of these limits the tag's alarm condition is obvious on the trend.

There are two ways to plot a value across the full width of the chart:

- On the **Connections** tab, assign a constant value to a pen.
  When values for the pen have been plotted across the full width of the chart, the pen appears as a solid line.
- On the **Connections** tab, assign to a pen the tag, expression, or constant whose value will be used to determine the position of the line, and then on the **Pens** tab, choose the pen type **Full Width**.
  As soon as the trend is shown, the pen appears as a horizontal line across the full width of the chart. Its vertical position is determined by the tag, expression, or constant's value. If the value changes, the position changes.

Choose trend colors, fonts, lines, and markers

The following table summarizes where in the **Trend Object Properties** dialog box to specify colors, fonts, lines, and markers for a trend.

You can also specify these settings in the **Properties** tab of the **Property Panel**.

To specify this	Use this box or column	In this tab
Chart background color	Background color	Display
Horizontal label color	Text color	Display
Text font, style, and size	Font (button)	Display
Pen line, pen marker, pen icon, and vertical label color	Color	Pens
Pen line width	Width	Pens
Pen line style	Style	Pens
Pen marker	Marker	Pens
Vertical grid line color	Grid color	X-Axis
Horizontal grid line color	Grid color	Y-Axis

The trend border color

The trend border uses the highlight color for the graphic display, specified in the **Behavior** tab of the **Display Settings** dialog box.

The trend window color

By default, the trend window uses the background color of the display, specified in the General tab of the **Display Settings** dialog box.

To use a different window color

- In the **Property Panel**, select the opaque WindowStyle, and then specify the WindowColor property.

For information about using the **Property Panel**, see Use the Property Panel on page 296.

Test the trend

You can quickly test the trend by switching to test mode. If communications are active and there is data for the tags, the pens plot values in the trend. When you are finished testing, switch back to edit mode to continue editing.

To switch between test and edit modes

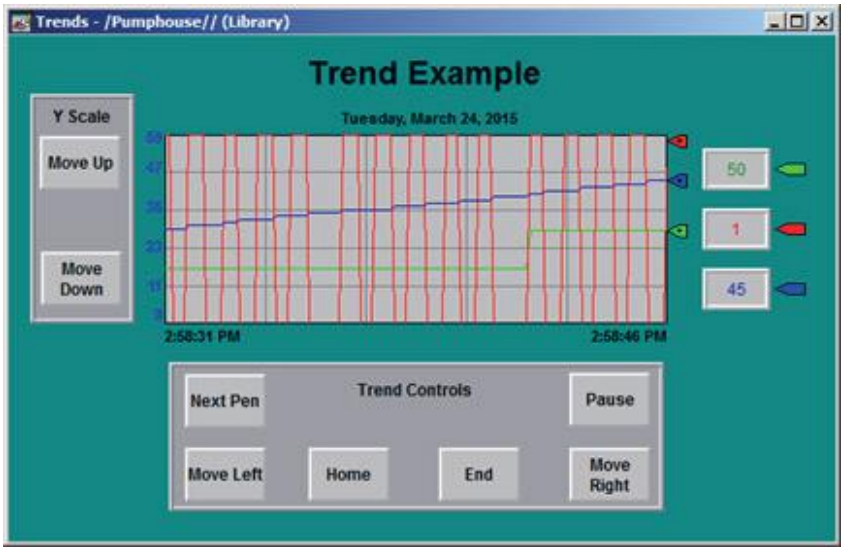
- From the **View** menu, select **Test Display** or **Edit Display**, or click the **Test Display** tool ▶, or the **Edit Display** tool ■.

> **Tip:** Test mode is not the same as running the display. Test mode does not change the appearance or position of the display as set up in the **Display Settings** dialog box. Also, data logging is not turned on in test mode.

Use the Trends graphic library

The Trends graphic library contains a trend graphic object and buttons for controlling the trend. It also contains numeric display objects that show the value of each tag used in the trend.

You can use the trend and objects as they are, or you can edit them to suit your needs. To use the objects, drag and drop (or copy and paste) them into your graphic display.



To use the Trends graphic library

1. Open the **Graphics** folder, and then open the **Libraries** folder.
2. Double-click the **Trends** library.
3. Drag and drop or copy and paste objects into your display.

Use buttons to control the trend at runtime

You can use button graphic objects with the trend, to allow the operator to pause the trend, switch between pens, or scroll the trend. You can link buttons to a specific trend object, or set up a button to work with whichever object is selected in the graphic display. For information about linking buttons to objects, see Link buttons to objects on page 332.

Use these buttons with trends:

This button	Does this
Pause	Toggles between pausing and automatic scrolling.

This button	Does this
	When the trend is paused, the pen icons continue to move vertically to indicate the pens' current values. When the trend resumes scrolling, values that occurred while the trend was paused are filled in, bringing the trend up to the current time (unless you are scrolling historical data).
Next pen	Changes the vertical axis labels to the scale for the next pen. The color of the labels matches the color of the selected pen.
Move up	Scrolls up to show higher values on the vertical scale. For example, if the visible scale range is 0 to 100, pressing move up could change the visible range to 10 to 110. The incremental amount that the axis scrolls depends on the pen's range and the number of horizontal grid lines. This button does not work if the **Minimum / maximum value** option in the **Y-Axis** tab is set to **Automatic**.
Move down	Scrolls down to show lower values on the vertical scale. This button does not work if the **Minimum /maximum value option** in the **Y-Axis** tab is set to **Automatic**.
Move left	Pauses the trend and scrolls to the left.
Move right	Pauses the trend and scrolls to the right.
Home	Pauses the trend and moves to the earliest data in the trend.
End	Resumes trend scrolling and moves to the current (latest) data in the trend.

To see how the buttons work with the trend, open the Trends graphic library (see Use the Trends graphic library on page 407), and start test mode.

For information about creating buttons, see Use graphic objects on page 274. For details about setting up the buttons, see Help.

Print trend data

To print trend data at runtime, provide the operator with a method for printing the graphic display.

You can use these methods to print graphic displays at runtime:

- Create a display print button. For information about creating graphic objects, see Use graphic objects on page 274.
- Assign a tag or expression to the **Remote Display Print** connection (in the **Global Connections** editor). When the value of the tag or expression changes from 0 to a nonzero value, the current display is automatically printed.
  Program the data source to trigger the change as often as you want the data printed.
  For more information about setting up remote display printing, see Set up global connections on page 89.

Everything on the screen is printed, including the current display, pop-up windows, and any visible background applications.

For information about specifying which printer to use at runtime for applications that will run on a personal computer or a MobileView terminal, see Specify the printers to use at runtime on page 191.

For information about specifying printer options for applications that will run on a PanelView Plus 7, PanelView Plus 6, PanelView Plus, PanelView Plus Compact, or PanelView Plus CE terminal, see the *PanelView Plus Terminals User Manual*.

Improve clarity of the trend printout

Depending on what type of printer you use, pen lines with a width of one pixel might not appear in the printout. Choose high-contrast colors and wider line widths to ensure that the trend data prints clearly.

Runtime errors for the trend

If data for the trend is not available at runtime due to communication errors, a message is sent to FactoryTalk Diagnostics.

See Help for information about solving common trend problems.

Set up RecipePlus

This chapter describes:

- What recipes are
- Summary of steps for creating a recipe system
- How the recipe system works
- How to specify the runtime location of recipe files
- How to create recipe files
- How to compare recipes
- How to create **RecipePlus** buttons, selectors, and tables
- How to test **RecipePlus** objects
- How to use objects from the **RecipePlus_Components** graphic library
- How to use buttons with the recipe objects
- How to view data values that are saved at runtime

About recipes

A recipe is a set of numeric and string data values (ingredients) that can be downloaded to their associated tags at the data source. Each ingredient has a pre-set data value assigned to it. The set of data values for all the ingredients in a recipe is called a data set. The set of numeric and string tags assigned to the ingredients in the recipe is called a tag set. The ingredients, data sets, and tag sets are stored together in a recipe file.

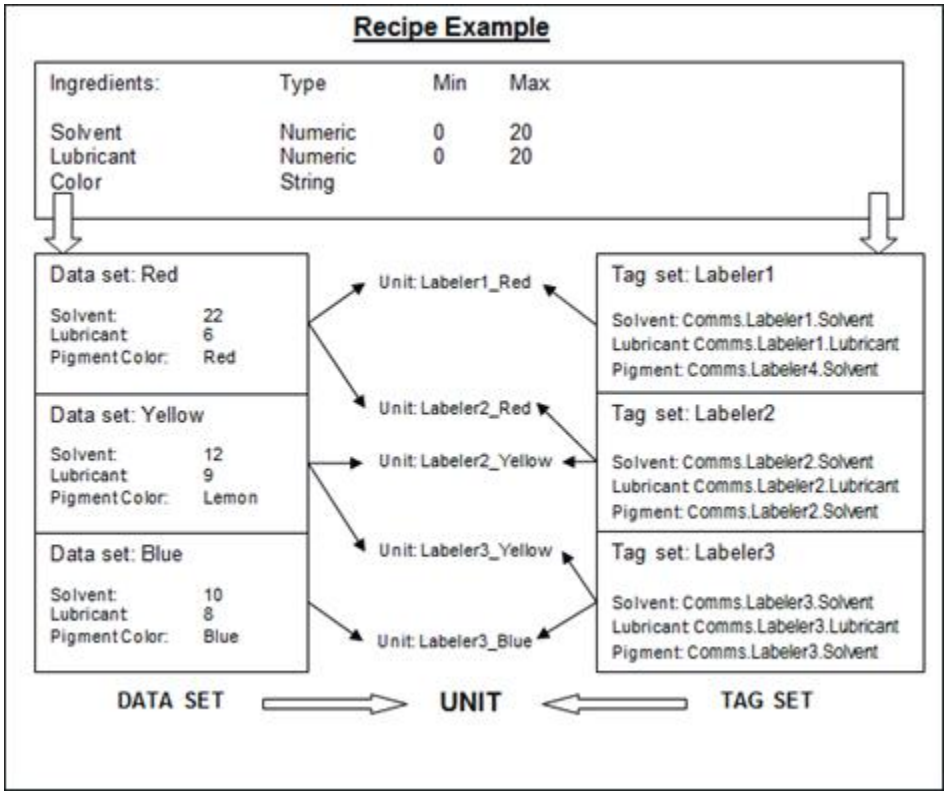
You can create different pairs of data sets and tag sets for the same set of ingredients. Each pairing of a data set with a tag set is called a unit. Each unit is like a unique recipe. At runtime, the operator can select the unit (recipe) that applies to the current operation.

For example, a bakery making whole wheat bread could use the same ingredients and tag sets, but depending on the type of crust desired, could use different data sets to specify different baking temperatures. As another example, you might want to have multiple production lines baking the same bread. In this case, the data set for all the production lines would be the same, but the tags receiving the recipe information would be different for each production line. Units allow you to combine different tag sets and data sets for the same set of ingredients.

The FactoryTalk View RecipePlus system allows you to create up to 15,000 ingredients, 500 data sets, 50 tag sets, and 2,500 units for each recipe file. You can create data sets at development time, edit them at runtime, and create new data sets from tag values at runtime. You can write the data set values to tags, or write tag values to data sets.

The RecipePlus system can be used for manufacturing food and beverages, but it can also be used for any application where you want to show, edit, download, or save multiple values at once. For example, recipes are used in the petrochemical and pharmaceutical industries. In the pharmaceutical industry, you could use recipes to design flexible packaging, creating recipes that specify the number of tissues to put in a box or the number of milliliters of shampoo to put in a bottle.

Units make it possible to combine different tag sets and data sets for the same set of ingredients.



As shown in the example, there are three labelers. Each one has its own set of tags which are grouped in a tag set. Each tag in the tag set is linked to an ingredient.

In the recipe, there are three different data sets of label colors: red, yellow, and blue.

Every data type within the data set is linked to an ingredient.

These three data sets can be used by any labeler because a unit is used to combine a tag set and a data set.

Create a recipe system

To create a recipe system

1. In the **RecipePlus Setup** editor, specify the runtime location for recipe files. The files can be stored with the application or in a separate location. For details, see Specify the runtime file location on page 413.
2. In the **RecipePlus Editor**, set up ingredients, data sets, tag sets, and units. You can also specify a percent complete tag and a status tag for the recipe. For details, see Set up recipe files on page 414.
3. Create a display in the **Graphics** editor, containing a **RecipePlus** selector, table, and buttons. For details, see Create RecipePlus objects on page 417.
4. If desired, create key buttons on the same graphic display, to allow the operator to use the selector and table without a keyboard.
   For information about the buttons you can use with recipes, see Use buttons with recipe objects on page 418.

How the recipe system works

A recipe system consists of a recipe file and the graphic objects used to work with the ingredients at runtime.

RecipePlus selector

Use the **RecipePlus** selector to select the recipe file and unit to work with.

RecipePlus table

Use the **RecipePlus** table to show the selected recipe file's ingredients, tag values, and data set values. The operator can edit data set values in the table, unless you select the **View only** option.

If desired, you can include a **Compare** column in the table, to compare tag values to data sets at a glance. If you choose this option, FactoryTalk View shows an X in the **Compare** column when the tag value and data set value for an ingredient differ. Ingredients with an X are listed first.

RecipePlus button

Use the **RecipePlus** button to perform actions on the selected recipe's ingredients. The recipe is selected using the **RecipePlus** selector object. Set up a separate **RecipePlus** button for each action you want to perform:

- **Download** — write the data set values to tags, for all the ingredients in the selected recipe.
- **Upload** — write tag values to the data set, for all the ingredients in the selected recipe. If all values are uploaded successfully, the recipe file is saved.
- **Upload and Create** — write tag values for all the ingredients in the selected recipe to a new data set, creating a new unit. The operator is prompted for a name for the new unit. If all values are uploaded successfully, the recipe file is saved.
  The new data set is named Data Set *n*, where *n* is the next available number (starting at 1) that will create a unique data set name.
- **Restore** — show the selected recipe in the **RecipePlus** table.
- **Save** — save the data set values for the recipe file and unit shown in the **RecipePlus** table. If the operator made changes in the data set values using the string pop-up keyboard or numeric pop-up keypad, the new values in the table overwrite existing data set values (if any) for the unit in the recipe file.
- **Delete** — delete the recipe unit selected in the **RecipePlus** selector object.
- **Rename** — rename the recipe unit selected in the **RecipePlus** selector object.

The following illustration shows a graphic library display that contains a **RecipePlus** selector, **RecipePlus** buttons, and a **RecipePlus** table. The display also contains key buttons for working with the selector and table, a bar graph

that shows the percentage complete of the recipe operation, and a string display that shows the status of the recipe operation.

Number format

The values in the recipe table are shown using the number format of the current application language. For example, if the application language uses a comma for the decimal symbol, floating point values in the table use commas for the decimal symbol. For information about using multiple languages, see Set up language switching on page 159.

Numeric limits

RecipePlus supports the range of numbers allowed by the VARIANT data type. This range is -1.797693E+308 to 1.797693E+308. This range applies to the numbers that you enter in the **RecipePlus Editor**, and it applies to the tag values that are uploaded to the recipe file at runtime.

Specify the runtime file location

Use the **RecipePlus Setup** editor to specify the runtime file location.

Store files outside the HMI project

If you store the recipe files outside the HMI project, the runtime application can use updated recipe files without creating a new runtime application (.mer) file.

Storing recipe files outside the HMI project also allows you to use FactoryTalk View Studio to view and edit recipe data that is saved at runtime without converting the .mer file to an .med file. For more information, see View data values that are saved at runtime on page 419.

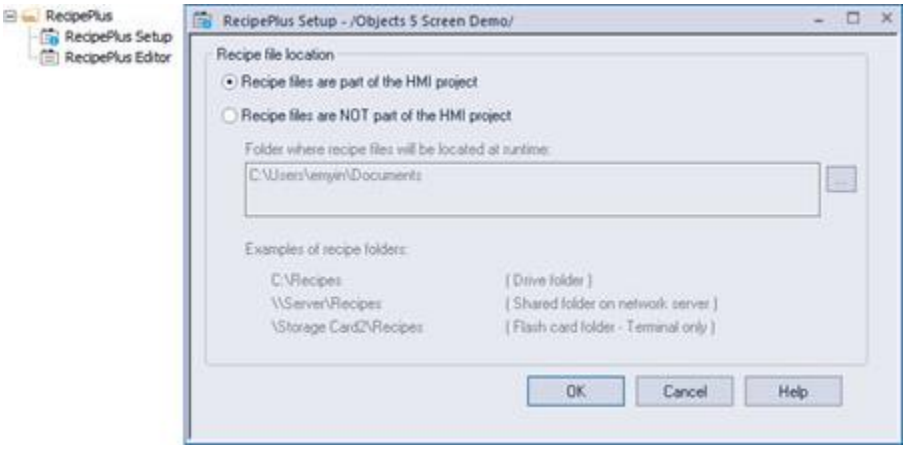
Tip: If you want to store recipe files outside of the HMI project at runtime, make sure you move the files from the application's **RecipePlus** folder to the specified runtime location before running the application.

By default, the **RecipePlus** folder is **\Users\Public\Public Documents\RSView Enterprise\ME\HMI projects ***Project name***\RecipePlus**

If recipe files are stored outside the HMI project, when you perform an action on a recipe file at runtime, the file is locked until the action is completed. This helps prevent other users from changing a file while you are working with it.

Store recipe files with the HMI project

If recipe files are part of the HMI project when a recipe file is saved at runtime, FactoryTalk View updates the .mer file with changes to the data sets. When you stop the runtime application, the changes are retained, and are shown the next time you run the application and show the recipe file. You can convert the .mer file to an .med file to view the changes in FactoryTalk View Studio. For more information about converting runtime application files, see Convert runtime application files to development applications on page 181.



For details about using the **RecipePlus Setup** editor, see Help.

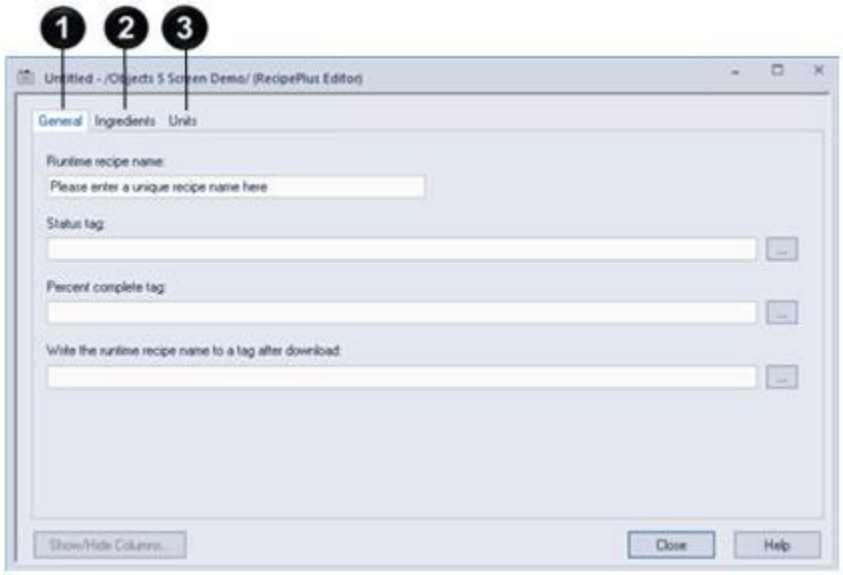
Set up recipe files

Use the **RecipePlus Editor** to set up one or more recipe files. Each file is stored in the editor's folder. You can open and work on multiple recipe files at the same time.

The **RecipePlus Editor** has special items on the **Edit** menu that allow you to easily copy and paste from the spreadsheet on the **Ingredients** tab to Microsoft Excel. This editor also has items on the Recipe menu, for adding, deleting, and renaming data sets and tag sets, and for comparing recipes.

To help you get started, FactoryTalk View creates one data set, tag set, and unit. You can rename them and assign data values and tags to them, or delete them and create your own.

For information about comparing recipes, see the next section.



Item	Name	Description
1	General tab	Specify the runtime name, status tag, percent complete tag, and the string tag to write the runtime recipe name and unit name to after download.
2	Ingredients tab	Set up ingredients, minimum and maximum values, decimal places, data sets, and tag sets
3	Units tab	Set up units (parts of data sets and tag sets).

For details about the options in the **RecipePlus Editor**, see Help.

Tip: You can also use the **RecipePlus Editor** to view the data values that are saved at runtime. For more information, see View data values that are saved at runtime on page 419.

Compare recipes

Use the **RecipePlus Editor** to compare data sets and tags sets within a single recipe, or between two recipes, and generate a report of the differences.

If you are comparing data sets or tag sets within a recipe, only ingredients with different values are shown in the report.

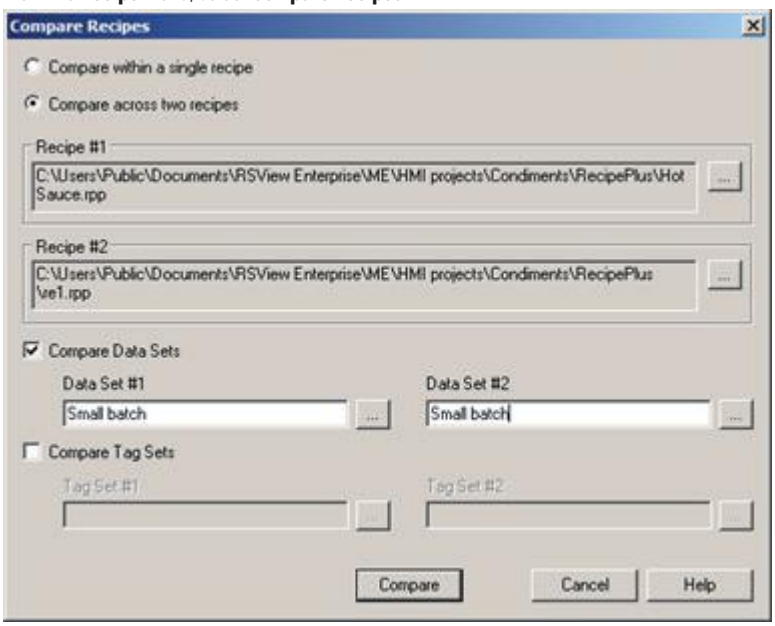
If you are comparing data sets or tag sets between two recipes, both common ingredients and unique ingredients are listed in the report.

The report is a Unicode text file, named *CompareReport.txt* and is saved in the same folder as the recipe file. If a comparison report already exists, it will be overwritten.

Tip: If you are comparing a recipe that is open in the editor to another recipe, save the changes before comparing. The Compare operation works with the saved file on disk, not with the file currently in memory.

To compare recipes

1. Open the **RecipePlus Editor**.
2. From the **Recipe** menu, select **Compare Recipes**.

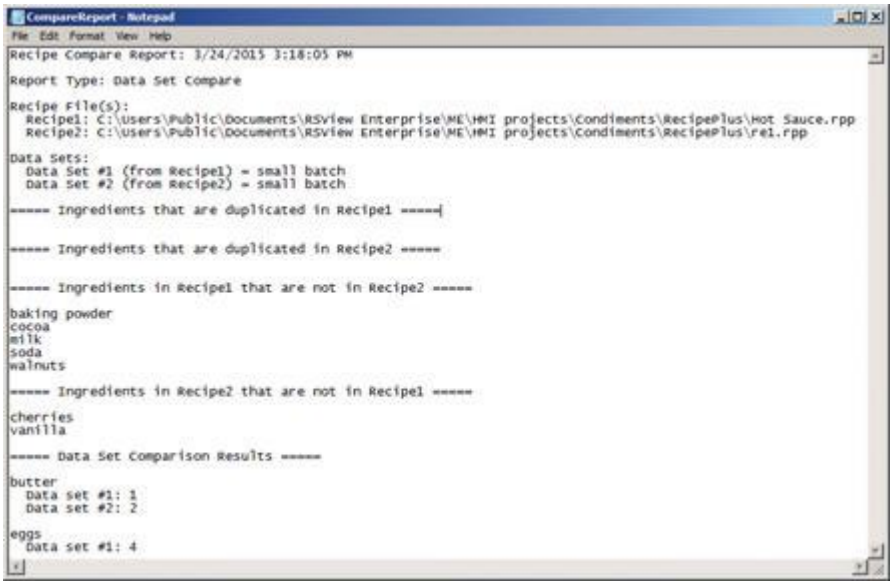


3. Specify the recipe file or files, data sets, and tag sets to compare.

   For information about the options in the **Compare Recipes** dialog box, see Help.

4. Click **Compare**.

   A report is shown in the Windows Notepad.



The time and date in the report use the time and short date format for the current application language. For information about using multiple languages, see Set up language switching on page 160.

Print recipes

You can use the **RecipePlus Editor** to print recipe data sets. For information about printing from editors, see Print on page 40.

Create RecipePlus objects

You can create a **RecipePlus** table and a **RecipePlus** selector per graphic display. You can create multiple **RecipePlus** buttons in a display, with a different action assigned to each.

The objects and button actions to use depend on how you want to use your recipe system. For example, if you want to write data set values to tags, all you need is a **RecipePlus** selector and a **RecipePlus** button with the download action. For information about how the different objects in the recipe system work, see How the recipe system works on page 412.

To create a recipe object

1. In the **Graphics Display** editor, select **Objects > RecipePlus** and then select the object you want to create, or drag the icon from the toolbox:
   - RecipePlus Button 
   - RecipePlus Selector 
   - RecipePlus Table 
2. Drag the mouse to create a box approximately the size you want for the object.
3. Double-click the object to open its **Properties** dialog box.
4. Set up the object. For details, see Help.

Once you have set up a RecipePlus object, you can edit and move it, resize it, attach animation to it, and so on. You can also use the object on other graphic displays by dragging it from one display and dropping it into another.

For more information about graphic objects, see Use graphic objects on page 274.

Test RecipePlus objects

You can quickly test the recipe objects in a display by switching to test mode. If communications are active and there is data for the tags, you can download and upload recipe tag values. When you are finished testing, switch back to edit mode to continue editing.

To switch between test and edit modes

- From the **View** menu, select **Test Display** or **Edit Display**, or click the **Test Display** tool  , or the **Edit Display** tool .

Tip: Test mode is not the same as running the display. Test mode does not change the appearance or position of the display as set up in the **Display Settings** dialog box. Also, data logging is not turned on in test mode.

Use the RecipePlus_Components graphic library

The **RecipePlus_Components** graphic library contains a **RecipePlus** selector and table and buttons for working with the objects. It also contains a bar graph and multistate indicator that show the status of recipe operations.

Use test mode to see how the different **RecipePlus** objects work together. In test mode, the **RecipePlus** selector in the library shows any recipe files and units that you have created in your application.

You can use the objects in the library as they are, or you can edit them to suit your needs. To use the objects, drag and drop (or copy and paste) them into your graphic display.

To use the RecipePlus_Components graphic library

1. Open the **Graphics** folder, and then open the **Libraries** folder.
2. Double-click the **RecipePlus_Components** library.
3. Drag and drop or copy and paste objects into your display.

Use buttons with recipe objects

You can use button graphic objects with the **RecipePlus** selector and table, to select the recipe and unit to work with, and to select ingredients in the table.

You can link buttons to a specific recipe object, or set up a button to work with whichever object is selected in the graphic display. For information about linking buttons to objects, see Link buttons to objects on page 332.

Use these buttons with recipe objects:

This button	Does this
Move up	Moves the highlight bar up one item in the list.
Move down	Moves the highlight bar down one item in the list.
Page up	Moves the highlight bar up one page in the list.
Page down	Moves the highlight bar down one page in the list.
Home	Moves the highlight bar to the top item in the list.
End	Moves the highlight bar to the bottom item in the list.
Enter (table only)	Opens the numeric keypad or string keyboard for the operator to edit the data set value. If a numeric ingredient has a minimum and maximum value defined, these values are shown in the numeric keypad.

This button	Does this
	If the table is defined as View only, the operator cannot edit it.

To see how the buttons work with the **RecipePlus** selector and table, open the **RecipePlus_Components** graphic library (see Use the RecipePlus_Components graphic library on page 417), and start test mode.

For information about creating buttons, see Use graphic objects on page 274.

Example: Edit and downloading recipe values at runtime

This example shows how to use the **RecipePlus** graphic objects to edit and download recipe values at runtime.

1. In the **RecipePlus** editor, create a **RecipePlus** file containing ingredients, several data sets, a tag set, and several units combining the different data sets with the tag set.
2. Open the **RecipePlus_Components** library.
3. Start test mode.
4. Use the move up and move down buttons next to the **RecipePlus** selector to highlight a unit in the selector, and then press the **Restore** button.
   The unit's ingredients are shown in the **RecipePlus** table, with the data set values in the **Recipe** column.
5. Use the move up and move down buttons next to the **RecipePlus** table to select an ingredient, and then press the **Enter** button.
   The numeric pop-up keypad opens, showing the minimum and maximum values for the ingredient. If the ingredient is a string ingredient, the string pop-up keyboard opens.
6. Type a new value for the ingredient, and then press **Enter**.
   The new value is shown in the **Recipe** column.
7. Press the **Save** button to save the new value.
8. Press the **Download** button to write all the values in the Recipe column to the tags associated with the ingredients.
   The values are downloaded to the data source.

View data values that are saved at runtime

Use the **RecipePlus Editor** in FactoryTalk View Studio to view data values that have been saved at runtime.

The operator can save tag values at runtime by uploading to an existing data set or to a new data set. The operator can also edit data set values in the **RecipePlus** table and save the edited values (unless the table is View only).

If recipe files are stored with the HMI project, changes are saved in the .mer file. To view the changes in FactoryTalk View Studio, convert the .mer file to an .med file. For more information about converting the runtime application file, see Convert runtime application files to development applications on page 181.

To view data values in modified recipe files

1. If recipe files are stored outside of the HMI project, do one of the following:
   - Add the recipe file (*.rpp) that you saved at runtime into the application using **Add Component into Project** (for details, see Add components to an application on page 244).
   - If the recipe file already exists in the application, you can just copy the modified file back into the application's **RecipePlus** folder. (For the path to the **RecipePlus** folder, see Store files outside the HMI project on page 413.)

2. If recipe files are stored with the HMI project, and you have not already done so, convert the runtime application file to a development application, as described on Convert runtime application files to development applications on page 181.
3. In the **Explorer** window in FactoryTalk View Studio, double-click the modified recipe file.
   The **RecipePlus Editor** opens.
4. Click the **Ingredients** tab.
5. If the data set you want to view is not visible, scroll right to see more data sets.

Use macros

This chapter describes:

- How to use macros to assign values to tags
- How to use the **Macros** editor
- When to use macros
- How to run macros when tags or expressions change value
- Where to assign macros

Use macros to assign values to tags

A macro is a list of tag assignments stored in a text file, in the format <tag>=<value>. Each assignment assigns a value to a tag. The value can be in the form of another tag, an expression, a numeric constant, or a string. Whenever you run the macro, the specified values are written to the tags.

Example

Tag1 = 8

Sets the value of Tag1 to 8.

Tank1\Message = "Tank1 overflow"

Sets the string tag Tank1\Message to Tank1 overflow.

Tag1 = Tag2

Sets the value of Tag1 to be the same as Tag2.

Tag1 = Tag1 + 1

Increases the value of Tag1 by 1.

Tag1 = if (Tag2 < Tag1) then 4 else 3

Performs the if-then-else calculation and stores the result in Tag1.

1Pump = {Industry-2} + {2Pump}

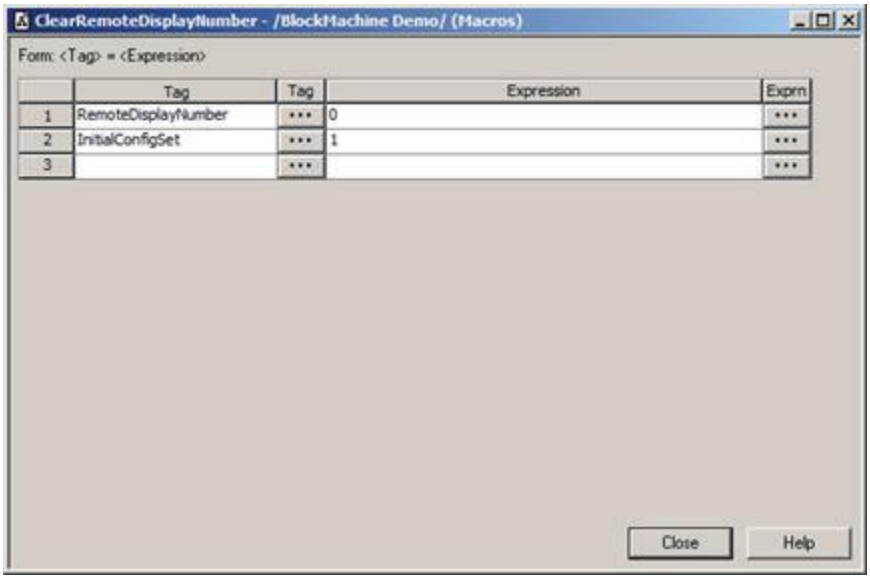
Adds the values of Industry-2 and 2Pump and stores the result in 1Pump.

Brackets surround Industry-2 because of the dash in the name. Brackets surround 2Pump because the name starts with a number. No brackets are used for 1Pump because this name is on the left side of the equal sign.

For more information about expression syntax, see Use expressions on page 352.

Use the Macros editor

Use the **Macros** editor to create macros.



For details about using the **Macros** editor, see Help.

When to use macros

You can assign macros to run when:

- The application starts or shuts down.
- A graphic display opens or closes.
- A user logs in or out. Macros can be assigned to individual users and to groups of users.
- A specified tag or expression changes to a new nonzero value (using global connections).
- An operator presses a Macro button.

At runtime, when the macro runs, the values are sent to the tags at the data source.

Tip: At runtime, the tag assignments are executed asynchronously. That is, the system does not wait for the completion of one tag assignment before executing the next. Therefore, do not rely on the order of assignments to control your process.

Run macros when tags or expressions change value

You can use global connections to run macros when tags or expressions change value. This means you can use the data source to trigger the macro to run.

FactoryTalk View allows you to create up to five macros for use with global connections. The macros must be named Macro1, Macro2, Macro3, Macro4, and Macro5 in order to work with global connections.

For more information about global connections, see Set up global connections on page 89.

Example: Use macros to reset tag values

This example shows you how to run a macro whenever the operator needs to reset production information tags to known values.

The macro writes the desired values to the tags whenever the operator presses a Momentary push button.

1. Create a memory tag called *ResetProdData*.
2. Create a **Momentary** push button with the caption *Reset Production Data*. Assign the **ResetProdData** tag to the **Value** connection.
3. Create a macro called *Macro1*, with these tag assignments:
   TotalProductionUnits=0
   LineDownTime=0
   TotalRejects=0
4. In the **Global Connections** editor, assign the **ResetProdData** tag to the **Remote Macro1** connection.

When the operator presses the **Reset Production Data** button, the value of the **ResetProdData** tag changes from 0 to 1. This tells FactoryTalk View to run Macro1, which writes the specified values to the tags in the macro.

Where to assign macros

Once you have created the macros that you want to use, assign the macros in these editors:

In this editor	Do this
Startup	Assign application startup and shutdown macros.
Graphics	Assign macros to run when displays open or close, using the **Display Settings** dialog box.
Runtime Security	Assign macros to run when users log in and log out. Macros that you assign to groups of users run each time any member of the group logs in or logs out.
Global Connections	Specify the tags or expressions that will run the macros named Macro1 to Macro5.

Features supported in different versions of FactoryTalk View

This appendix describes:

- Which versions of FactoryTalk View ME Station are supported
- Which features are not supported in previous versions of FactoryTalk View ME Station

Which versions are supported

FactoryTalk View Studio allows you to create runtime (.mer) files for multiple versions of FactoryTalk View ME Station. Options are:

- Runtime 15.0 Application (*.mer)
- Runtime 14.0 Application (*.mer)
- Runtime 13.0 Application (*.mer)
- Runtime 12.0 Application (*.mer)
- Runtime 11.0 Application (*.mer)
- Runtime 10.0 Application (*.mer)
- Runtime 9.0 Application (*.mer)
- Runtime 8.20 Application (*.mer)
- Runtime 8.10 Application (*.mer)
- Runtime 8.0 Application (*.mer)
- Runtime 7.0 Application (*.mer)
- Runtime 6.10 Application (*.mer)
- Runtime 6.0 Application (*.mer)
- Runtime 5.10 Application (*.mer)
- Runtime 5.0 Application (*.mer)

Multiple version support is useful for system designers and others who create and modify applications for different versions of FactoryTalk View ME Station on an ongoing basis. You can use the latest version of FactoryTalk View Studio on a single development computer to provide applications for terminals that use previous versions of FactoryTalk View ME Station.

To check which version of FactoryTalk View ME Station you are using

1. In FactoryTalk View ME Station, click **Terminal Settings**.
2. Click **System Information**.
3. Click **About FactoryTalk View ME Station**.

Create runtime application files for previous versions

When you create the runtime application file (with the file extension .mer), you can specify the version of FactoryTalk View ME Station for which to create the file. For example, if the application will run on a terminal that uses FactoryTalk View ME Station 6.0, you can specify that version for the .mer file.

If the application contains features that are not supported by the version you select, FactoryTalk View shows a validation report that lists the unsupported features. The runtime application file is not created. You must remove or turn off the unsupported features before you can create the runtime application file.

For information about creating runtime application files, see Create runtime applications on page 177.

The remainder of this appendix lists the features that are not supported in previous versions of FactoryTalk View ME Station. The tables also show how to remove or replace the unsupported features.

Features that are not supported in version 12.0 or earlier

These version 13.0 features are not supported in FactoryTalk View ME Station version 12.0 or earlier. The right column describes how to remove the features.

To remove this feature	Do this
Write the runtime recipe name to a tag after download	Remove the specified tag on the **General** tab of the RecipePlus editor.

Features that are not supported in version 9.0 or earlier

These version 10.0 features are not supported in FactoryTalk View ME Station version 9.0 or earlier. The right column describes how to remove the features.

To remove this feature	Do this
Variable message file	On the **General** tab of the **Local Message Display Properties** dialog box, clear the **Use variable message file** checkbox and click the **Browse** button to select a message file if needed.
Echo message	On the **General** tab of the **Local Message Display Properties** dialog box, clear the **Use echo message** checkbox.
Audit trail	In the **Startup** editor, clear the **Audit trail** checkbox.
CSV export	In the **CSV Export Setup** editor, clear the **Enable CSV Export** checkbox.
Audit trail list, audit trail detail, and clear audit trail graphic objects	Remove the graphic objects from graphic displays.

Features that are not supported in version 8.20 or earlier

These version 9.0 features are not supported in FactoryTalk View ME Station version 8.20 or earlier. The right column describes how to remove the features.

To remove this feature	Do this
Alarm banner, alarm list, and alarm status list objects with Use Alarm Identifier property enabled	Disable the **Use Alarm Identifier** function on the **General** tab of the object's **Properties** dialog box and the **Messages** tab of the **Alarm Setup** editor.
Hyperlink animation	Remove the hyperlink animation.
Extended tag properties	Remove the extended tag properties.
Multiple displays are selected to be shown at startup	Select only one or none graphic display to be shown at startup.

Features that are not supported in version 8.10 or earlier

These version 8.20 features are not supported in FactoryTalk View ME Station version 8.10 or earlier. The right column describes how to remove or replace the features.

To remove or replace this feature	Do this
Graphics object with electronic signature property enabled	Disable the electronic signature function on the **E-Signature** tab of the object's **Properties** dialog box.
Application with advanced inactivity actions property enabled	Disable the advanced inactivity actions on the **Inactivity** tab of the **Project Settings** dialog box.

Features that are not supported in version 7.0 or earlier

These version 8.0 features are not supported in FactoryTalk View ME Station version 7.0 or earlier. The right column describes how to remove or replace the features.

To remove or replace this feature	Do this
Add User/Group button	Delete the button object from graphic displays.
Delete User/Group button	Delete the button object from graphic displays.
Modify Group Membership button	Delete the button object from graphic displays.
Unlock User button	Delete the button object from graphic displays.
Enable User button	Delete the button object from graphic displays.
Disable User button	Delete the button object from graphic displays.
Login button	Clear the **Domain name** and **User name** boxes on the **General** tab of the **Login Button Properties** dialog box.
Password button	Clear the **Change any user password (Administrative)** option on the **General** tab of the **Password Button Properties** dialog box.
Change User Properties button	Delete the button object from graphic displays.

Features that are not supported in version 6.10 or earlier

These version 7.00 features are not supported in FactoryTalk View ME Station version 6.10 or earlier. The right column describes how to remove or replace the features.

To remove or replace this feature	Do this
ME Alarm Embedded Server Update Rate	Select **Match Alarm Update Rate** or the same number as Maximum update rate on the **Advance** tab of the **Alarm Setup** editor.
MAXIMUM DATA POINTS enlargement support	Change the Maximum data point value to be equal to or less than 300000 on the **Setup** tab of the **Data Log Models** editor.
Parameterized Macro button	Clear the **Use Variable Macro** checkbox on the **General** tab of the **Macro Button Properties** dialog box.
Parameterized Goto display button	Clear the **Use Variable Display** checkbox on the **General** tab of the **Goto Display Button Properties** dialog box.

To remove or replace this feature	Do this
Parameterized display position	Clear the **Use Variable Display Position** checkbox on the **General** tab of the **Goto Display Button Properties** dialog box.
Graphic Objects configured with Gradient Shading background	Change the Gradient Shading background style to **Solid** or **Transparent**. Delete Gradient animation from Graphics Objects animation or select other Fill Styles in color animation.
PNG image support	Delete PNG image object from graphic displays.

Feature that is not supported in version 5.10 or earlier

The version 6.0 feature is not supported in FactoryTalk View ME Station version 5.10 or earlier. The right column describes how to remove or replace the feature.

To remove or replace this feature	Do this
Graphics object with Shaded or Original animation	Delete the Shaded or Original animation from the Graphics object animation or select **Solid** in **Fill Style**.

Features that are not supported in version 5.0 or earlier

These version 5.10 features are not supported in FactoryTalk View ME Station version 5.0 or earlier. The right column describes how to remove or replace the features.

To remove or replace this feature	Do this
Tag or expression assigned to the **Close an On Top Display**, or **Close All On Top Displays** connections	Delete the tag or expression assigned on the **Display** tab of the **Global Connections** editor.
Tag or expression assigned to the **Parameter File**, or **Use Parameter List** connections	Delete the tag or expression assigned on the **Display** tab of the **Global Connections** editor.
Tag or expression assigned to one or more of the **Parameter Tag #1-10** connections	Delete the tag or expression on the **Display** tab of the **Global Connections** editor.
Parameter list property assigned in the **Properties** dialog box of a Display List Selector, Goto display button, or Logout button graphic object	In the object's **Properties** dialog box, clear the **Parameter list** assignment.
Parameter list property assigned to the display specified for auto logout in **Project Settings**	On the **Runtime** tab of the **Project Settings** dialog box, clear the **Parameter list** assignment.
Parameter list assigned to the **Initial graphic** property in the Startup editor	In the **Startup** editor, clear the **Parameter list assignment** for the **Initial graphic**.
Use variable minimum/maximum property assigned to **Numeric Input Cursor Point** or Numeric Input Enable object	On the **Numeric** tab of the object's **Properties** dialog box, clear the **Use variable minimum/maximum** property.
Caption on Keypad property assigned to **Numeric Input Cursor Point** object	On the **Numeric** tab of the object's **Properties** dialog box, set the **Caption on keypad/scratchpad** property to blank.
Minimum or **Maximum** connections assigned for **Numeric Input Cursor Point** or **Numeric Input Enable object**	On the **Connections** tab of the object's **Properties** dialog box, delete the tag or expression assigned to the **Minimum** or **Maximum** connection.

To remove or replace this feature	Do this
Use variable ramp or **Use variable limit** property set for the **Ramp** button object	On the **General** tab of the object's **Properties** dialog box, clear the **Use variable lower/upper limit** and **Use variable ramp value** properties.
Ramp or **Limit** connection set for the **Ramp** button object	On the **Connections** tab of the object's **Properties** dialog box, delete the tag or expression assigned to the **Ramp** or **Limit** connection.
The application is designated as a Compact Machine Edition application in **Project Settings**	On the **General** tab of the **Project Settings** dialog box, clear the **Compact Machine Edition application** checkbox.

Format an SD card using exFAT

This appendix describes:

- Saving files on an SD card
- How to format an SD card using exFAT

Saving files on an SD card

On a PanelView Plus 7 or PanelView Plus 6 terminal, when saving files that have the same file name extension, and whose names start with the same eight characters on an SD card:

- A maximum of 999 such files can be stored in a folder if the SD card is formatted using FAT32 (File Allocation Table 32).
- More than 999 such files can be stored in a folder if the SD card is formated using exFAT (Extended File Allocation Table). For example, when exporting CSV files (YYYY_MM_DD_HH_mm_ss-<FileLabel>-TYPE.csv), you may need to format the SD card using exFAT to save more files.

Format an SD card using exFAT

To format an SD card using exFAT

1. On the PanelView Plus 7 or PanelView Plus 6 terminal, open Control Panel.
2. Open Storage Manager.
3. In the **Storage Properties** dialog box, from the **Store Info** list, select the **USB SD Hard Drive**.
4. If * shows in the **Partitions** box, click **Dismount**. * disappears.
5. Click **Format** and then **Yes**.
6. In the **Format store succeeded** dialog box, click **OK**.
7. In the **Storage Properties** dialog box, click **New** to create one or multiple partitions.
8. Click **OK** to close the **Storage Properties** dialog box.

Once the SD card is successfully formatted, one or multiple **Storage Card** icons show in **My Device**.

Import and export alarm XML files

This appendix describes:

- How to create alarm XML files
- How to export, editing, and importing XML files
- The alarm XML file structure

About alarm XML files

XML is the Extensible Markup Language used to create documents with structured text information. It has a standardized format and structure. You can use XML to edit the elements and attributes needed to create an alarm setup file or to modify graphic displays. For information about working with graphics XML files, see Import and export graphics XML files on page 433.

Sometimes editing your alarm setup in an XML file is quicker than working in FactoryTalk View. For example, if you have a list of 100 tags to monitor for alarms, with multiple messages for each tag, you might prefer to enter all the information in a text editor, and then import the alarm setup information into FactoryTalk View.

Another example of using XML files is to export the alarm setup information you develop in one application, import the setup information to another FactoryTalk View application, and then modify the alarm setup as needed. Or, you could modify the information in the XML file before importing it.

For more information about XML, see the World Wide Web Consortium's webpage about XML at https://www.w3.org/XML

Create alarm XML files by exporting

The quickest way to create an XML file for your application's alarm setup is to export the data from FactoryTalk View. You can then open the XML file in Notepad, make your changes, and import the file back into FactoryTalk View.

Tip: The strings for the application's current language are exported to the XML file. To export strings for another language, reopen the application in the new language and repeat the XML export.

To export alarm information to an XML file

1. In the **Explorer** window, right-click the **Alarm Setup** editor and select **Import and Export**.
   The Alarm Import Export Wizard opens.
2. Follow the instructions in the wizard.
   For information about using the Alarm Import Export Wizard, see Help.

FactoryTalk View creates a file with the name you specify, in the location you specify.

Edit alarm XML files

We recommend that you use Notepad to edit your alarm XML files.

If you do not want to change a property, you don't need to include it in the XML file.

Save XML files in Notepad

Save XML files created or edited in Notepad using either UTF-8 or UTF-16 file format. Notepad's Unicode file type corresponds to the UTF-16 file format. For files containing strings in English or other Latin-based languages, UTF-8 is recommended, to reduce the size of the XML file. For other languages such as Chinese, Japanese, or Korean, UTF-16 is recommended.

The first line of every XML file contains XML version and encoding attributes. Make sure that 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 that the first line specifies encoding="UTF-16".

Test 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 shows an error message.

Import alarm XML files

You can import an alarm setup that has been created using an external programming tool or editor, or you can import an XML file that you originally exported from FactoryTalk View and then modified.

When you import an alarm setup, your existing alarm setup will be overwritten. Back up your application first using FactoryTalk View ME Application Manager. Or, you can save a copy of your existing alarm setup by exporting it to an XML file before you import the new one.

If the application is a PanelView Plus 7 Standard application or a Compact Machine Edition application (an application running on a PanelView Plus 6 Compact or PanelView Plus Compact terminal), only the number of alarms allowed by the application limit will be imported.

To import alarm information from an XML file

1. In the **Explorer** window, right-click the **Alarm Setup** editor and select **Import and Export**.
   The Alarm Import Export Wizard opens.
2. Follow the instructions in the wizard.
   For more information about using the Alarm Import Export Wizard, see Help.

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.

Alarm setup XML file structure

An alarm setup XML file is a FactoryTalk View XML document that describes the alarm setup for an application. The root element of the XML document is called **alarms**. It represents the **Alarm Setup** editor.

An XML document can contain only one root element. All other elements in the document must be contained or nested within the beginning and end markers of the root element.

In an XML document, the start of an element is marked by **<element name>**, and the end is marked by **</element name>**.

If the element contains no sub-elements, the end can be marked by **/>**. For example, **<**trigger id="T1" type="value" ack-all-value="0" **/>**.

The syntax for specifying an attribute for an element is **attribute="value"**. The attribute value must be enclosed in single or double quotes. You can specify multiple attributes for an element. For example, an alarm element can contain 12 possible attributes from the **Advanced** tab of the **Alarm Setup** editor.

For more information about alarm elements and their attributes, see Help.

Here is a sample structure for an alarm XML document.

Element	Description
<alarms>	Root element.
<alarm>	Contains attributes from the **Advanced** tab of the **Alarm Setup** editor, as well as the triggers and messages elements.
<triggers>	Contains a trigger element for each trigger on the **Triggers** tab of the **Alarm Setup** editor.
<trigger id="T1" />	Contains attributes for the first alarm trigger.
<trigger id="T2" />	Contains attributes for the second alarm trigger.
</triggers>	Indicates the end of the triggers element.
<messages>	Contains a message element for each message on the **Messages** tab of the **Alarm Setup** editor.
<message id="M1" />	Contains attributes for the first alarm message.
<message id="M2" />	Contains attributes for the second alarm message.
</messages>	Indicates the end of the messages element.
</alarm>	Indicates the end of the alarm element.
</alarms>	Indicates the end of the alarms element.

Import and export graphics XML files

This appendix describes:

- How to create graphics XML files
- How to export, edit, and import XML files
- The graphics XML file structure

About graphic XML files

Sometimes editing your display information in an XML file is quicker than working in FactoryTalk View. For example, if you have a list of 100 local messages to set up for a graphic display, you might prefer to enter all the information in a text editor, and then import the display information into FactoryTalk View.

For more information about XML, see the World Wide Web Consortium's webpage about XML at https://www.w3.org/XML.

For information about working with alarm XML files, see Import and export alarm XML files on page 430.

Create graphics XML files by exporting

The quickest way to create an XML file for your application's graphic displays is to export the data from FactoryTalk View. You can then open the XML file in Notepad, make your changes, and import the file back into FactoryTalk View.

Tip: The strings for the application's current language are exported to the XML file. To export strings for another language, reopen the application in the new language and repeat the XML export.

To export graphic display information to an XML file

1. In the **Explorer** window, right-click the **Displays** editor or the **Global Objects** editor.
2. Select **Import and Export**.
   The Graphics Import Export Wizard opens.
3. Follow the instructions in the wizard.
   For information about using the Graphics Import Export Wizard, see Help.

FactoryTalk View creates XML files for the selected graphic displays, in the location you specify.

FactoryTalk View also creates a file called BatchImport_*Application name*.xml, in the same location. You can use this file to import multiple displays at the same time. To import a different set of displays than you exported, edit the list of display names in the BatchImport_*Application name*.xml file.

Edit graphic XML files

We recommend that you use Notepad to edit your graphic XML files.

If you do not want to change a property, you don't need to include it in the XML file. When you import the file, if you select the option **Create new objects in the display**, properties that are not listed in the file are set to their default values. If you select the option **Update existing objects on the display**, only properties that are listed in the file are updated with imported information.

Tip: If you include attributes for an object whose name does not match one of those in the graphic display, the attributes for that object are not imported. Attributes for all other objects in the file whose names do match the ones in the graphic display are imported.

Save XML files in Notepad

Save XML files created or edited in Notepad using either UTF-8 or UTF-16 file format. Notepad's Unicode file type corresponds to the UTF-16 file format. For files containing strings in English or other Latin-based languages, UTF-8 is recommended, to reduce the size of the XML file. For other languages such as Chinese, Japanese, or Korean, UTF-16 is recommended.

The first line of every XML file contains XML version and encoding attributes. Make sure that 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 that the first line specifies encoding="UTF-16".

Test 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 shows an error message.

Import graphic XML files

You can import a graphic display or global object display that has been created using an external programming tool or editor, or you can import an XML file that you originally exported from FactoryTalk View and then modified.

Tip: When you import a graphic display or global object display, your existing display will be overwritten. Back up your application first, using FactoryTalk View ME Application Manager. Or, you can save a copy of your existing display by exporting it to an XML file before you import the new one.

If the application is a PanelView Plus 7 Standard or PanelView Plus 6 Compact application, only the number of displays allowed by the application limit will be imported.

You can import a single graphic or global object display XML file at a time, or import multiple displays. You can also choose whether to import new objects or update existing objects.

To import multiple displays, specify the names of the displays in the file BatchImport_*Application name*.xml. FactoryTalk View creates this file when you export multiple displays. For details, see Create graphics XML files by exporting on page 433.

To import display information from an XML file

1. In the **Explorer** window, right-click the **Displays** or **Global Objects** editor.
2. Select **Import and Export**.
   The Graphics Import Export Wizard opens.
3. Follow the instructions in the wizard.
4. For more information about using the Graphics Import Export Wizard, see Help.

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.

Graphic display XML file structure

A graphic display XML file is a FactoryTalk View XML document that describes the objects and settings for a graphic display. The root element of the XML document is called **gfx**. It represents the graphic display.

An XML document can contain only one root element. All other elements must be contained within the beginning and end markers of the root element.

In an XML document, the start of an element is marked by **<element name>**, and the end is marked by **</element name>**.

If an element contains no sub-elements, the end can be marked by **/>**. For example, **<**trigger id="T1" type="value" ack-all-value="0" **/>**.

The syntax for specifying an attribute for an element is **attribute="value"**. The attribute value must be enclosed in single or double quotes. You can specify multiple attributes for an element. For example, the caption element contains 13 possible attributes.

Elements for group objects begin with **<group name>**, and end with **</group name>**.

The <*group*> element contains all the elements and attributes for each object in the group.

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 of 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.

Element	Description
<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 of 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.

Element	Description
</vbaItem>	Indicates the end of the vbaItem element.
</vbaProject>	Indicates the end of the vbaProject element.
</gfx>	Indicates the end of the gfx element.

Convert PanelBuilder 1400e applications

This appendix describes:

- Terms that are different in PanelBuilder 1400e and FactoryTalk View
- Steps for converting PanelBuilder 1400e applications
- Names of equivalent graphic objects in the two products
- PanelBuilder 1400e graphic objects that are not supported in FactoryTalk View
- PanelBuilder 1400e settings and controls that are not supported in FactoryTalk View
- How communications are converted and which PanelBuilder 1400e communication protocols are not supported in FactoryTalk View
- Convert PanelBuilder 1400e Remote I/O communications
- PanelBuilder 1400e graphic object features that are not supported in FactoryTalk View, with information about how to achieve the same result when possible
- Convert PanelBuilder 1400e expressions

PanelBuilder 1400e applications are applications that you create using PanelBuilder 1400e Configuration Software for Windows. For information about converting applications from PanelBuilder or PanelBuilder32, see Convert PanelBuilder and PanelBuilder32 applications on page 451.

Tip: You can convert PanelView 1200 applications to PanelBuilder 1400e applications, and then convert the PanelBuilder 1400e applications to FactoryTalk View ME applications.

Different terms in PanelBuilder 1400e and FactoryTalk View

These terms are different in PanelBuilder 1400e and FactoryTalk View.

PanelBuilder 1400e term	FactoryTalk View term
screen	display, graphic display
Optional Keypad Write Expression	Optional Expression
programmable controller	data source
control	connection

In FactoryTalk View, the data source can be memory or a device such as a programmable controller or an OPC server. FactoryTalk View writes values to and reads values from the data source. The data source is configured to exchange information (in the form of numeric or string values) between FactoryTalk View and the machine that your application is controlling. The general term data source is used unless specifically discussing a programmable controller.

Convert PanelBuilder 1400e applications

To convert a PanelBuilder 1400e application

1. Prepare the application in PanelBuilder 1400e, and then convert the application file, as described in the next section.
2. Specify additional project settings, as described on About project settings on page 57. For example, if you want the application to have a border around its graphic displays, or to use a title bar, you can specify these options in the **Project Settings** editor.

   Tip: We recommend that you use the **Project Settings** editor to change the project window size, rather than using the **Convert to new window size** option in the Machine Edition Import Wizard.

3. If you use the **Convert to new window size** option in the Machine Edition Import Wizard, check the position of the graphic objects in each display.
4. Set up communications and edit tags that don't convert directly.
   For more information, see Convert non-RIO communications. on page 444
5. Set up graphic object features that don't convert directly.
   For more information, see Unsupported graphic object features on page 446.
6. Check each expression that you used in PanelBuilder 1400e.
   For more information, see Convert expressions on page 448.
7. If you are going to use a printer at runtime, set it up for Ethernet or USB printing. Adjust the printer settings on the PanelView Plus or PanelView Plus CE terminal.
   For information about setting up printers on the terminal, see the *PanelView Plus Terminals User Manual*.

Convert PanelBuilder 1400e application files

Follow these steps to convert a PanelBuilder 1400e application file with the extension .pvc to a FactoryTalk View application file with the extension .med. The original PanelBuilder 1400e application file is not modified by the conversion.

For information about converting RIO applications, see Convert RIO communications on page 445.

Before converting a PanelBuilder 1400e application

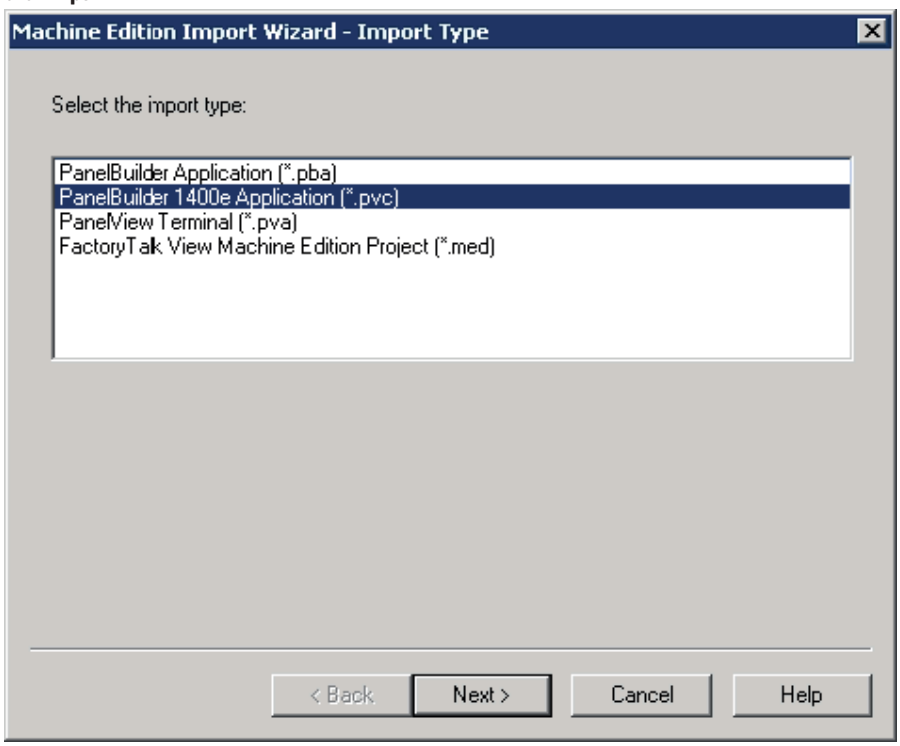
1. Delete the Pass-Through file assignment. FactoryTalk View Studio does not support pass-through file transfers.
2. Make sure that the block transfer file numbers are sequential without gaps. If necessary, renumber the block transfer file numbers so there are no missing numbers. Tag addresses in the application will change automatically to match the new number.
3. Save the application.

You can convert the PanelBuilder 1400e application when you open FactoryTalk View Studio, or once FactoryTalk View Studio is already open.

Convert a PanelBuilder 1400e application when you open FactoryTalk View Studio

To convert a PanelBuilder 1400e application when you open FactoryTalk View Studio

1. Open FactoryTalk View Studio.
2. On the **New** tab, in the **Application name** box, type a name for your converted application, up to 32 characters long.
3. If desired, type a description of the application.
   If the PanelBuilder 1400e application contains an Application File Comment, the Application File Comment will overwrite the description you type here. You can add or change the description later, as described on View application properties on page 61.
4. Specify a language for the converted application. For information about using different languages, see Set up language switching on page 159.
5. Click **Import**.



6. Follow the steps in the Machine Edition Import Wizard.
   For details about the options in the Machine Edition Import Wizard, see Help.

When you complete the steps of the wizard, FactoryTalk View Studio converts the PanelBuilder 1400e application, creates the converted application's folders and files, and then shows the converted application in the **Explorer** window in FactoryTalk View Studio.

If there are any messages about conversion, they are shown automatically in the **Project Status** dialog box.

By default, the converted application is created in **\Users\Public\Public Documents\RSView Enterprise\ME\HMI projects***Application Name*

where *Application Name* is the name of your application specified in step 2.

Conversion messages are saved in a file called *Convert.log* in the **HMI projects** folder.

Convert a PanelBuilder 1400e application when FactoryTalk View Studio is already open

To convert a PanelBuilder 1400e application when FactoryTalk View Studio is already open

1. From the **File** menu, select **New Application**, or click the **New Application** tool .
   If an application is already open, FactoryTalk View Studio asks you whether to close the application that is currently open. Click **Yes**.
2. Follow steps 2 through 6 in the previous procedure.

Equivalent graphic objects in PanelBuilder 1400e and FactoryTalk View

This section describes graphic objects that are equivalent in PanelBuilder 1400e and FactoryTalk View, but have different names in the two products.

This PanelBuilder 1400e object	Is converted to this FactoryTalk View object	Notes
Increment Value Button	Ramp button	During conversion, the button is set up to increment.
Decrement Value Button	Ramp button	During conversion, the button is set up to decrement.
Increment Value Button with Display	Ramp button and numeric display	The Increment Value Button with Display is divided into two separate FactoryTalk View objects.
Decrement Value Button with Display	Ramp button and numeric display	The Decrement Value Button with Display is divided into two separate FactoryTalk View objects.
ASCII Input (small and large)	String Input Enable button	
Numeric Entry Keypad (small and large)	Numeric input enable button	
Screen List Selector's list	Display list selector	The PanelBuilder 1400e Screen List Selector is divided into four separate FactoryTalk View graphic objects.
Screen List Selector's Enter Key	Enter button	
Screen List Selector's Down Cursor	Move down button	
Screen List Selector's Up Cursor	Move up button	
Control List Selector's list	Control list selector	The PanelBuilder 1400e Control List Selector is divided into four separate FactoryTalk View graphic objects.
Control List Selector's Enter Key	Enter button	
Control List Selector's Down Cursor	Move down button	
Control List Selector's Up Cursor	Move up button	
Screen Select Keypad (small and large)	Display list selector	Specify the graphic displays that the display list selector can open.

This PanelBuilder 1400e object	Is converted to this FactoryTalk View object	Notes
Screen Keypad Enable Button	Display list selector	Specify the graphic displays that the display list selector can open.
Goto Screen Button	Goto display button	
Return to Previous Screen Button	Return to display button	
ASCII Display	String display	
Numeric Keypad Enable Button	Numeric input enable button	
Normally Open Momentary Push Button	Momentary push button	During conversion, the button is set up to be normally open.
Normally Closed Momentary Push Button	Momentary push button	During conversion, the button is set up to be normally closed.
Screen Print Button	Display print button	
Alarm History Sort By Time/Sort By Value Button	Sort alarms button	
Alarm Status Reset Qty/Time Button	Reset alarm status button	
Alarm Panel	Alarm banner	
Single Line Alarm Window	Alarm banner	
Alarm Status Screen	Alarm status list	
Clear All Button	Clear alarm history button	
Print Button (in Alarm History screen)	Print alarm history button	
Print Button (in Alarm Status screen)	Print alarm status button	
Alarm Status Button/Alarm History Button	Goto display button	
Exit Button	Close display button	
Alarm History List	Alarm list	
Display Mode Button	Alarm status mode button	
Time Display	Time and date display	During conversion, the display is set up to show the time only. The PanelBuilder 1400e time format is not converted. For details about the FactoryTalk View time format, see Time, date, and number formats for graphic objects on page 337.
Date Display	Time and date display	During conversion, the display is set up to show the date only. The PanelBuilder 1400e date format is not converted. For details about the FactoryTalk View date format, see Time, date, and number formats for graphic objects on page 337.
Arc (with solid fill style)	Arc (with solid back style) and line	The line graphic object is added because the solid FactoryTalk View arc shape does

This PanelBuilder 1400e object	Is converted to this FactoryTalk View object	Notes
		not have a line between the two points of the arc.  FactoryTalk View arc  FactoryTalk View arc with line

Unsupported PanelBuilder 1400e objects

These PanelBuilder 1400e objects are not supported in FactoryTalk View:

- Scrolling List (includes Cursor List, Multistate Indicator Object List, Local Message Object List, Numeric Data Display Object List)
- Set Bit Cursor Point

Unsupported PanelBuilder 1400e settings and controls

This section describes PanelBuilder 1400e settings and controls that are not used in FactoryTalk View.

Controls for transferring runtime application files

PanelBuilder 1400e uses these optional controls for transferring files to the runtime terminal:

- Transfer Inhibit control
- Transfer Request control
- Transfer Status control

These controls are not necessary in FactoryTalk View because the ME Transfer Utility allows you to transfer the runtime project file while running a project on the runtime terminal.

Settings and controls for alarms

FactoryTalk View does not use these PanelBuilder 1400e features and settings to manage alarms:

- Alarm relays
- Bit alarm acknowledgement
- Remote Alarm Operation Hold Time. The PanelBuilder 1400e Remote Alarm Ack Control Hold Time will be used for all alarm hold times. You can change the hold time on the **Advanced** tab in the FactoryTalk View **Alarm Setup** editor.
- Remote Alarm Control Delay Time. In FactoryTalk View, if an **Ack** connection is assigned, when an alarm is acknowledged the **Ack** connection is set immediately, without waiting for a delay time.

FactoryTalk View does not use these PanelBuilder 1400e controls to manage alarms:

- PLC Controlled Relay control
- PLC Controlled Audio control
- Acknowledge to PLC control (if the Alarm Acknowledge to PLC option is set to Bit)

Invalid characters in screen names

Characters in PanelBuilder 1400e screen names that are not supported in FactoryTalk View are replaced with the underscore character.

PanelBuilder 1400e screen security settings

PanelBuilder 1400e screen security settings are not converted, because FactoryTalk View uses a different method to assign security to graphic displays. For information about setting up security in FactoryTalk View, see Set up security on page 142.

Block tags

Block tags are not supported in FactoryTalk View. Block tags that are monitored for alarms in your PanelBuilder 1400e application are converted to bit arrays. For information about monitoring bit arrays for alarm conditions, see Set up alarms on page 95.

Convert non-RIO communications

This section describes how communications that do not use Remote I/O (RIO) are converted.

FactoryTalk View does not use nodes for communications. Nodes are converted to RSLinx topics. Topics are then converted into device shortcuts to run with FactoryTalk Linx. You must have both RSLinx Classic and FactoryTalk Linx installed to make this two-step conversion.

Tags are converted to HMI device tags and RSLinx aliases. The **Unsolicited_Msgs** node is not converted.

If you import an application multiple times, delete the device shortcuts in FactoryTalk Linx before reimporting. Otherwise, multiple unused device shortcuts will be created in FactoryTalk Linx.

For more information about setting up communications, see Set up communications on page 63.

Unsupported tag data types

These tag data types are not supported in FactoryTalk View:

- Bit Position
- 1-BCD, 2-BCD, 5-BCD, 6-BCD, 7-BCD, 8-BCD
- BIN3, BIN4, BIN6, BIN8 (used with Modbus communications)

Tags that use these data types are converted to analog HMI tags with the Default data type. The Default data type uses floating point values.

For **Bit Position** data types, use the bitwise expression operators to show data that does not reference supported lengths. For information about using bitwise expression operators, see Bitwise operators on page 358.

Unsupported initial values

Device tags in FactoryTalk View do not use initial values. Memory tags are converted with their initial values.

Convert RIO communications from PanelBuilder 1400e to FactoryTalk View

You can use Remote I/O (RIO) communications on the PanelView Plus, and PanelView Plus CE runtime platforms.

Tip: For PanelView Plus 7 and PanelView Plus 6 terminals, you need a third-party RIO to Ethernet gateway module for RIO communications. For more information, see Knowledgebase Document ID: QA17542 - 3rd Party Remote I/O solution for the PanelView Plus 6 and PanelView Plus 7.

RIO communications are not supported for the applications that run on a personal computer. However, you can test run your RIO applications on the development computer.

To convert an RIO application from PanelBuilder 1400e to FactoryTalk View

1. Convert the application.
   For more information, see Convert PanelBuilder 1400e application files on page 439.
2. Open the FactoryTalk Linx data server, and then double-click **Communication Setup**.
3. In the **Communication Setup** editor, add an RIO driver.
   - For PanelView Plus 400 and 600 terminals, use the 2711P-RN1 driver.
   - For all other PanelView Plus or PanelView Plus CE terminals, use the 2711P-RN6 driver.

   For information about adding drivers in FactoryTalk Linx, see FactoryTalk Linx Help.
4. Expand the RIO tree, right-click RIO Data, and then click **Configure RIO**.
5. In the **RIO Configuration** dialog box, right-click RIO, and then click **Import**.
6. Browse to the location of the RIO configuration file.
   The file is saved in the root of the application's directory.
7. In the **Communications Setup** editor, create a device shortcut that points to the RIO data device.
   For information about creating a device shortcut, see RSLinx Help.
8. Apply the shortcut to the RIO driver.
9. Correct any invalid RIO configurations. Invalid RIO configurations are highlighted with red "x" icons.

   Tip: If red "x" icons appear after you import the RIO .xml file, you can fix block tags in the **Communication Setup** editor in FactoryTalk View. The length of block tags must be the same for Read and Write pairs sharing the same rack, group, and slot.

10. Create an alias for any data that is not a 16-bit integer or bit.
11. Save the converted RIO application.

IMPORTANT: RIO configurations are not saved with the application when you exit FactoryTalk View Studio. However, they are backed up with the application in the FactoryTalk View ME Application Manager.
For information about handling multiple applications with different RIO settings, see the Rockwell Automation Knowledgebase.
For information about using the FactoryTalk View ME Application Manager, see Rename, copy, delete, back up, and restore applications on page 56.

Unsupported PanelBuilder 1400e RIO tags

When a PanelBuilder 1400e RIO application is converted into a FactoryTalk View ME application, the Machine Edition Import Wizard converts all the valid tags in the imported applications into HMI device tags. Tags that the Wizard is not able to convert into device tags are converted into HMI memory tags, and an error is logged for each one, except for Block tags. For tags of tag type Block, a log file containing an equivalent direct reference will be created instead.

Unsupported tags include tags that in the PanelBuilder 1400e configuration have:

- A blank address
- A data type of 1 Digit BCD, 2 Digit BCD, 3 Digit BCD, 5 Digit BCD, 6 Digit BCD, or 7 Digit BCD
- A data type of Bit Position, but an address that does not reference a single bit
- A data type of Default, Unsigned Integer, Signed Integer, Long Integer, Float, 4 Digit BCD, 8 Digit BCD, or String, but an address whose assigned bit offset is not 0
- A tag type of Block
- An invalid PanelBuilder 1400e block transfer or I/O address

Unsupported PanelBuilder 1400e graphic object features

This section describes the features of PanelBuilder 1400e graphic objects that are not supported in FactoryTalk View. The Notes column provides additional information and describes methods for achieving the same result when possible.

Graphic object	Unsupported feature in FactoryTalk View	Notes
Image text, arc, ellipse, line, panel, rectangle, wedge	Blinking wallpaper objects	If you want an object to blink at runtime, unlock the wallpaper. In FactoryTalk View, all of the listed objects except images use color animation to blink. For details, see Set up color animation on page 344. Color images do not blink. Monochrome images use the **Blink** property to blink.
Numeric Display	Polarity	If a PanelBuilder 1400e application was set up with the Polarity control requiring a negative number to show the minus sign, the numeric display will not work properly after the application is converted to FactoryTalk View ME.
Numeric Input Cursor Point, Numeric Data Display	Fixed Position and PLC Controlled decimal display options	Use an expression to achieve the same result. Assign the expression to the object's **Value** connection. For information about expressions, see Use expressions on page 352.
Numeric Input Cursor Point, Numeric Keypad Enable Button, Numeric Keypad	**PLC Controlled** and **Decimal Key Controlled** input options	Objects are converted with the **Decimal Point** property set to **Implicit**.

Graphic object	Unsupported feature in FactoryTalk View	Notes
Numeric Input Cursor Point	Retain Cursor on Cancel	The numeric input cursor point retains focus when the operator cancels entering a numeric value.
Maintained push Button, Multistate Push Button, Control List Selector	Initial state values	If you want to set these objects' states on application startup, create a macro to set the appropriate tag values for the objects' connections. For information about macros, see Use macros on page 421. Assign the macro in the **Startup** editor. For details, see Help.
Trend	Blinking pens Date labels on the X-Axis Background screen plotting	The date is shown in the title. You can plot tag values in the background by assigning the tags to a data log model. Tags set up for background screen plotting are automatically assigned to a data log model on conversion. However, data log models do not plot expression values. Therefore, expressions set up for background screen plotting are not converted. For information about data logging, see Set up data logging on page 387.
All objects	PanelBuilder 1400e object name	Object names are replaced with the FactoryTalk View default object names. The PanelBuilder 1400e object name is used for the object's description. You can view and edit the name and description in the **Property Panel**. For details, see Help.
All objects	Caption and image placement	FactoryTalk View supports one, three, or nine positions for captions and images, depending on the type of object. On conversion, captions and images are positioned using the closest match. Therefore some captions might overlap images, some captions might be truncated, and some images might be clipped to fit the object.
All objects	Multiple image labels	FactoryTalk View supports one image label per object or state. If a PanelBuilder 1400e object is set up to use multiple

Graphic object	Unsupported feature in FactoryTalk View	Notes
		image labels, only the top-left image is converted.

Convert expressions

Some PanelBuilder 1400e expression syntax is not supported in FactoryTalk View. Expressions are converted without modification, and then turned off by placing warning text at the beginning of the first line of the expression. In addition, exclamation marks (!) are placed at the beginning of each subsequent line of the expression. Warming text is also placed in expressions assigned to alarm triggers in the **Alarm Setup** editor. To turn on the expression, you must remove the warning text and exclamation marks, and revise the syntax if necessary.

The maximum expression length in FactoryTalk View is 16,000 characters. If a PanelBuilder 1400e expression contains more than 16,000 characters, the excess characters are not converted.

Some PanelBuilder 1400e objects support both tags and expressions. For these objects, if the text assigned to a connection could be valid syntax for both a tag and an expression, the connection is treated as an expression, and is therefore turned off.

For example, N20-0_String_64 could be the name of a tag, or it could be an expression that subtracts "0_String_64" from the tag "N20." The text would be converted as an expression, and turned off.

Turn on an expression

To turn on an expression

1. Select the object containing the expression.
2. Open the **Property Panel**, and then click the **Connections** tab.
3. In the **Exprn** column, click the **Browse** button beside the expression to turn on.
4. In the **Expression** editor, delete the warning text and exclamation marks.
5. Revise the expression, if necessary, using the tables in the following three sections as guides.
6. Click **Check Syntax**.

For more information about using the Expression editor, see Use expressions on page 352 or Help.

Equivalent expression syntax

This table describes FactoryTalk View expression syntax that is equivalent to PanelBuilder 1400e syntax. When you edit the converted expressions, replace the PanelBuilder 1400e syntax with the FactoryTalk View equivalent.

Syntax that is not listed in this table or in the next section is okay the way it is.

Type of expression component or operator	PanelBuilder 1400e syntax	FactoryTalk View syntax
Comment	REM or '	!
Line continuation	_ (underscore)	Not needed.
Equality	(=)	EQ or ==

Type of expression component or operator	PanelBuilder 1400e syntax	FactoryTalk View syntax
Bitwise Not	Not	~ (tilde)
Bitwise And	And	&
Bitwise Or	Or	| (pipe)
Bitwise XOr	XOr	^
If both operands are Byte, Integer, Long, Variant, or any combination of these data types, use the FactoryTalk View syntax. For other data types, no change is needed.		

Unsupported expression syntax

This table describes the PanelBuilder 1400e expression syntax that is not supported in FactoryTalk View with information about how to achieve the same result where possible.

Type of expression component or operator	PanelBuilder 1400e syntax	Equivalent FactoryTalk View syntax (if any)
Exit statement	Exit	
Local variables	DIM *varname* AS ... *varname* =	
Integer division	\	(*x* - (*x* MOD *y*))/*y*
Endif	If then endif If then else endif	If then else 0 If then else
Select case	Select Case Case1...CaseN CaseElse EndSelect	Use nested if-then-else.
Logical Xor (if one or both operands are Boolean or Single data types)	Xor	NOT ((*x* AND *y*) OR NOT (*x* OR *y*))

Order of precedence

The order of precedence is slightly different in FactoryTalk View. Check your expressions to make sure the result is what you intend.

PanelBuilder 1400e order of precedence	FactoryTalk View order of precedence
( )	( )
- (negation)	NOT, ~ (tilde)
*, / (floating point division)	*, /, MOD, %, **, AND, &&, &, >>, <<
\ (integer division)	+, -, OR, ||, |, ^

PanelBuilder 1400e order of precedence	FactoryTalk View order of precedence
MOD	EQ, ==, NE, <>, LT, <, GT, >, LE, <=, GE, >=
+, - (subtraction)	
=, <>, <, >, <=, >=	
Not	
And	
Or	
Xor	

For more information about the order of precedence, see Evaluation order of operators on page 360.

Convert PanelBuilder and PanelBuilder32 applications

This appendix describes:

- Terms that are different in PanelBuilder and FactoryTalk View
- Steps for converting PanelBuilder applications
- Names of equivalent graphic objects in PanelBuilder and FactoryTalk View
- PanelBuilder graphic objects that are not supported in FactoryTalk View
- PanelBuilder settings and controls that are not supported in FactoryTalk View
- How communications are converted and which PanelBuilder communication protocols are not supported in FactoryTalk View
- Convert PanelBuilder Remote I/O communications
- PanelBuilder graphic object features that are not supported in FactoryTalk View, with information about how to achieve the same result when possible

This appendix uses the term PanelBuilder to refer to both PanelBuilder and PanelBuilder32 features.

For information about converting applications from PanelBuilder 1400e, see Convert PanelBuilder 1400e applications on page 438.

Different terms in PanelBuilder and FactoryTalk View

These terms are different in PanelBuilder and FactoryTalk View.

PanelBuilder term	FactoryTalk View term
screen	display, graphic display
programmable controller	data source
control	connection

In FactoryTalk View, the data source can be memory or a device such as a programmable controller or an OPC server. FactoryTalk View writes values to and reads values from the data source. The data source is configured to exchange information (in the form of numeric or string values) between FactoryTalk View and the machine that your application is controlling. The general term data source is used unless specifically discussing a programmable controller.

Convert PanelBuilder applications

To convert a PanelBuilder application

1. Convert the application file, as described in the next section.
2. Specify additional project settings, as described on About project settings on page 57. For example, if you want the application to have a border around its graphic displays, or to use a title bar, you can specify these options in the **Project Settings** editor.
3. If you select **Convert to new window size**, check the position of the graphic objects in each display.
4. Set up communications and edit tags that don't convert directly.
   For more information, see Convert non-RIO communications on page 444.
5. Set up graphic object features that don't convert directly.
   For more information, see Unsupported graphic object features on page 458.

Convert PanelBuilder application files

Follow these steps to convert a PanelBuilder application file with the extension .pba or .pva to a FactoryTalk View application file with the extension .med. The original PanelBuilder application file is not modified by the conversion.

Before converting a PanelBuilder application

The following are supported in PanelBuilder but not in FactoryTalk View Studio:

- Semicolons (;) in tag addresses
- Dashes (-) in tag names

Before converting the PanelBuilder application, in the PanelBuilder Tag Editor:

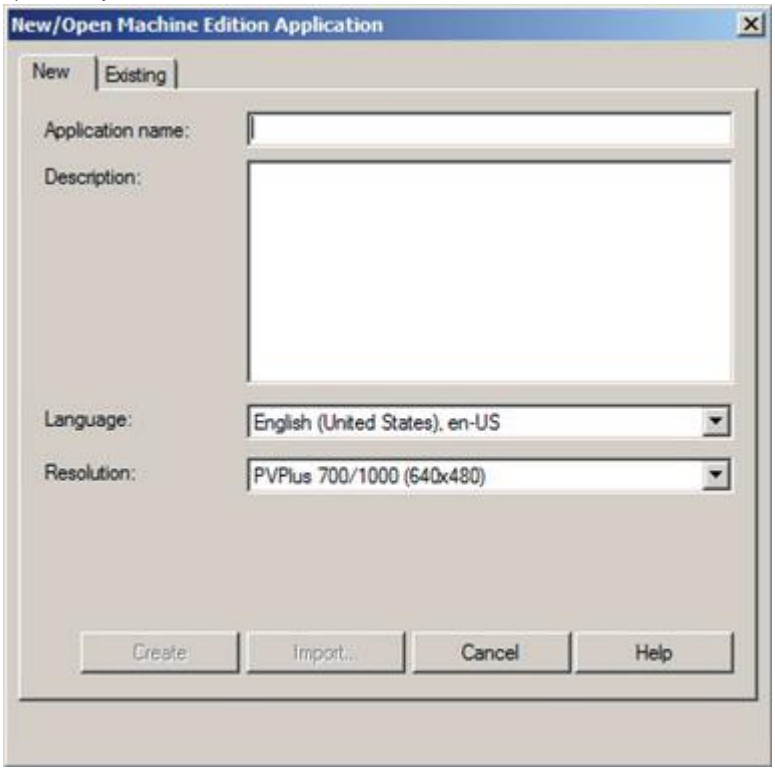
1. Change the semicolons to colons (:).
2. Locate any tags whose names contain dashes and duplicate the tags. Then rename the tags without the dash, or replace the dash with an underscore (_). Once the tags have been renamed, use the Tag Search feature to find the graphic objects using the original tag names and edit the objects to replace the old tag names with the new ones.

You can convert the PanelBuilder application when you open FactoryTalk View Studio or once FactoryTalk View Studio is already open.

Convert a PanelBuilder application when you open FactoryTalk View Studio

To convert a PanelBuilder application when you open FactoryTalk View Studio

1. Open FactoryTalk View Studio.



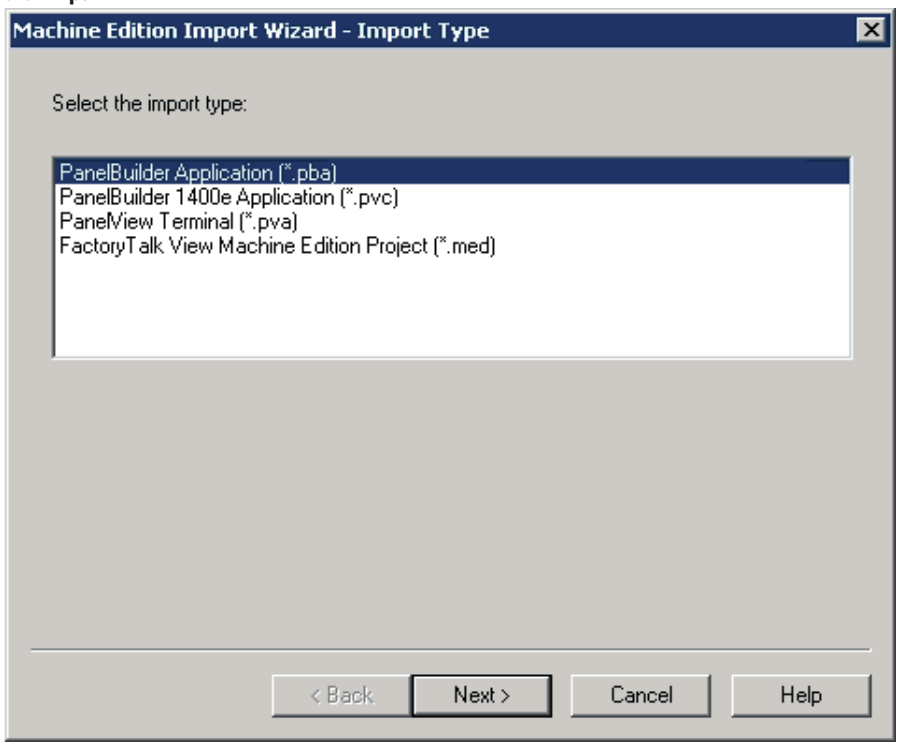
2. In the **New** tab, in the **Application name** box, type a name for your converted application, up to 32 characters long.

3. If desired, type a description of the application.

   If the PanelBuilder application contains an Application Description, the Application Description will overwrite the description you type here. You can add or change the description later, as described in View application properties on page 61.

   Specify the last language that was used to edit the application. This will be used for the converted application. You can only import one language for your application, even if the original application uses multiple languages. The imported language will be the last language used to edit the application. For information about using different languages, see Set up language switching on page 159.

4. Click **Import**.



5. Follow the steps in the Machine Edition Import Wizard.

   For details about the options in the Machine Edition Import Wizard, see Help.

When you complete the steps of the wizard, FactoryTalk View Studio converts the PanelBuilder application, creates the converted application's folders and files, and then shows the converted application in the **Explorer** window in FactoryTalk View Studio.

If there are any messages about conversion, they are shown automatically in the **Project Status** dialog box.

By default, the converted application is created in **\Users\Public\Public Documents\RSView Enterprise\ME\HMI projects***Application Name*

where *Application Name* is the name of your application specified in step 2.

Conversion messages are saved in a file called *Convert.log* in the **HMI projects** folder.

Convert a PanelBuilder application when FactoryTalk View Studio is already open

To convert a PanelBuilder application when FactoryTalk View Studio is already open

1. From the **File** menu, select **New Application**, or click the **New Application** tool .
   If an application is already open, FactoryTalk View Studio asks you whether to close the application that is currently open. Click **Yes**.
2. Follow steps 2 through 5 in the previous procedure.

Equivalent graphic objects in PanelBuilder and FactoryTalk View

This section describes graphic objects that are equivalent in PanelBuilder and FactoryTalk View, but have different names in the two products.

This PanelBuilder object	Is converted to this FactoryTalk View object	Notes
Numeric Entry Keypad Enable Button	Numeric input enable button	
Numeric Entry Cursor Point	Numeric input enable button	
Increment/Decrement Entry Button	Numeric input enable button	The Numeric Input Enable button is set up to work as a Ramp button, using the Fine Step value. The Coarse Step value is not converted.
ASCII Entry Keypad Enable button	String Input Enable button	The Show Current String on ASCII Scratchpad setting is not converted. The pop-up scratchpad or keyboard is always blank when opened.
ASCII Entry Cursor Point	String Input Enable button	The Show Current String on ASCII Scratchpad setting is not converted. The pop-up scratchpad or keyboard is always blank when opened.
Message Display	Multistate indicator	
Numeric Data Display	Text	The text object contains a numeric embedded variable that shows the read tag. If the original numeric data display object does not contain text, manually create a numeric display object with the desired properties in FactoryTalk View.
Connected Line	Polyline	
Circle	Ellipse	The ellipse has a circular shape.
Freeform	Freehand	
Screen List Selector	Display list selector	
Goto Screen Button	Goto display button	

This PanelBuilder object	Is converted to this FactoryTalk View object	Notes
Return Screen Button	Return to display button	
New Password Button	Password button	
Print Alarm List Button	Print alarm history button	
Clear Alarm List Button	Clear alarm history button	

Unsupported PanelBuilder objects

These PanelBuilder objects are not supported in FactoryTalk View:

- Print Only Object
- Circular Scale
- Scrolling Text
- Print Alarm Button
- Horn Silence Button
- Lamp/Horn Test Button
- Select Operator Button
- Enable/Disable Security Button
- Verify Password Button. The FactoryTalk View Password button opens a dialog box that allows the user to type and verify a new password.

Unsupported PanelBuilder settings and controls

This section describes PanelBuilder settings and controls that are not used in FactoryTalk View.

Settings and controls for alarms

FactoryTalk View does not use these PanelBuilder features and settings to manage alarms:

- Ack setting for alarm messages - in FactoryTalk View, all alarms can be acknowledged
- Bit alarm acknowledgement

FactoryTalk View does not use these PanelBuilder controls to manage alarms:

- Remote Ack All Handshake Tag
- Remote Clear All Alarm Tag
- Remote Clear All Alarm Handshake Tag

Invalid characters in screen names and tag names

Characters in PanelBuilder screen names and tag names that are not supported in FactoryTalk View are replaced with the underscore character.

Time and date

PanelBuilder time and date formats are not converted. For details about FactoryTalk View time and date formats, see Time, date, and number formats for graphic objects on page 337.

External fonts

PanelBuilder external fonts are not converted. When you convert your application, you can specify the font to use instead. For details, see Help for the Machine Edition Import Wizard.

PanelBuilder screen security settings

PanelBuilder screen security settings are not converted, because FactoryTalk View uses a different method to assign security to graphic displays. For information about setting up security in FactoryTalk View, see Set up security on page 142.

Power-up options

These PanelBuilder power-up options are not imported into FactoryTalk View:

- Write Last Terminal State to Controller
- Display Last User Screen
- Use Terminal Presets

Convert non-RIO communications

This section describes how communications that do not use Remote I/O (RIO) are converted.

FactoryTalk View does not use nodes for communications. Nodes are converted to RSLinx topics. Topics are then converted into device shortcuts to run with FactoryTalk Linx. You must have both RSLinx Classic and FactoryTalk Linx installed to make this two-step conversion.

Tags are converted to HMI device tags and RSLinx aliases. The **Unsolicited_Msgs** node is not converted.

If you import an application multiple times, delete the device shortcuts in FactoryTalk Linx before reimporting. Otherwise, multiple unused device shortcuts will be created in FactoryTalk Linx.

For more information about setting up communications, see Set up communications on page 63.

Tags for unsupported communication protocols

Some communication protocols are not supported in FactoryTalk View. For example, DH+™ communications that use the AutoMax node type are not supported. FactoryTalk View does not support CIP and Assembly Object addressing (used in PanelBuilder32 Ethernet communications).

Tags that use unsupported communication protocols are converted to HMI memory tags. Once you have set up communications for your converted application, change the memory tags to device tags that point to the correct addresses. All other imported tags are converted to HMI device tags.

For information about editing HMI tags, see Use HMI tags on page 79. For more information about which communication protocols are not supported, see Help or the Rockwell Automation Knowledgebase.

Bit array tags

You can monitor bit arrays for alarm conditions in FactoryTalk View, but you can't assign bit arrays to most graphic objects or write to bit arrays. (The only exception is the piloted control list selector object. For this object, you can assign a bit array tag to the **Visible States** connection.)

All bit array tags in your PanelBuilder application are converted to HMI memory tags.

For information about monitoring bit arrays for alarm conditions, see Set up alarms on page 95. For information about editing HMI tags, see Use HMI tags on page 79. For information about the piloted control list selector, see Help.

Convert RIO communications from PanelBuilder to FactoryTalk View

Remote I/O (RIO) communications are not supported for applications that will run on personal computers. You can use RIO communications on the PanelView Plus and PanelView Plus CE runtime platforms.

Supported PanelBuilder tags are converted to HMI device tags and RSLinx aliases. For information about unsupported PanelBuilder tags, see Unsupported PanelBuilder RIO tags on page 457.

To convert an RIO application from PanelBuilder to FactoryTalk View

1. Convert the application, as described on Convert PanelBuilder application files on page 452.
2. Open the FactoryTalk Linx data server, and then double-click **Communication Setup**.
3. In the **Communication Setup** editor, add an RIO driver.
   - For PanelView Plus 400 and 600 terminals, use the 2711P-RN1 driver.
   - For all other PanelView Plus or PanelView Plus CE terminals, use the 2711P-RN6 driver.

     For information about adding drivers in FactoryTalk Linx, see FactoryTalk Linx Help.
4. Expand the RIO tree, right-click RIO Data, and then click **Configure RIO**.
5. In the **RIO Configuration** dialog box, right-click RIO, and then click **Import**.
6. Browse to the location of the RIO configuration file.

   The file is saved in the root of the application's directory.
7. In the **Communications Setup** editor, create a device shortcut that points to the RIO data device.

   For information about creating a device shortcut, see RSLinx Help.
8. Apply the shortcut to the RIO driver.
9. Correct any invalid RIO configurations. Invalid RIO configurations are highlighted with red "x" icons.

   Tip: If red "x" icons appear after you import the RIO .xml file, you can fix block tags in the **Communication Setup** editor in FactoryTalk View. The length of block tags must be the same for Read and Write pairs sharing the same rack, group, and slot.
10. Save the converted RIO application.

Unsupported PanelBuilder RIO tags

A PanelBuilder RIO tag will be converted to an HMI memory tag and an error will be logged to the conversion log file if the RIO tag:

- Has a blank address.
- Has a data type of Bit Array and its address does not have an array size of 1, 8, 16, or 32.
- Has a data type of Bit Array, and its array size is 16 or 32, but its address does not have a bit offset of 0.
- Has a data type of Bit or BOOL, and its address does not contain the bit delimiter character "/".
- Has a data type of 4-BCD, Unsigned Integer, Signed Integer or INT, Character Array, or DINT, and its address contains the bit delimiter character "/".

- Is a block transfer tag with a data type of Bit Array, and its array size is 8, but its address does not have a bit offset of 0 or 8.
- Is a block transfer tag with a data type of SINT, and its address does not have a bit offset of 0 or 8.
- Is an I/O tag with an address that references an undefined rack.
- Is an I/O tag with a data type of SINT, and its address does not have a bit offset of 0 or 10.
- Does not have a valid I/O address or block transfer address.

Unsupported PanelBuilder graphic object features

This section describes features of PanelBuilder graphic objects that are not supported in FactoryTalk View. The Notes column provides additional information and describes methods for achieving the same result when possible.

Graphic object	Unsupported feature in FactoryTalk View	Notes
Image text, arc, ellipse, freehand, line, polyline, rectangle, wedge	Blink property	In FactoryTalk View, all of the listed objects except images use color animation to blink. For details, see Set up color animation on page 344. Color images do not blink. Monochrome images use the Blink property to blink.
Increment/Decrement Entry Button (Converted to Numeric Input Enable button)	Allow Home/End Allow Wrap Ramp by coarse steps	
Maintained push Button, Multistate Push Button, Standard Control List Selector	Initial state values	If you want to set these objects' states on application startup, create a macro to set the appropriate tag values for the objects' connections. For information about macros, see Use macros on page 421. Assign the macro in the **Startup** editor. For details, see Help.
Multistate Indicator, Message Display	Print Setting	
Bar Graph	Inner text and inner graphic	Converted to a separate text object and image object.
Gauge	Inner text and inner graphic Scale clipping Needle	Converted to a separate text object and image object. If the scale does not fit within the height or width of the gauge, it is not clipped. Check the position of the scale to ensure it does not overlap other objects. Converted to a separate gauge object; if the gauge had 2 needles, each needle is converted to a separate gauge object.
Alarm List	No Acknowledgement Required	All alarms can be acknowledged.
All objects	Image placement	FactoryTalk View supports one, three, or nine positions for images, depending on

Graphic object	Unsupported feature in FactoryTalk View	Notes
		the type of object. On conversion, images are positioned using the closest match. Therefore some images might be clipped to fit the object.
All objects	Turn Object View On property	If this property is set to False, the converted object has a transparent background, no border, no caption, and no image.
All objects	Blinking inner graphics	If the inner graphic uses a color image, it will not blink. Use a monochrome image if you want the inner graphic to blink.

System tags

This appendix describes system tags.

System tags provide the following types of information:

- Alarms
- Time
- Graphics
- User

Alarms

The following tag contains the time and date when the status of alarms was last reset. The date uses the long date format.

Tag name	Type	Description
system\AlarmResetDateAndTimeString	String	Read-only. Contains the date and time of the last alarm reset. At runtime, date and time information stored in this tag is shown in the language of the runtime Windows operating system and in the format specified by the **Regional** or **Regional and Language** options in the Windows Control Panel.

For information about resetting alarms, see Methods for resetting alarms on page 101.

Time

These system tags record time and date information in various formats.

Tag names	Types	Descriptions
system\Date	String	Read-only. System date. At runtime, date and time information stored in this tag is shown in the format specified by the **Regional** or **Regional and Language** options in the Windows Control Panel.
system\DateAndTimeInteger	Analog	Read-only. Number of seconds elapsed since midnight (00:00:00) January 1, 1970, coordinated universal time (UTC).
system\DateAndTimeString	String	Read-only.

Tag names	Types	Descriptions
		Complete date and time display. For example: Monday, December 12 2001 10:47:50 AM.
system\DayOfMonth	Analog	Read-only. Day of the month (1 - 31).
system\DayOfWeek	Analog	Read-only. Day of the week (1-7); Sunday = 1.
system\DayOfYear	Analog	Read-only. Day of the year (1-366).
system\Hour	Analog	Read/write. Hour of the day (0-23).
system\Minute	Analog	Read/write. Minutes (0 - 59).
system\Month	Analog	Read-only. Number for month (1-12).
system\MonthString	String	Read-only. Name of the month. At runtime, date and time information stored in this tag is shown in the format specified by the **Regional** or **Regional and Language** options in the Windows Control Panel.
system\Second	Analog	Read/write. Seconds (0 - 59).
system\Time	String	Read-only. System time. At runtime, date and time information stored in this tag is shown in the format specified by the **Regional** or **Regional and Language** options in the Windows Control Panel.
system\Year	Analog	Read-only. The year (1980-2099). At runtime, date and time information stored in this tag is shown in the format specified by the **Regional** or **Regional and Language** options in the Windows Control Panel.

For information about using the data source to update the system date and time, or about sending the runtime computer's date and time to the data source, see Set up global connections on page 89.

Graphics

The following HMI tags can be used to make graphic objects appear as though they are blinking on and off.

Tag names	Types	Descriptions
system\BlinkFast	Digital	Toggles on and off every 100 ms (10 times per second).
system\BlinkSlow	Digital	Toggles on and off every 500 ms (twice per second).

A more efficient way to make graphic objects blink is to use the blinking color option in color animation. For details, see Set up color animation on page 344.

Many objects have a **Blink** property that you can set up. For information about specific objects, see Help.

User

The following tag contains the name of the current user.

Tag names	Types	Descriptions
system\User	String	Contains name of the logged-in user.

We recommend that you use the expression security function CurrentUserName( ) instead of the system\User tag, especially if you intend to convert the application to FactoryTalk View Supervisory Edition. In distributed applications, system\User returns the name of the user logged into the HMI server, not the user logged into the display client.

For more information about the security functions, see Security functions on page 362.

ODBC database schema

This appendix describes the ODBC database format, or schema, for messages from FactoryTalk Diagnostics. The target table of the ODBC database to which you are sending messages must use the format shown in this appendix.

The option of logging FactoryTalk Diagnostics messages to an ODBC database is available for personal computers only.

For information about setting up FactoryTalk Diagnostics, see Set up FactoryTalk Diagnostics on page 126.

FactoryTalk Diagnostics log table

FactoryTalk Diagnostics log data in ODBC format uses one table.

This column	Contains	SQL data type	Length
TimeStmp	The time and date data was logged in coordinated universal time format. Encoded as a date variant.	SQL_TIMESTAMP	Driver dependent
MessageText	Message to be logged.	SQL_VARCHAR, *or* SQL_CHAR	254
Audience	A number representing the message audience: • 0 for Operator • 1 for Engineer • 2 for Developer • 3 for Secure	SQL_SMALLINT, *or* SQL_INTEGER	1
Severity	A number representing the severity of the diagnostics message: • 0 for Error • 1 for Warning • 2 for Information • 3 for Audit	SQL_SMALLINT, *or* SQL_INTEGER	1
Area	The FactoryTalk path to the area in which the activity occurred. Used for FactoryTalk View SE only.	SQL_VARCHAR, *or* SQL_CHAR	80
Location	The name of the computer where the message was generated.	SQL_VARCHAR, *or* SQL_CHAR	15
UserID	The name of the user (including domain name, if there is one) that initiated the action that caused the	SQL_VARCHAR, *or* SQL_CHAR	38

This column	Contains	SQL data type	Length
	diagnostics message. If the diagnostics message was caused by an HMI server, the user column contains "System."		
UserFullName	The full name of the user that was logged in when the activity occurred.	SQL_VARCHAR, *or* SQL_CHAR	255
Provider	The name of the product that generated the message.	SQL_VARCHAR, *or* SQL_CHAR	20

RFC1766 names

This appendix describes RFC1766 names for Windows languages.

Mapping languages to RFC1766 names

The following table lists the languages that Windows supports and the RFC1766 name associated with each language. You can use the codes to name the translated application files before importing them. The codes are also used with the CurrentLanguage function.

RFC1766 Name	Language – Country/Region
af-ZA	Afrikaans – South Africa
sq-AL	Albanian – Albania
ar-DZ	Arabic – Algeria
ar-BH	Arabic – Bahrain
ar-EG	Arabic – Egypt
ar-IQ	Arabic – Iraq
ar-JO	Arabic – Jordan
ar-KW	Arabic – Kuwait
ar-LB	Arabic – Lebanon
ar-LY	Arabic – Lybia
ar-MA	Arabic – Morocco
ar-OM	Arabic – Oman
ar-QA	Arabic – Qatar
ar-SA	Arabic – Saudi Arabia
ar-SY	Arabic – Syria
ar-TN	Arabic – Tunisia
ar-AE	Arabic – United Arab Emirates
ar-YE	Arabic – Yemen
hy-AM	Armenian – Armenia
az-AZ-Cyrl	Azeri (Cyrillic) – Azerbaijan
az-AZ-Latn	Azeri (Latin) – Azerbaijan
eu-ES	Basque – Basque
be-BY	Belarusian – Belarus
bg-BG	Bulgarian – Bulgaria
ca-ES	Catalan – Catalan
zh-HK	Chinese – Hong Kong SAR (Default Sort Order – Stroke Count)
zh-HK	Chinese – Hong Kong SAR (Alternate Sort Order – Stroke Count)
zh-MO	Chinese – Macau SAR (Default Sort Order – Pronunciation)
zh-MO	Chinese – Macau SAR (Alternate Sort Order – Stroke Count)

RFC1766 Name	Language – Country/Region
zh-CN	Chinese – China (Default Sort Order – Pronunciation)
zh-CN	Chinese – China (Alternate Sort Order – Stroke Count)
zh-SG	Chinese – Singapore (Default Sort Order – Pronunciation)
zh-SG	Chinese – Singapore (Alternate Sort Order – Stroke Count)
zh-TW	Chinese – Taiwan (Default Sort Order – Stroke Count)
zh-TW	Chinese – Taiwan (Alternate Sort Order – Bopomofo)
hr-HR	Croatian – Croatia
cs-CZ	Czech – Czech Republic
da-DK	Danish – Denmark
div-MV	Dhivehi – Maldives
nl-BE	Dutch – Belgium
nl-NL	Dutch – The Netherlands
en-AU	English – Australia
en-BZ	English – Belize
en-CA	English – Canada
en-CB	English – Caribbean
en-IE	English – Ireland
en-JM	English – Jamaica
en-NZ	English – New Zealand
en-PH	English – Philippines
en-ZA	English – South Africa
en-TT	English – Trinidad and Tobago
en-GB	English – United Kingdom
en-US	English – United States
en-ZW	English – Zimbabwe
et-EE	Estonian – Estonia
fo-FO	Faroese – Faroe Islands
fa-IR	Farsi – Iran
fi-FI	Finnish – Finland
fr-BE	French – Belgium
fr-CA	French – Canada
fr-FR	French – France
fr-LU	French – Luxembourg
fr-MC	French – Monaco
fr-CH	French – Switzerland
mk-MK	FYRO Macedonian
gl-ES	Galician – Galician

RFC1766 Name	Language – Country/Region
ka-GE	Georgian – Georgia (Default Sort Order – Traditional)
ka-GE	Georgian – Georgia (Alternate Sort Order – Modern Sort)
de-AT	German – Austria
de-DE	German – Germany (Default Sort Order – Dictionary)
de-DE	German – Germany (Alternate Sort Order – Phone Book Sort DIN)
de-LI	German – Liechtenstein
de-LU	German – Luxembourg
de-CH	German – Switzerland
el-GR	Greek – Greece
gu-IN	Gujarati – India
he-IL	Hebrew – Israel
hi-IN	Hindi – India
hu-HU	Hungarian – Hungary (Default Sort Order)
hu-HU	Hungarian – Hungary (Alternate Sort Order – Technical Sort)
is-IS	Icelandic – Iceland
id-ID	Indonesian – Indonesia
it-IT	Italian – Italy
it-CH	Italian – Switzerland
ja-JP	Japanese – Japan (Default Sort Order)
ja-JP	Japanese – Japan (Alternate Sort Order – Unicode)
kn-IN	Kannada – India
kk-KZ	Kazakh – Kazakhstan
kok-IN	Konkani – India
ko-KR	Korean – Korea (Default Sort Order)
ko-KR	Korean – Korea (Alternate Sort Order – Korean Xwansung Unicode)
ky-KZ	Kyrgyz – Kazakhstan
lv-LV	Latvian – Latvia
lt-LT	Lithuanian – Lithuania
ms-BN	Malay – Brunei
ms-MY	Malay – Malaysia
mr-IN	Marathi – India
mn-MN	Mongolian – Mongolia
nb-NO	Norwegian (Bokml) – Norway
nn-NO	Norwegian (Nynorsk) – Norway
pl-PL	Polish – Poland
pt-BR	Portuguese – Brazil

RFC1766 Name	Language – Country/Region
pt-PT	Portuguese – Portugal
pa-IN	Punjabi – India
ro-RO	Romanian – Romania
ru-RU	Russian – Russia
sa-IN	Sanskrit – India
sr-SP-Cyrl	Serbian (Cyrillic) – Serbia
sr-SP-Latn	Serbian (Latin) – Serbia
sk-SK	Slovak – Slovakia
sl-SI	Slovenian – Slovenia
es-AR	Spanish – Argentina
es-BO	Spanish – Bolivia
es-CL	Spanish – Chile
es-CO	Spanish – Colombia
es-CR	Spanish – Costa Rica
es-DO	Spanish – Dominican Republic
es-EC	Spanish – Ecuador
es-SV	Spanish – El Salvador
es-GT	Spanish – Guatemala
es-HN	Spanish – Honduras
es-MX	Spanish – Mexico
es-NI	Spanish – Nicaragua
es-PA	Spanish – Panama
es-PY	Spanish – Paraguay
es-PE	Spanish – Peru
es-PR	Spanish – Puerto Rico
es-ES	Spanish – Spain (Default Sort Order – International)
es-ES	Spanish – Spain (Alternate Sort Order – Traditional)
es-UY	Spanish – Uruguay
es-VE	Spanish – Venezuela
sw-KE	Swahili – Kenya
sv-FI	Swedish – Finland
sv-SE	Swedish – Sweden
syr-SY	Syriac – Syria
ta-IN	Tamil – India
tt-RU	Tatar – Russia
te-IN	Telugu – India
th-TH	Thai – Thailand

RFC1766 Name	Language – Country/Region
tr–TR	Turkish – Turkey
uk–UA	Ukrainian – Ukraine
ur–PK	Urdu – Pakistan
uz–UZ–Cyrl	Uzbek (Cyrillic) – Uzbekistan
uz–UZ–Latn	Uzbek (Latin) – Uzbekistan
vi–VN	Vietnamese – Vietnam

Rockwell Automation Support

Use these resources to access support information.

Technical Support Center	Find help with how-to videos, FAQs, chat, user forums, and product notification updates.	rok.auto/support
Knowledgebase	Access Knowledgebase articles.	rok.auto/knowledgebase
Local Technical Support Phone Numbers	Locate the telephone number for your country.	rok.auto/phonesupport
Literature Library	Find installation instructions, manuals, brochures, and technical data publications.	rok.auto/literature
Product Compatibility and Download Center (PCDC)	Get help determining how products interact, check features and capabilities, and find associated firmware.	rok.auto/pcdc

Documentation feedback

Your comments help us serve your documentation needs better. If you have any suggestions on how to improve our content, complete the form at rok.auto/docfeedback.

Waste Electrical and Electronic Equipment (WEEE)

At the end of life, this equipment should be collected separately from any unsorted municipal waste.

Rockwell Automation maintains current product environmental information on its website at rok.auto/pec.

Rockwell Otomasyon Ticaret A.Ş. Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400 EEE Yönetmeliğine Uygundur

Connect with us.

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