

Studio 5000 View Designer Getting Results Guide

Version 9.01

Rockwell Automation Publication 9324-GR001E-EN-P - June 2023
Supersedes Publication 9324-GR001D-EN-P - November 2021



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.

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).

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.

Summary of changes

This manual includes new and updated information. Use these reference tables to locate changed information.

Global changes

None for this release.

New or enhanced features

This table contains a list of topics changed in this version, the reason for the change, and a link to the topic that contains the changed information.

Topic Name	Reason
Predefined Screens, the system banner, user defined screens and popups on page 21	New topic name
Create screens and popups on page 24	Popup cache information added
Automatic Diagnostics on page 113	New topic
Automatic Diagnostics and Automatic Diagnostics History on page 114	New topic
Automatic Diagnostics at Runtime on page 116	New topic
Automatic Diagnostics History at Runtime on page 117	New topic
Web Browser on page 147	New topic
Web Browser at Runtime on page 149	New topic
Web Browser Guidelines on page 147	New topic
Clearing a data log on page 159	New topic
Log data on page 156	Updated due to the functionality change
Use images on page 119	Bullet list updated

Preface	Studio5000 environment_View Designer.....	11
	Legal Notices.....	11
	Additional resources	13
	 Chapter 1	
Project Administration	Add a controller reference.....	15
	Configure a View Designer path to PanelView 5000 HMI device.....	17
	Download a project.....	17
	Upload a project.....	18
	Change the size of the onscreen keyboard and keypad.....	19
	 Chapter 2	
Predefined Screens, the system banner, user defined screens and popups	System Banner contents.....	22
	Use the settings screen at runtime.....	23
	Configure the update rate of screens and popups.....	25
	 Chapter 3	
Property binding	Bind a property.....	27
	Supported color keywords and RGB values.....	29
	Color Picker overview.....	30
	Example 1: Bind a property.....	32
	Example 2: Bind graphic element properties to other properties.....	34
	Expression overview.....	35
	Expression examples.....	35
	 Chapter 4	
Color and state tables	Example 1: Use a state table to make a motor a multi-state indicator.....	45

Example 2: Use a color table to indicate level in a mixer..... 47
 Example 3: Use a state table to display fault code text..... 48

Chapter 5

Events

Example 1: Configure button behaviors..... 52
 Example 2: Configure multiple events on a graphic element..... 54
 Example 3: Use State Enter and State Exit Events..... 59
 Example 4: Use key press and key release events..... 62

Chapter 6

Project Events

Create a project event..... 65
 Example: Write the current user name or screen name to a controller
 using a project event..... 68

Chapter 7

Navigation menu

Define shortcuts to setup Navigation Menu..... 72
 Example: Configure Navigation Menu properties for runtime..... 73

Chapter 8

Reuse screens

Example 1: Use alias properties..... 77
 Example 2: Use data type properties to create a reusable screen..... 80

Chapter 9

Active Directory Authentication

Configure network-based security logon using Active Directory..... 89
 Find the Active Directory Logon Information..... 91
 Log on using network-based security..... 92
 Adding a Certificate for Encrypted..... 93

	Chapter 10	
Add-On Graphics		
	Create an Add-On Graphic.....	95
	Example: Use custom properties for an Add-On Graphic.....	97
	Create an instance for the Add-On Graphic definition.....	99
	Chapter 11	
Alarms		
	Add alarms to the system.....	103
	View Designer alarm guidelines.....	104
	Alarm Summary and Alarm Manager.....	106
	Alarm History.....	109
	Export Alarm history.....	111
	Chapter 12	
Automatic Diagnostics		
	Enable Automatic Diagnostics.....	113
	Automatic Diagnostics and Automatic Diagnostics History.....	114
	Automatic Diagnostics at Runtime.....	116
	Automatic Diagnostics History at Runtime.....	117
	Chapter 13	
Use images		
	Example 1: Configure the image to work as a momentary button.....	120
	Example 2: Change images displayed in an image element at runtime...	121
	Chapter 14	
Language switching		
	Configure language switching.....	125
	Example: Configure language commands to switch languages.....	127
	Chapter 15	
PDF Viewer		

Example 1: Add a PDF Viewer Add-On Graphic to a screen.....	133
Example 2: Drag a PDF document onto the screen.....	135
Example 3: Create custom PDF Viewer.....	138
Bind bookmark properties of a PDF graphic element.....	143
Guidelines for working with PDF documents.....	145

Chapter 16

Web Browser

Web Browser Guidelines.....	147
Open Web Browser Command.....	149
Browser Certificates.....	150

Chapter 17

Trend chart and data log overview

Display data on a trend chart.....	153
Log data.....	156
Clearing a data log.....	159

Chapter 18

Load from Media

Load a project from media.....	161
--------------------------------	-----

Chapter 19

VNC

Configure VNC access in View Designer.....	163
VNC popup.....	163

Chapter 20

Email notification

Configure the connection to the email server.....	165
Send email command.....	166
Emails as texts.....	167

Example: Configure a list of email recipients
online and send them emails.....167

Studio5000 environment_View Designer

The Studio 5000 Automation Engineering & Design Environment® combines engineering and design elements into a common environment. The Studio 5000® environment is the foundation for the future of Rockwell Automation® engineering design tools and capabilities. The Studio 5000 environment is the one place for design engineers to develop all elements of their control system.



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.

End User License Agreement (EULA)

You can view the Rockwell Automation End User License Agreement (EULA) by opening the license.rtf file located in your product's install folder on your hard drive.

The default location of this file is:

C:\Program Files (x86)\Common Files\Rockwell\license.rtf.

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 licenses.txt files located 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 locations of this file is:

C:\Program Files (x86)\Common Files\Rockwell\Help\Studio 5000 View Designer Help\V9\Release Notes\OPENSOURCE\licenses.txt

You may obtain Corresponding Source code for open source packages included in this product from its 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
Actipro WPF Controls	Copyright © Actipro Software LLC
KODAK DIGITAL SCIENCE Java CMM	Copyright © Eastman Kodak Company
Microsoft .NET	Copyright © Microsoft Corporation
Microsoft Visual Studio 2015 Isolated Shell	Copyright © Microsoft Corporation
Microsoft Visual Studio Internal Debugging Proxy/Stub	Copyright © Microsoft Corporation
Microsoft Expression Interactions	Copyright © Microsoft Corporation
Microsoft Runtime Library	Copyright © Microsoft Corporation
Foxit PDF Viewer	Copyright © Foxit Software Incorporated
QT	Copyright © The Qt Company
Telerik Windows Controls	Copyright © Progress Software Corporation
Telerik Windows Data	Copyright © Progress Software Corporation
Xceed Controls for WPF	Copyright © Xceed Software Inc.
Xceed DataGrid for WPF	Copyright © Xceed Software Inc.

THIS PRODUCT IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO (i) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD (“AVC VIDEO”) AND/OR (ii) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO

PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION MAY BE OBTAINED FROM MPEG LA, L.L.C. SEE [HTTP://WWW.MPEGLA.COM](http://www.mpegla.com)

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.

Additional resources

These documents contain additional information concerning related Rockwell Automation products.

Resource	Description
Industrial Automation Wiring and Grounding Guidelines , publication 1770-4.1	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Certifications webpage, available at http://ab.rockwellautomation.com	Provides declarations of conformity, certificates, and other certification details.
PanelView 5310 Terminals User Manual publication 2713P-UM001	Describes how to install, configure, operate, and troubleshoot the PanelView 5310 terminals.
PanelView 5510 Terminals User Manual publication 2715P-UM001	Describes how to install, configure, operate, and troubleshoot the PanelView 5510 terminals.

View or download publications at <http://www.rockwellautomation.com/literature>. To order paper copies of technical documentation, contact the local Rockwell Automation distributor or sales representative.

Project Administration

Creating a project is the first step in building a runtime application. A View Designer project includes screens, controller references, and information about the target HMI device. View Designer creates and uses the file extension .vpd for user-created projects. By default, projects are saved to <user>\My Documents\Studio 5000\Projects. In View Designer a project is stored as a single .vpd file.

Use projects stored in a single .vpd files to:

- Copy a project to another computer. Copy the .vpd file and open with View Designer.
- Save a project as a new name. Select **File > Save Project As**.
- Open a View Designer project by selecting **File > Open Project** or double-clicking the .vpd file.

See also

[Add a controller reference](#) on [page 15](#)

[Configure a View Designer path to PanelView 5000 HMI device](#) on [page 17](#)

[Download a project](#) on [page 17](#)

[Upload a project](#) on [page 18](#)

Add a controller reference

A controller reference is a connection to a Logix Designer project file (.acd) from a View Designer project file (.vpd). The connection automatically synchronizes data between a Logix Designer project file (.acd) and View Designer project file (.vpd). After referencing the Logix Designer project, browse the Logix Designer tags in a View Designer project. When saving an (.acd) file with new data, such as new tags, Studio 5000 View Designer automatically synchronizes the new data with the View Designer project file every 15 seconds.

The controller reference card shows a progress symbol when View Designer synchronizes.

For a PanelView 5510, View Designer supports up to four controller references. For a PanelView 5310, View Designer supports a single controller reference.

Adding a controller reference adds the controller reference to the next available reference card on the **References** tab. Each controller reference has a number in the reference card. This controller number appears when browsing for HMI device controller tags in the **Tag Browser**.

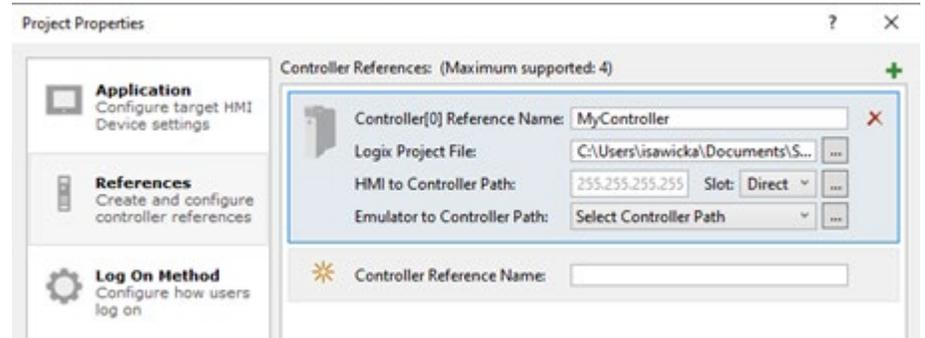
IMPORTANT Select a controller only from an Ethernet network. Logix controllers and project files must be version 27 or later.



Tip: For projects with a PanelView 5000 HMI device, the recommended number of tags in a single Logix controller is 200,000. If referencing four controllers, the recommended number of tags is 50,000. A tag is a scalar tag. Scalar tags are BOOL, DINT, REAL, a member of structure, or an element of an array.

To add a controller reference

1. On the **Project Properties** dialog box, click the **References** tab.



2. In the **Controller[#] Reference Name** box, type a unique name for the controller reference.



Tips:

- Add up to four controller references for projects configured with an HMI device.
 - Entering a controller reference name or clicking **Add controller reference**  adds a controller reference card for projects configured with an HMI device.
 - Added controller references appear after the last controller reference card. After saving controller references, reference cards appear in order by controller reference number that appears in the **Controller[#] Reference Name** box.
 - Adding the maximum number of controller references for an HMI device makes **Add controller reference**  unavailable.
 - Controller References each contain an index number (0, 1, 2, or 3). When controllers replicate at the same time, the controller with the lowest index number (0) replicates before the other controllers in the queue. The controller with the lowest index number in queue has the highest priority.
3. Type the path of the file or select a Logix Designer project file (.acd) by clicking **Browse**  next to the **Logix Project File** box.
 4. Configure an Ethernet driver in FactoryTalk Linx for browsing to the controller path on the same network as the controller.
 5. Set the path from the HMI device to the controller:
 - a. Type the IP address of the controller. To browse for the controller, after the **HMI to Controller Path** box, click **Browse**  and select the controller running the selected Logix Designer project file (.acd).

- b. In the **Slot** box, select the slot number of the controller to reference. The slot number appears automatically after browsing for an HMI to Controller Path. Specifying the slot is not applicable for CompactLogix controllers.
 6. Set the path from the **View 5000 Emulator** to the controller:
 - a. After the **Emulator to Controller Path** box, click **Browse**  and select the controller running the selected Logix Designer project file (.acd).
 - b. (optional) If the controller was previously selected, click the **Select Controller Path** list to select a recently used path.
-  **Tips:**
- Create the path to the same or different controller for the physical HMI device to use to enable testing projects on a different test controller and avoid affecting the controller running the process or machine.
 - The **Select Controller Path** list has up to the last five recently selected controller paths. If there are no previously viewed paths, the list does not appear.
7. Click **Apply** to save changes and keep the **Project Properties** dialog box open or click **OK** to save changes and close the dialog box.
-  **Tip:** When browsing for a controller, the **OK** button is available when selecting a View Designer-compatible device.

See also

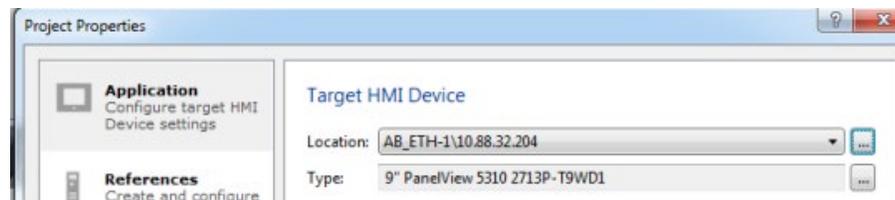
[Configure a View Designer path to PanelView 5000 HMI device on page 17](#)

[Download a project on page 17](#)

[Upload a project on page 18](#)

Configure a View Designer path to PanelView 5000 HMI device

After defining controller paths, define the path in the **Location** field on the **Application** tab for downloading and uploading projects to the PanelView 5000 HMI device.



See also

[Download a project on page 17](#)

[Upload a project on page 18](#)

Download a project

After defining the path to the PanelView 5000 HMI device, download projects to the PanelView 5000 HMI device.

To download a project

1. On the **Menu** bar, click **Communications > Download** or press the **Download** icon  on the **Menu** bar to open the **Download Runtime Application** wizard.
2. Next to the **Location** box on the **HMI Device Location** page either:
 - click **Browse**  and select the IP address of the HMI device to download the runtime application
 - Select the IP address in the **Location** box. The **Location** box lists up to five previously used download locations.
3. (optional) If the project has multiple languages, in the **Language** box, select the default language to display in the project.



Tip: If a translatable string is blank, View Designer defaults to the download language for runtime applications downloaded to the HMI device. If the translated string for the download language is also blank, View Designer defaults to the language used for designing the project.

4. Click **Next**.
5. On the **Controller References** page verify that the information is correct.



Tip: To make changes, exit the wizard. Make and save the necessary changes on the **References** tab (**Project > Project Properties > References**).

6. Click **Download**. The HMI device displays a progress screen.



Tip: Once download is complete, the HMI device automatically starts running the project.

See also

[Upload a project](#) on [page 18](#)

Upload a project

Upload a project from an HMI device, which is useful when you do not have the .vpd file for the project running in the HMI device.

IMPORTANT Runtime applications have .vpd extensions. An uploaded file with a changed extension will not download to the HMI device

To upload and save a runtime application

1. On the Menu bar, click **COMMUNICATIONS > Upload**. The Upload Runtime Application wizard opens.
2. In the HMI Device Path box on the HMI Device Location page, select the IP address of the HMI device that has the runtime application:
 - Click the arrow to select a previously used path.
 - Click **Browse**  and select the HMI device.
3. Click **Upload**. The HMI Device Location page displays the progress of the upload process.



Tip: If a problem occurs, an error message displays and the process stops. Resolve all errors before uploading the runtime application.

4. In the Project File box on the Save page, select the location to save the file:
 - Click Browse  to navigate to the location.
 - Type the path in the Project File box.
5. Click Save. Finished appears selected when the process is complete.
6. Click Close.



Tip: During the first use of the Upload Runtime Application wizard, the HMI Device Path box appears blank. Reopening the wizard defaults to the most recent and successfully used IP address.

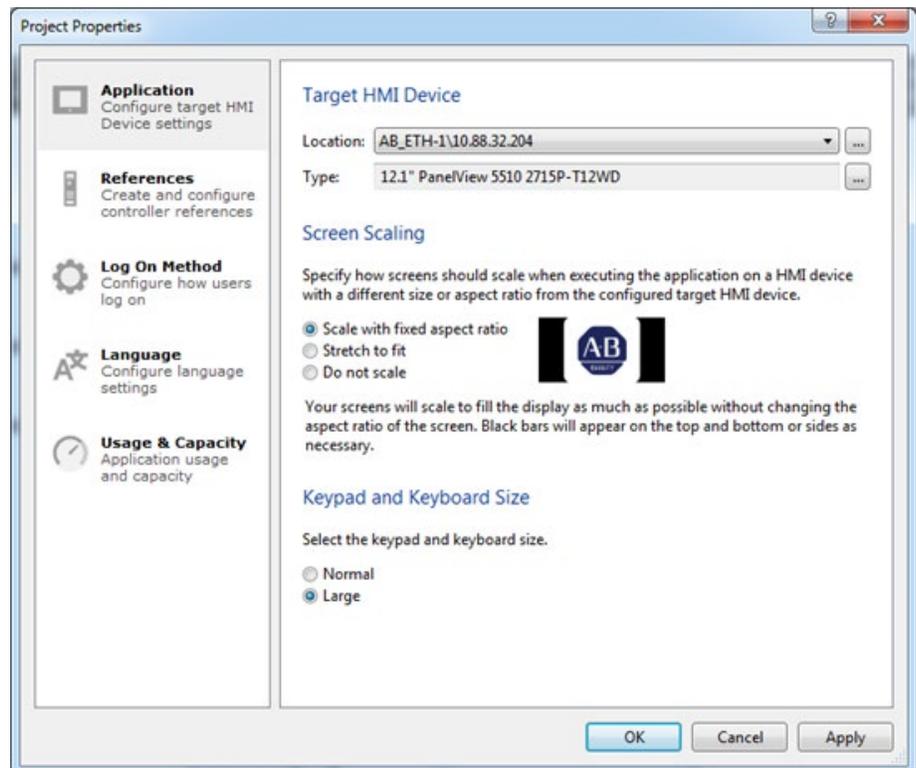
See also

[Project Administration](#) on [page 15](#)

Change the size of the onscreen keyboard and keypad

The onscreen keyboard and keypad is the primary way to input values to the HMI device. You can change the size of the keypad and keyboard to adapt it to the size of the screen.

To change the size of the keypad and keyboard, open Project Properties, select the Application tab and select a size.



The **Normal** size covers a smaller area of the screen while entering values. The Large size is better for entering data while wearing gloves. If the large keypad or keyboard does not fully fit on the PanelView 5000 HMI device, it automatically scales down to fit the screen.

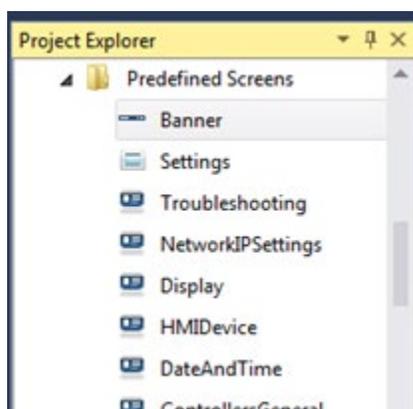
Predefined Screens, the system banner, user defined screens and popups

View Designer includes a set of predefined screens and a banner in the **Predefined Screens** folder in **Project Explorer**. Use predefined screens and the system banner to configure and view the status of an application. Optionally modify the content of the predefined banner and screens.



Tips:

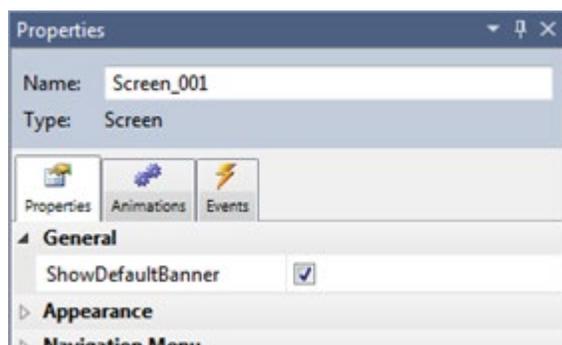
- View Designer does not allow renaming, deleting or adding screens or popups to the **Predefined Screens** folder.
- Content in the **Predefined Screens** folder does not count toward the screen limit of the HMI device.



The system banner displays at the top of every screen by default. To remove the system banner from any screen, click the **Properties** tab of the screen and clear the **ShowDefaultBanner** property check box.



Tip: Removing the system banner from a screen does not remove the system banner or its items from the project.



See also

[System Banner contents](#) on [page 22](#)

[Use the settings screen at runtime](#) on [page 23](#)

System Banner contents

The **System Banner** has a default set of items:



Item	Description
1	 <p>Alarm Status Indicator opens the Alarm Summary screen on the HMI device.</p> <ul style="list-style-type: none">  The button shows that there are no alarms.  The button on the screen turns red and flashes when there are active alarms that are unacknowledged. The numbered notification shows the number of unacknowledged alarms.  The button returns to a gray icon with a numbered notification showing the number of unacknowledged alarms when all alarms return to normal but some alarms are still unacknowledged.  The button turns to red when there are active alarms and all of them are acknowledged.
2	 <p>Automatic Diagnostics Indicator opens Automatic Diagnostics screen on the HMI device. The button shows a notification with the number of active diagnostics. The button returns to an outlined icon when no diagnostics are active.</p>
3	 <p>Previous displays the previous screen in the navigation history on the HMI device.</p>
4	 <p>Next displays the next screen in the navigation history on the HMI device. To navigate with Next requires displaying a previous screen first.</p>
5	 <p>Navigation displays the Navigation menu at the bottom of the screen on the HMI device. Use the Navigation menu to display shortcuts and folder contents.</p>
6	 <p>Log On logs on or off the project. The name of the logged on user appears in the System Banner. The logged on user sees only the screens for which the assigned user role has access.</p>
7	<p>General Diagnostics Status for the data log status and project event errors. These icons appear depending on the status of data export and project events.</p> <ul style="list-style-type: none">  Data export ongoing. The data log is exporting.  Data export error. The data log is in error while collecting data or exporting.  Data export remove. The media containing the data log is ready to remove without corrupting files or making files read-only on the USB storage device or SD card.  Data export warning. Tags in the data log are in error. Select the icon to view the error details. <p>A data log status icon only appears when a project contains a data log and there are no project event errors.</p> <ul style="list-style-type: none">  Error. Project events and data logs are in error. Select this icon to open the predefined popup for data log and project event status details.  Project Event. Project events are in error. Select the icon to view the error details. The Project Event Error icon only appears when a project event is in error.
8	<p>A Network Status icon appears only if there is an issue with network communication. Select the icon to open the HMI Device Configuration Network popup for details about the issue.</p> <ul style="list-style-type: none">  Network caution. The network is functioning but has an issue that requires attention. For example, the Device Level Ring may have a break.  Network error. The network is not functioning. An Ethernet link is disconnected, or there is a duplicate IP address.

9	<p>A Controller Status icon appears only if there is an issue with any controller in the project. Select the icon to open the Controllers General popup for details about the issue.</p> <ul style="list-style-type: none"> • Controller error. Any controller in the project is disconnected, powered down, or not configured. • Controller caution. Any controller in the project is not in Run mode or the tag data of a controller is not synchronized with the HMI device. • Controller unknown. No controller in the project is visible and it is unclear if a controller should be visible to the HMI device. This may occur when a controller is on a disconnected side of the network.
10	The current time and date of the HMI device.

To open and change the System Banner, in the **Predefined Screens** folder, double-click **System Banner**. Changing the System Banner includes moving or deleting existing elements or adding user-defined content. For example, add a company logo to the System Banner to have the logo appear at the top of every screen.

See also

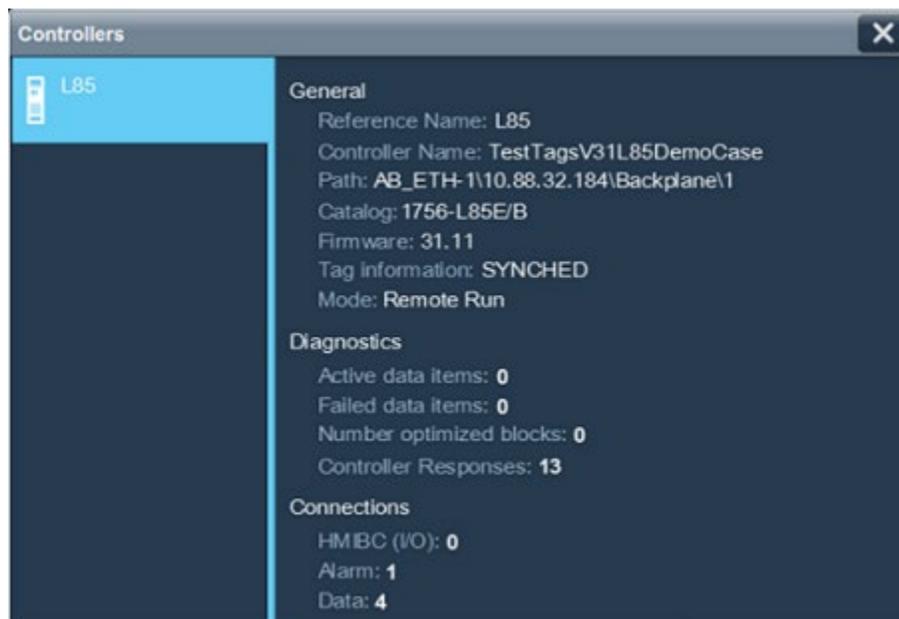
[Predefined Screens, the system banner, user defined screens and popups](#) on [page 21](#)

Use the settings screen at runtime

Launch predefined screens from the **Settings** screen to configure or interact with a specific part of the View Designer application.



For example, select **Network** to configure network settings and view network status. Select **Controllers** to display the status of the controllers configured for the HMI device.



All predefined screens and popups support security settings for assigning user roles that can view and change screens and interact with screens at runtime.

See also

[Predefined Screens, the system banner, user defined screens and popups](#) on [page 21](#)

[System Banner contents](#) on [page 22](#)

Create screens and popups

Use screens and popups to create the content visible on the HMI device. A View Designer project for a PanelView 5310 can contain no more than a total of 100 screens and popups. A View Designer project for a PanelView 5510 can contain no more than a total of 500 screens and popups. Projects must have a minimum of one screen.

Characteristics of screens:

- Screens fill the entire display space of a HMI device.
- Screens can be referenced by shortcuts in the Navigation menu.
- Screens display the System Banner at the top by default.
- Screens have a fixed opacity of 100%.

Characteristics of popups:

- Users can define the size of each popup in View Designer.

- For popups displayed on HMI device screens smaller than the popup, the popup is automatically scaled to fit the smaller HMI device screen.
- Popups always overlay a screen.
- Only one popup can be displayed at a time.
- The default position of a popup is the center of the screen. The operator can move the popup by selecting and dragging the caption bar.
- Selecting an element outside the popup closes the popup.
- The Caption property of a popup allows displaying of static text or binding to display dynamic text in the caption bar.
- The CaptionVisible property allows to set the caption bar on or off.
- The CloseButtonVisible property allows to set the close button on or off.
- The BackgroundOpacity property sets the opacity of the popup allowing the user to see the content of the background screen.
- You can cache a popup for better performance by using the Cacheable property.
 - A popup with Cachable set will be cached if it fits into the 20MB of memory reserved for popup caching.
 - Most popups display quickly without caching.
 - Popups with complex graphics may display quicker when cached.
 - Popups with Cachable set which do not fit in the cache are identified by a warning during verification in View Designer.

To create a new screen or popup in View Designer:

1. Click the right mouse button on the **User-Defined Screens** folder in the **Project Explorer**, and select **New Screen** or **New Popup**.
The screen or popup appears in the User-Defined Screens folder and can be renamed.
2. Double-click the screen or popup name to open it in the View Designer canvas.
3. Double-click or drag content from the **Toolbox** to place graphic elements on the screen or popup.

To display a screen at runtime, create a shortcut to the screen in Navigation Menu folder or use a **Screen Navigate** command. To display a popup at runtime, use a **Popup Open** command.

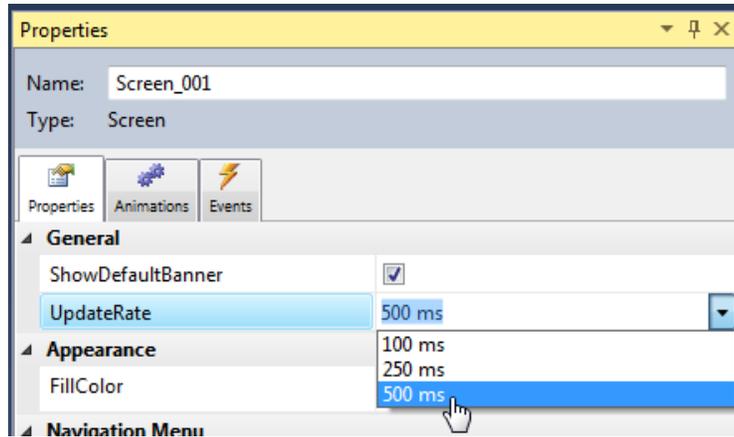
Configure the update rate of screens and popups

You can configure screens and popups with different update rates. By default, screens and popups update with new values every 500 milliseconds. To change the update rate, select a value of the UpdateRate property from the dropdown list.

The update rates are:

- 100 ms

- 250 ms
- 500 ms



For best results, limit the number of tag references on a screen.

The approximate number of tag references supported on a screen at different update rates are:

HMI device	Tag references at 100 ms	Tag references at 250 ms	Tag references at 500 ms
PanelView 5310	50	100	1000
PanelView 5510	100	150	2000

The actual number of tag references at the different update rates may change depending on other background tasks in your applications, such as data logging, alarm handling, and project events.

If the screen or popup cannot process the tag references as quickly as the configured update rate, the HMI device automatically uses a slower update rate. Closing and reopening the screen or popup attempts to return the update rate back to the faster setting.

Use the system tag `::Local:HMIDevice.Display.CurrentScreenUpdateRate` to see the current rate used by a screen.

Property binding

Binding specifies the value of a property dynamically. Graphic element properties that are bound to data update automatically when the value of the data changes. Most properties of graphic elements are available for binding.

Properties of a graphic element can be bound to the following items:

- A tag, including system tags. Browse for tags to bind to after you configure a controller reference on the **Project Properties** dialog box.
- Extended properties of tags, such as `.@Description`, `.@Max`, and so forth.
- An expression. For example, `::controller\program.tag1+::controller\program.tag2`.
- A property of a graphic element or screen. For example, you can bind the `Text` property of `TextDisplay_003` to `TextDisplay_001.Text`.



Tips:

- Every Logix analog tag can have `.@EngineeringUnit`, `.@Min`, and `.@Max` extended properties.
- Every Logix BOOLEAN tag reference can have `@EngineeringUnit`, `.@State0`, and `.@State1` extended properties.
- Every Logix tag also has `.@Description` and `.@Name` extended properties.

Use binding to attach a tag or expression to a property to change its value at runtime. Animate graphic elements by binding to their properties. Many graphic elements in the **Toolbox** have built-in animations to bind to properties. For example, bind a tag or an expression to the **Level** property of a Tank element to animate the level that appears on the screen.

See also

[Bind a property](#) on [page 27](#)

[Example 1: Bind a property](#) on [page 32](#)

[Example 2: Bind graphic element properties to other properties](#) on [page 34](#)

Bind a property

Bind a property to attach a tag or expression to a property to change the property value at runtime.

To bind a property

1. Expand the categories in the **Properties** tab to find the property to bind to a tag or expression.
2. Hover over the property to display and click the **Binding** button  and select **Bind property to item**.
3. Click **Select Tag**  to open the **Tag Browser**.
4. Create the binding by performing one of the following:
 - Navigate to and select a tag or extended property.
 - Type an expression in the value field for the property. Click **Open Expression Editor**  to edit complex expressions

-
- IMPORTANT**
- When binding to a property that consists of a list such as FontName, Format, Rounding, use a string tag that exactly matches the text of the item in the list. For example, to bind and use the Saturday Sans font, the string must have Saturday Sans ICG to be an exact match. If the value of the tag does not match a value in the list, the binding uses the first item.
 - References to graphic element properties in a binding do not update automatically if a graphic element is renamed.
 - When binding tags to numeric properties:
 - Binding a numeric property to a REAL tag rounds the resulting value.
 - Binding a numeric property to a STRING tag or property source truncates the resulting value.
-

Color property bindings support other binding methods.

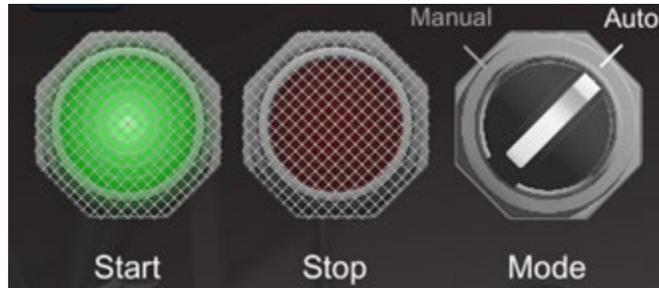
- Use one of the following methods to bind a color property to a tag or an expression:
 - Bind to a string tag or an expression that resolves to a string in the format #RRGGBB. R, G, B are the Hex digits that represent the values for the red, green, and blue color channels. For example, the string "#ffff00" produces yellow.
 - Bind to a string tag or to an expression that resolves to a string of a supported color name. For example, the string "yellow" produces yellow.

-
- IMPORTANT**
- Bind color properties only to string values. Binding to an integer tag, a floating point tag, or an expression which resolves to a non-string does not produce a change in color.
 - View Designer requires selecting a default color. Selecting NoFill does not apply the bound tag value at runtime.
-

These properties have additional considerations when binding:

- **Enabled.** Controls whether a touch or key event runs if configured for a graphic element. For example, bind to this property to control when an operator can press the element to execute any commands of a touch event for the element. For example, do this to disable start and stop buttons when the device is not in Manual mode.
- **UsePredefinedDisabled.** Enables turning off the built-in disabled animation. The button and numeric input graphic have built-in

disabled animations. For a complete list of graphic elements that have built-in disabled animations, see the [Studio 500 View Designer User Manual](#), publication 9324-UM001A. A white cross hatch appears on the element if the Enabled property is False or if the current user role has read-only access to the screen. This informs the operator that the graphic element is not operational.



- **Access.** Override the security of a screen for an individual graphic element. The default value of *Inherit* assigns the graphic element the security access of the screen. For example, if a user logs on with read-only access to the screen, touch events on graphic elements do not work. If a button must run, even on a read-only screen, change the Access property to "Full Access" to override security for that specific graphic element. Full Access is often useful for navigation buttons on a screen.

See also

[Example 1: Bind a property](#) on [page 32](#)

[Example 2: Bind graphic element properties to other properties](#) on [page 34](#)

[Property binding](#) on [page 27](#)

[Supported color keywords and RGB values](#) on [page 29](#)

Supported color keywords and RGB values

Keywords to use as a string to select a color when binding a color property include:

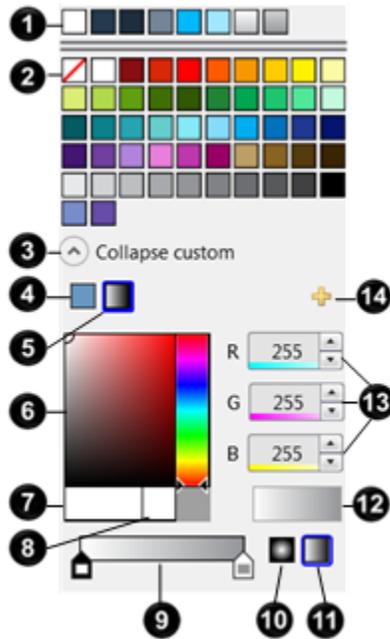
Color Name	#RRGGBB	Color Name	#RRGGBB	Color Name	#RRGGBB
aliceblue	F0F8FF	gainsboro	DCDCDC	mistyrose	FFE4E1
antiquewhite	FAEBD7	ghostwhite	F8F8FF	moccasin	FFE4B5
aqua	00FFFF	gold	FFD700	navajowhite	FFDEAD
aquamarine	7FFFD4	goldenrod	DAA520	navy	000080
azure	F0FFFF	gray	808080	oldlace	FDF5E6
beige	F5F5DC	grey	808080	olive	808000
bisque	FFE4C4	green	008000	olivedrab	6B8E23
black	000000	greenyellow	ADFF2F	orange	FFA500
blanchedalmond	FFEBCD	honeydew	F0FFD0	orangered	FF4500
blue	0000FF	hotpink	FF69B4	orchid	DA70D6
blueviolet	8A2BE2	indianred	CD5C5C	palegoldenrod	EEE8AA
brown	A52A2A	indigo	4B0082	palegreen	98FB98
burlywood	DEB887	ivory	FFFFF0	paleturquoise	AFEEEE
cadetblue	5F9EAO	khaki	F0E68C	palevioletred	DB7093
chartreuse	7FFF00	lavender	E6E6FA	papayawhip	FFEDD5
chocolate	D2691E	lavenderblush	FFF0F5	peachpuff	FFDAB9
coral	FF7F50	lawngreen	7CFC00	peru	CD853F
cornflowerblue	6495ED	lemonchiffon	FFFACD	pink	FFC0CB
cornsilk	FFF8DC	lightblue	ADD8E6	plum	DDA0DD
crimson	DC143C	lightcoral	F08080	powderblue	BOE0E6
cyan	00FFFF	lightcyan	E0FFFF	purple	800080
darkblue	00008B	lightgoldenrodyellow	FADAD2	red	FF0000
darkcyan	008B8B	lightgray	D3D3D3	rosybrown	BC8F8F
darkgoldenrod	B8860B	lightgreen	90EE90	royalblue	4169E1
darkgray	A9A9A9	lightgrey	D3D3D3	saddlebrown	8B4513
darkgreen	006400	lightpink	FFB6C1	salmon	F8BBD0
darkgrey	A9A9A9	lightsalmon	FFA07A	sandybrown	F4A460
darkkhaki	BDB76B	lightseagreen	20B2AA	seagreen	2E8B57
darkmagenta	8B008B	lightskyblue	87CEFA	seashell	FFF5EE
darkolivegreen	556B2F	lightslategray	778899	sienna	A0522D
darkorange	FF8C00	lightslategray	778899	silver	C0C0C0
darkorchid	9932CC	lightsteelblue	BOC4DE	skyblue	87CEEB
darkred	8B0000	lightyellow	FFFFE0	slateblue	6A5ACD
darksalmon	E9967A	lime	00FF00	slategray	708090
darkseagreen	8FBC8F	limegreen	32CD32	slategray	708090
darkslateblue	483D8B	linen	FAFAD2	snow	FFFAFA
darkslategray	2F4F4F	magenta	FF00FF	springgreen	00FF7F
darkslategray	2F4F4F	maroon	800000	steelblue	4682B4
darkturquoise	00CED1	mediumaquamarine	66CDAA	tan	D2B48C
darkviolet	9400D3	mediumblue	0000CD	teal	008080
deeppink	FF1493	mediumorchid	BA55D3	thistle	D8BFD8
deepskyblue	00BFFF	mediumpurple	9370DB	tomato	FF6347
dimgray	696969	mediumseagreen	3CB371	turquoise	40E0D0
dimgray	696969	mediumslateblue	7B68EE	violet	EE82EE
dodgerblue	1E90FF	mediumspringgreen	00FA9A	wheat	F5DEB3
firebrick	B22222	mediumturquoise	48D1CC	white	FFFFFF
floralwhite	FFFACD	mediumvioletred	C71585	whitesmoke	F5F5F5
forestgreen	228B22	midnightblue	191970	yellow	FFFF00
fuchsia	FF00FF	mintcream	F5FFFA	yellowgreen	9ACD32

See also

[Bind a property](#) on [page 27](#)

Color Picker overview

An overview of the **Color Picker**:



Item	Name	Purpose
1	System color swatches	Standard system colors.
2	Favorite swatches	A set of pre-configured colors. Stores colors and gradients that the user creates.
3	Custom/Collapse custom	Shows or hides the spectrum color picker and the gradient color selector.
4	Create solid color	Applies a single color evenly.
5	Create gradient color	Opens the gradient color selector below the spectrum color picker.
6	Spectrum color picker	Selects a custom color.
7	Spectrum color	Shows the color selected in the spectrum color picker.
8	Current color	Shows the currently selected color.
9	Gradient selector	Creates a gradient color. Stops at either end of the gradient selector for selecting the starting and ending color.
10	Radial Gradient	Creates a radial gradient using the selected gradient stop colors. Fades and blends colors in a circular path outward from a center point. The leftmost stop selects the center color.
11	Linear Gradient	Creates a linear gradient using the selected gradient stop colors. Fades and blends color along a straight line.
12	Selected color	Shows the selected color or gradient.
13	R, G, B	Shows the proportion of red, green, and blue in the selected color. Also allows manual adjustment of the proportion value.

14	Add color swatch	Adds a custom color or gradient to the Favorite swatches.
-----------	------------------	---

See also

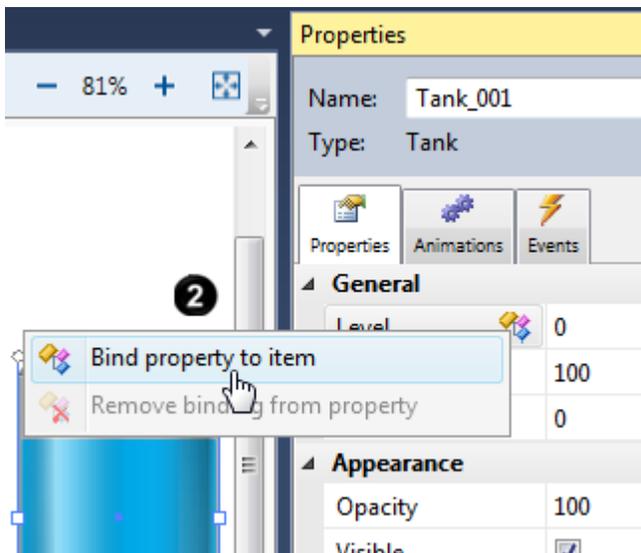
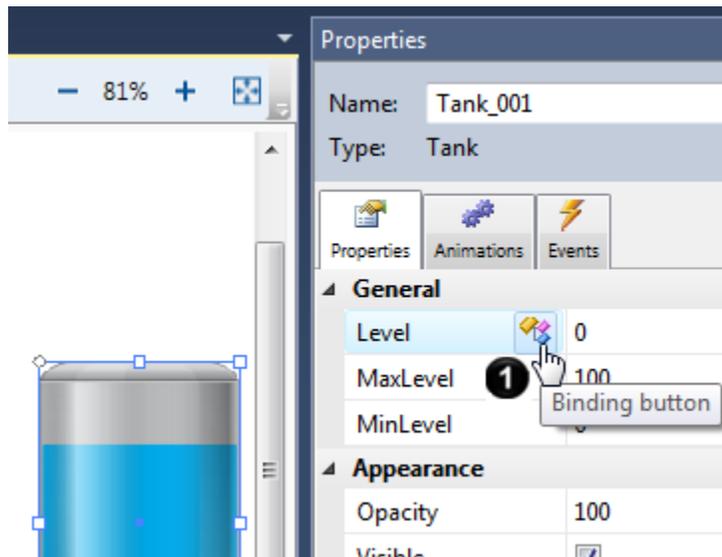
[Example 1: Bind a property on page 32](#)

[Example 2: Bind graphic element properties to other properties on page 34](#)

[Expression overview on page 35](#)

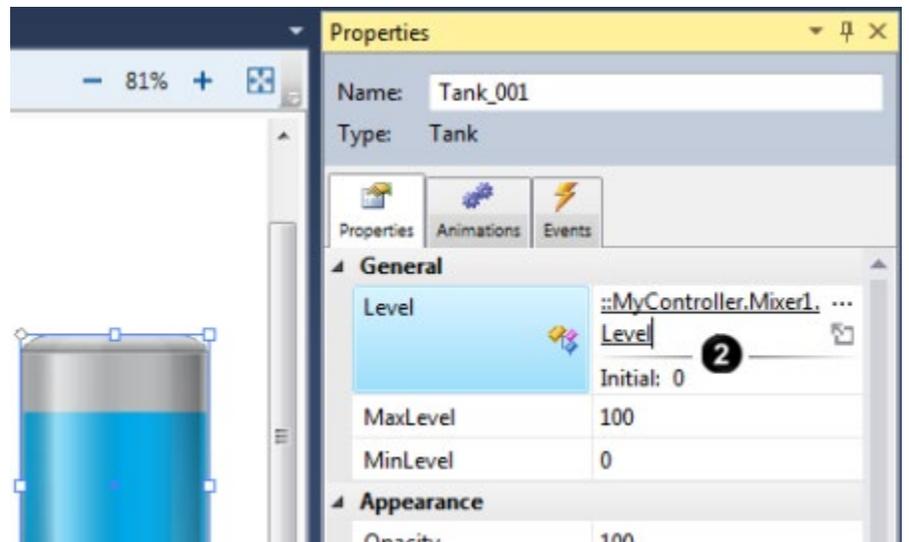
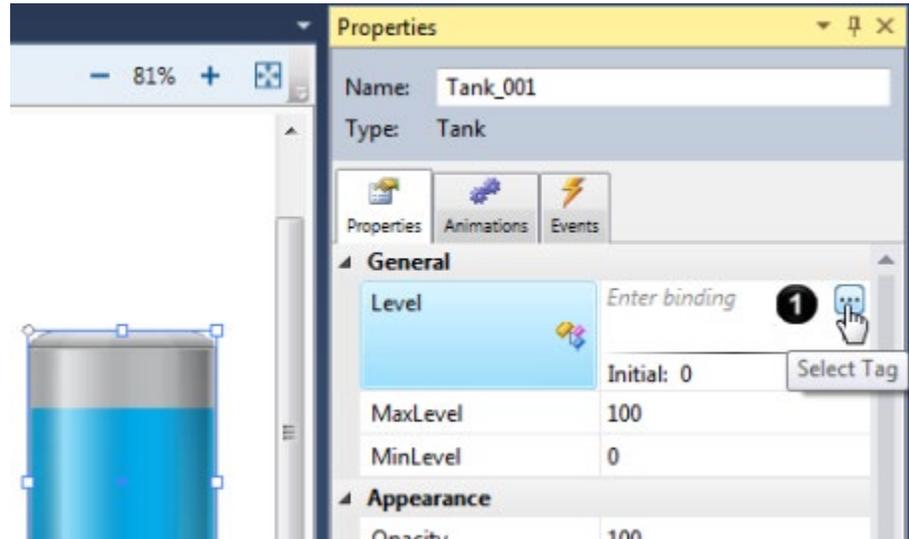
Example 1: Bind a property

Many graphic elements in the View Designer **Toolbox** have built-in animations. Use animations by binding to their properties. For example, bind a tag or an expression to the **Level** property of the Tank element to animate the level shown on the screen. To create a binding, hover on the property in the **Properties** pane, click the **Binding** button  and select **Bind property to item**.



Item	Description
1	Binding button
2	Bind property to item

Use the tag browser to select a tag for binding to the property. As an option, use expression operators in combinations with tags for more complex bindings.



Item	Description
1	Select Tag
2	Add expression

Bindings are available for most of the properties of an element. For example, move the tank on the screen by binding the X and Y properties of the tank. If

rotating the tank, bind to the **Angle** property of the tank. Have the tank fade in and out by binding to the **Opacity** property of the tank.

See also

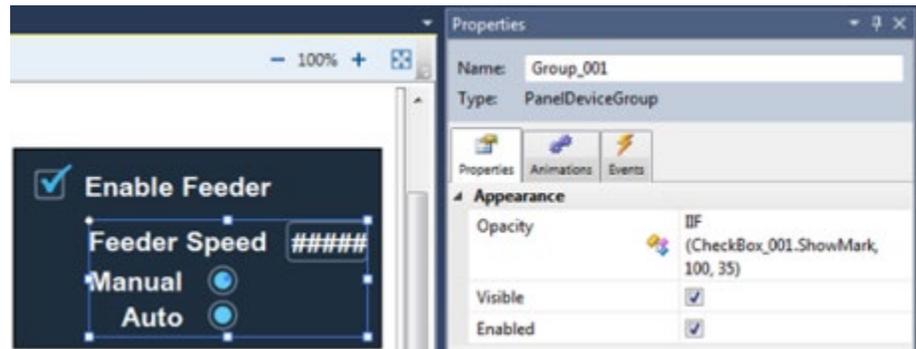
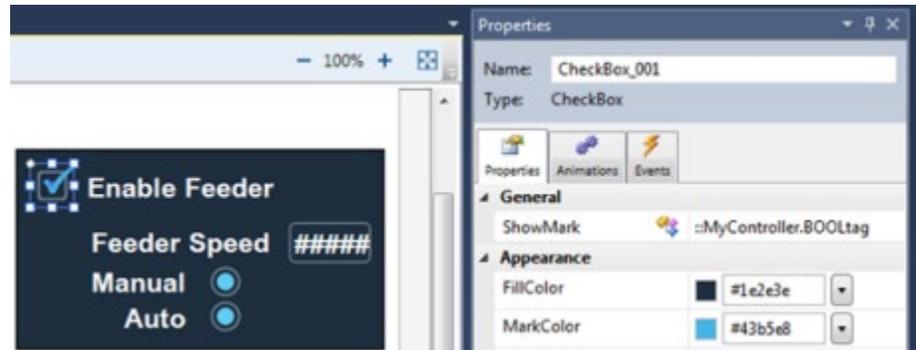
[Bind a property](#) on [page 27](#)

[Property Binding](#) on [page 27](#)

Example 2: Bind graphic element properties to other properties

Creating bindings between graphical elements on a screen to have the property of one element affect other graphic elements.

In this example, use a checkbox to make a group of graphic elements appear disabled when the user clears the checkbox. Use an expression to change the opacity of the group of elements based on the **ShowMark** property of the checkbox.



At runtime, clearing the checkbox changes the opacity of the group of elements.



See also

[Bind a property](#) on [page 27](#)

[Property binding](#) on [page 27](#)

Expression overview

Use the **Expression Editor** to create or edit lengthy or complex expressions in a resizable window.

Most expressions consist of one or more tag references combined with numbers, mathematical operators, or built-in functions. Expressions do not need a tag.

View Designer checks the expression syntax as you create the expression. A red outline appears around the expression with an invalid syntax. The red outline disappears when the syntax becomes valid.

View Designer also verifies the syntax for all expressions in a project before downloading to the HMI device. In addition, it verifies that tag references exist in the controller project. It also verifies the property of a graphic element referenced in an expression exists on the screen with the graphic element. If there are invalid expressions, the download process stops, and the invalid expressions appear in the Errors window. Verify expressions any time during the project development process by using the **PROJECT > Verify Project** command.

See also

[Expression examples](#) on [page 35](#)

Expression examples

Examples of combinations to use for creating an expression:

Numeric literals in expressions

Name	Description	Example
Integers	Numbers 0-9 specified in base-10 format. Leading zeros are not available. 0 is available, 01 is not available.	123
Floating point	Numbers that begin with zero or more base-10 digits, then the period character as decimal separator, followed by one or more base-10 digits. Leading zeros are not available to the left of the decimal point, unless it is just a single zero immediately preceding the decimal point. Examples: 1.23 .23 0.23	1.23, .23, and 123.45

Exponential	Numbers that begin with an integer or floating-point number, followed by the character 'e' (or 'E') and an integer. A negation operator '-' is available before the exponent. Other operators are not available.	$6.02e23$ means $6.02 * 10^{23}$ $1.66e-27$ means $1.66 * 10^{-27}$
-------------	--	--

String values (literal or tags) in expressions

Name	Description	Example: If tag1 = "SOME" and tag2 + "THING", then:
String	<p>A string literal begins with a double-quote character and ends with a double-quote character. Anything between the two double-quotes is part of the string literal. You can use strings as operands with the plus (+) operator and with relational operators. The + operator joins string operands. String tags and string literals behave the same way.</p> <p>Keep the following in mind when using string tags with the relational operators:</p> <ul style="list-style-type: none"> • All relational operators are case-sensitive. • String comparisons occur on a character-by-character basis. • Compares individual letters based on their Unicode value. This means that a lowercase letter is greater than its corresponding uppercase letter. • If two strings are different lengths and the characters are the same, the longer string is greater than the shorter string. 	$tag1 + tag2$ returns "SOMETHING"

Tags in expressions

Name	Description	Example: In a controller named Packing, if tag1 = 5 and tag2 = 7, then:

Tag	<p>A tag can stand alone as an expression exist as part of an expression that consists of other components. An expression that is a tag name returns the value of the tag. Tags begin with either "::" or "\". Use the format</p> <pre>::ControllerName.TagName.</pre> <p>To find a tag, type the tag name or open the Tag Browser and select a tag.</p>	<pre>::Packing.tag1 returns 5 ::Packing.tag1 + 25 returns 30 ::Packing.tag1 % ::Packing.tag2 returns 5 ::Packing.tag1>: :Packing.tag2 returns 0 (false) (::Packing.tag1< ::Packing.tag2) && (::Packing.tag 1==5) returns 1(both statements are true)</pre>
-----	--	--

Numeric operators in expressions

- IMPORTANT**
- Avoid using strings in numeric operations with the exception is the + operator.
 - Avoid using LINT tags in expressions, write commands, or increment commands. The values may lose resolution after performing the expression calculations.
 - Tag values that are divisors cannot have a value of zero at any point. Expressions that try to divide a number by zero produce an error at runtime.

Symbol (Operation)	Description	Example: If tag1 = 5 and tag2 = 7, then:
+ (Addition)	Returns the sum of the values. Use SINT, INT, DINT, LINT, and REAL tags in any order. All values convert to double-precision floating point before performing the operation. If any operand is a string, it is concatenated.	tag1 + tag2 returns a value of 12 MyDINT + "SomeString" returns a value of 5SomeString, if MyDINT = 5
- (Subtraction)	Returns the difference of the two values. Use SINT, INT, DINT, LINT, and REAL tags in any order. All values convert to double-precision floating point first before performing the operation.	tag1 - tag2 returns a value of -2
- (Negation)	Inverts the sign of the operand, making a positive number into a negative number or a negative number into a positive number.	-tag2 returns -7
* (Multiplication)	Returns the product of two values. Use SINT, INT, DINT, LINT, and REAL tags in any order. All values convert to double-precision floating-point before performing the operation.	tag1 * tag2 returns a value of 35

/ (Division)	Returns the quotient of the two values. Use SINT, INT, DINT, LINT, and REAL tags in any order. All values convert to double-precision floating point before performing the operation.	tag1 / tag2 returns a value of 0.71
% (Modulus)	Returns the rest of one number divided by another. Use SINT, INT, DINT, LINT, and REAL tags in any order. All values convert to double-precision floating point first before performing the operation.	tag1 % tag2 returns a value of 5.

Math Function in expressions

Function	Description	Example
ABS, Abs, or abs	Returns the absolute value of the expression.	ABS (-1.23) returns 1.23. ABS (1.23) returns 1.23.
ARCCOS, arccos, ACOS, or acos	Returns the arc cosine of the expression in radians.	ACOS (-1.0) returns 3.14159.
ARCSIN, arcsin, ASIN, or asin	Returns the arc sine of the expression in radians.	ARCSIN (1) returns 1.570796.
ARCTAN, arctan, ATAN, or atan	Returns the arc tan of the expression in radians.	ATAN (-45.01) returns -1.54858.
COS, Cos, or cos	Returns the cosine of the expression in radians.	COS (14.78) returns -0.599465.
LOG, Log, or log	Returns the natural log of the expression.	LOG (2) returns 0.69314718
LOG10, Log10, or log10	Returns the base-10 log of the parameter.	LOG10 (100) returns 2.
SIN, Sin, or sin	Returns the sine of the expression in radians.	SIN (45.175643) returns 0.929607.
SQRT, Sqrt, or sqrt	Returns the square root of an expression.	SQRT (144) returns 12.
TAN, Tan, or tan	Returns the tangent of the expression in radians.	TAN (1) returns 1.5574077246549023.
TRUNC, Trunc, or trunc	Returns the value of the parameter with any digits to the right of the decimal point removed.	TRUNC (10.8282) returns 10.
POW, pow	Returns the result of raising the first value to the power of the second value. All values convert to double-precision floating point before performing the operation.	POW (10, 3) returns 1000.

String functions in expressions

Function	Description	Example
TOLOWER, ToLower, or tolower	Converts any uppercase characters to lower lowercase. Converts a numeric to a string that has the number.	TOLOWER ("New York") returns "new york". This expression converts a numeric to a string containing that number: TOLOWER (32) + TOLOWER (34) returns "3234".
TOUPPER, ToUpper, or toupper	Converts any lowercase characters to uppercase.	TOUPPER ("hello") returns "HELLO".

Conditional statements in expressions

IMPORTANT

Keep the following in mind when using conditional statements:

- The condition portion of the statement can be any numeric or Boolean expression.
- If the condition is a numeric expression, including a tag, zero is treated as false and any nonzero value is treated as true.
- Avoid using a string operand (string Literals, string properties, string tags, or the result of a String Operation) as the condition of a conditional statement.
- Expressions need all parts of a conditional statement.
- The value if true and value if false portions of the conditional statement can be any valid expression, including another conditional statement.

Function	Description	Example: If tag1 = 5 and tag2 = 7, then:
IIF, Iif, iif	Evaluates with the same rules as numeric operands. If the condition evaluates to 0 (false), the function uses the second value. If the condition evaluates to nonzero (true), the first value is used. Use this format: IIF(condition, value if true, value if false)	IIF(tag1 > tag2, tag1, tag2) returns 7 (the value of tag2) because tag1 is not greater than tag2
?:	The following requirements are specific to the ?: statement: <ul style="list-style-type: none"> • The conditional operator is right-associative. Therefore, a ? b : c ? d : e is evaluated as a ? b : (c ? d : e). • White space or parentheses must surround the segments of a conditional operation. • A new line counts as white space. Use this format: condition ? value if true : value if false	(tag1 > tag2) ? tag1 : tag2 returns 7 (the value of tag2) because tag1 is not greater than tag2

Relational operators in expressions

Symbol (Function)	Description	Example: If tag1 = 5 and tag2 = 7, then:
== (Equal to)	Compares the two values and returns a 1 (true) if they are equal. Returns a 0 (false) if they are not equal. Performs a case-sensitive string comparison. All values convert to double-precision floating point before performing the operation.	<code>tag1 == tag2</code> is false, so the expression returns 0
!= (Not equal to)	Compares the two values and returns a 1 (true) if they are not equal. Returns a 0 (false) if they are equal. All values convert to double-precision floating point before performing the operation.	<code>tag1 != tag2</code> is true, so the expression returns 1
< (Less than)	Compares two values and returns a 1 (true) if the value on the left is smaller than the value on the right. Returns a 0 (false) if not. All values convert to double-precision floating point before performing the operation.	<code>tag1 < tag2</code> is true, so the expression returns 1
> (Greater than)	Compares two values and returns a 1 (true) if the value on the left is larger than the value on the right. Returns a 0 (false) if not. All values convert to double-precision floating point before performing the operation.	<code>tag1 > tag2</code> is false, so the expression returns 0
<= (Less than or equal to)	Compares two values and returns a 1 (true) if the value on the left is smaller or the same as the value on the right. Returns a 0 (false) if not. All values convert to double-precision floating point before performing the operation.	<code>tag1 <= tag2</code> is true, so the expression returns 1
>= (Greater than or equal to)	Compares two values and returns a 1 (true) if the value on the left is larger or the same as the value on the right. Returns a 0 (false) if not. All values convert to double-precision floating point before performing the operation.	<code>tag1 >= tag2</code> is false, so the expression returns 0

Logical operators in expressions

Symbol (Function)	Description	Example: If tag1 = 5 and tag2 = 7, then:
&& (Logical AND)	Returns a value of 1 if the statements to the right and to the left of the operator are both true (nonzero).	<code>(tag1 < tag2) && (tag1 == 5)</code> returns a 1 because both statements are nonzero (true) <code>tag1 && tag2</code> returns a 1 because both tag1 and tag2 are nonzero (true)

(Logical OR)	Returns a value of 1 if either the statement to the left or to the right of the operator is true (nonzero).	<code>(tag1 > tag2) (tag1 == 5)</code> returns a value of 1 because <code>tag1 == 5</code> is true
!(Logical NOT)	Inverts the Boolean value of an expression. Returns true if the expression is false and returns false if the expression is true. If the expression <code>(a>b)</code> evaluates to true, then <code>!(a>b)</code> evaluates to false.	<code>!(tag1 < tag2)</code> The expression <code>tag1 < tag2</code> evaluates to true and returns a value of 1, but the NOT operator reverses the logical value, and returns 0.

Bitwise operators in expressions

Symbol (Function)	Description	Example: If <code>tag1 = 5</code> (binary <code>0000 0000 0000 0101</code>) and <code>tag2 = 3</code> (binary <code>0000 0000 0000 0011</code>), then:
& (Bitwise AND)	Returns an integer with a bit set to 1 if both corresponding bits in the original numbers are 1. Otherwise, the resulting bit is 0.	<code>tag1 & tag2</code> returns 1 (binary <code>0000 0000 0000 0001</code>)
(Bitwise inclusive OR)	Returns an integer with a bit set to 1 if either or both corresponding bits in the original numbers are 1. If both bits are 0, the resulting bit is 0.	<code>tag1 tag2</code> returns 7 (binary <code>0000 0000 0000 0111</code>)
^ (Bitwise exclusive XOR)	Returns an integer with a bit set to 1 if either of the corresponding bits in the original numbers is 1. If both bits are 1 or both are 0, the resulting bit is 0.	<code>tag1 ^ tag2</code> returns 6 (binary <code>0000 0000 0000 0110</code>)
>> (Right Shift)	Shifts the bits within the left operand by the amount specified in the right operand. The bit on the right disappears. Either a 0 or a 1 shift on the left depending on whether the integer is signed or unsigned. With unsigned integers, 0 always shifts on the left. With signed integers, a 0 shifts when the number is positive (that is, the leftmost bit--the sign bit--is 0), and a 1 shifts when the number is negative (that is, the leftmost bit--the sign bit--is 1). In other words, with signed integers, the sign of the number is always maintained.	<code>tag1 >> 1</code> returns 2 (binary <code>0000 0000 0000 0010</code>)

<< (Left Shift)	<p>Shifts the bits within the left operand by the amount specified in the right operand. The bit on the left disappears and a 0 shifts on the right.</p> <p>If the left bit is a 1, an overflow occurs, and an error message appears. To help prevent this, use the bitwise AND (&&) operator in an expression.</p> <p>For example, (dev << 1) && 65535, where 65535 is 1111 1111 1111 1111 in binary form.</p> <p>Where dev is a tag name whose value is shifting left.</p>	<p>tag1 << 1 returns 10 (binary 0000 0000 0000 1010)</p>
~ (Complement)	<p>Gives the ones complement of a number.</p> <p>For example, use this operator to reverse every bit within the number so that every 1 bit becomes a 0 and vice versa.</p>	<p>~ tag1 returns -6 (binary 1111 1111 1111 1010)</p>

Properties of graphic elements in expressions

IMPORTANT Avoid setting up circular references using graphic element property bindings because this can result in unpredictable values or a runtime error.

An example of a circular reference would be having three elements on the screen such that, for example, Element1.Opacity is bound to Element2.Opacity, Element2.Opacity is bound to Element3.Opacity, and Element3.Opacity is bound to Element1.Opacity.

Name	Description	Example
Property name	<p>Property names refer to components of the current application. This differs from tags, which refer to data items in a controller.</p> <p>The graphic elements must all be on the same screen, popup, or System Banner.</p> <p>Expressions do not support alias properties.</p> <p>Use the syntax: ElementName.PropertyName</p> <p>An expression can have:</p> <ul style="list-style-type: none"> • Properties that support bindings. • Multiple graphic element properties. • Graphic element property bindings AND tags AND user-defined properties or any combination. 	<p>MyElement.X + MyElement.Y / MyElement.Opacity</p> <p>MyElement.X + ::MyController.MyTag + MyCustomProperty</p>

Punctuators in expressions

Symbol	Description	Example
()	Opening and closing parentheses. Use parentheses around any subexpression to specify the order of operations. Subexpressions inside parentheses evaluate before applying operators outside the parentheses.	$3 + 2 * 6$ returns 15 $(3 + 2) * 6$ returns 30

See also

[Expression overview](#) on [page 35](#)

Color and state tables

Use color and state tables to define step-wise animations on a graphic element. This allows any graphic element to act as a multi-state indicator.

-
- IMPORTANT**
- Do not use a State Table or Color Table to change a property that is already bound to a tag or expression. This can cause undesired results. Property values change based on which input changes the property first. There is no way to determine the order of inputs.
 - Modifying a property across multiple State Tables or Color Tables causes undesired results. Property value changes occur based on which input changes the property first.
 - When renaming a graphic element that a State Table or a Color Table references, change the reference to reflect the new graphic element name. View Designer does not automatically update the references with the new name.
 - Do not use these words as a state name:
 - state
 - states
 - properties
 - readonly
 - parent
 - type
 - name
 - id
 - persisted
 - objectname
 - Do not configure a State Table that causes the states to act on each other in an infinite loop. This creates slow performance.
 - The HMI device updates data from controllers at a default rate of 500 milliseconds. You can change the update rate of screens, popups and the System banner to be 100, 250 or 500 milliseconds. This update is asynchronous to any scan in a controller.
-

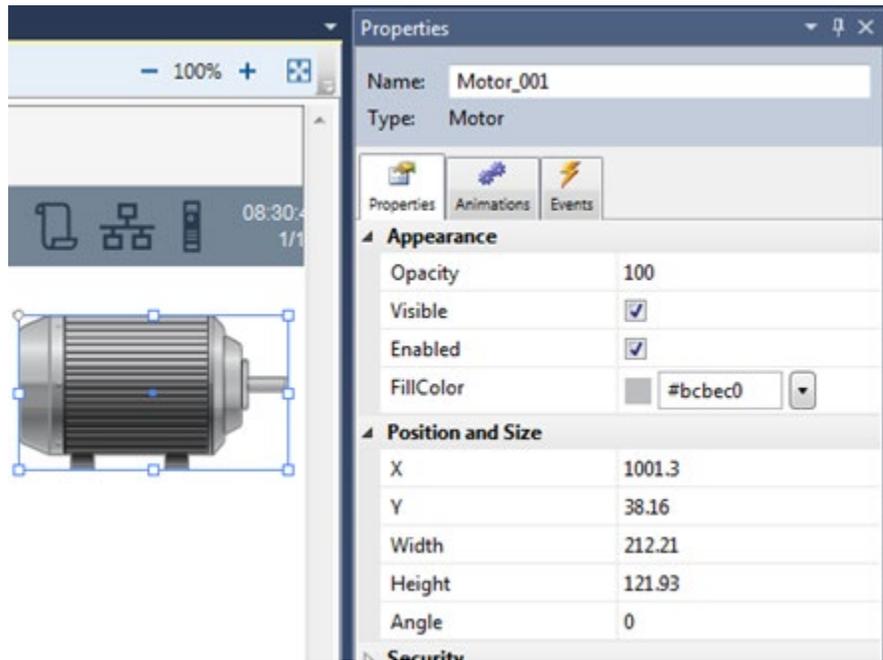
See also

[Example 1: Use a state table to make a motor a multi-state indicator](#) on [page 45](#)

[Example 2: Use a color table to indicate level in a mixer](#) on [page 47](#)

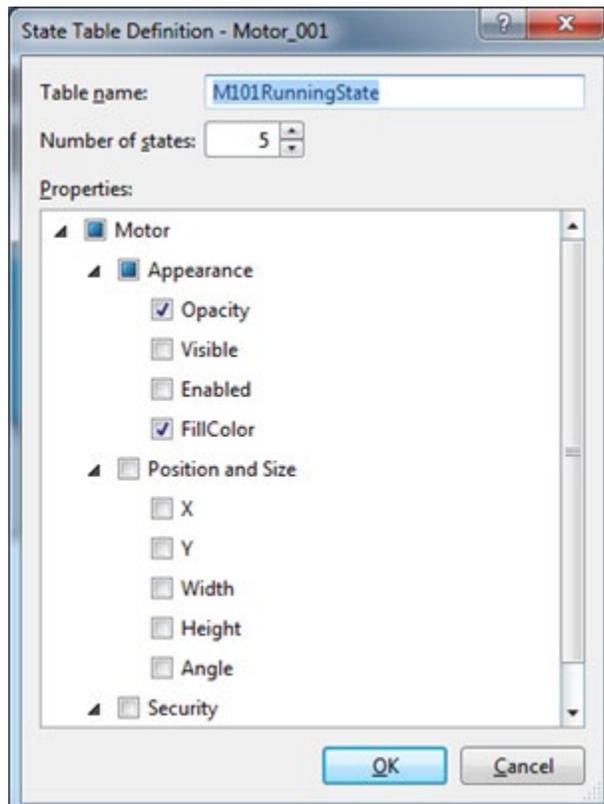
Example 1: Use a state table to make a motor a multi-state indicator

Place a motor graphic element on the screen. The motor has properties, such as **FillColor** and **Opacity**.

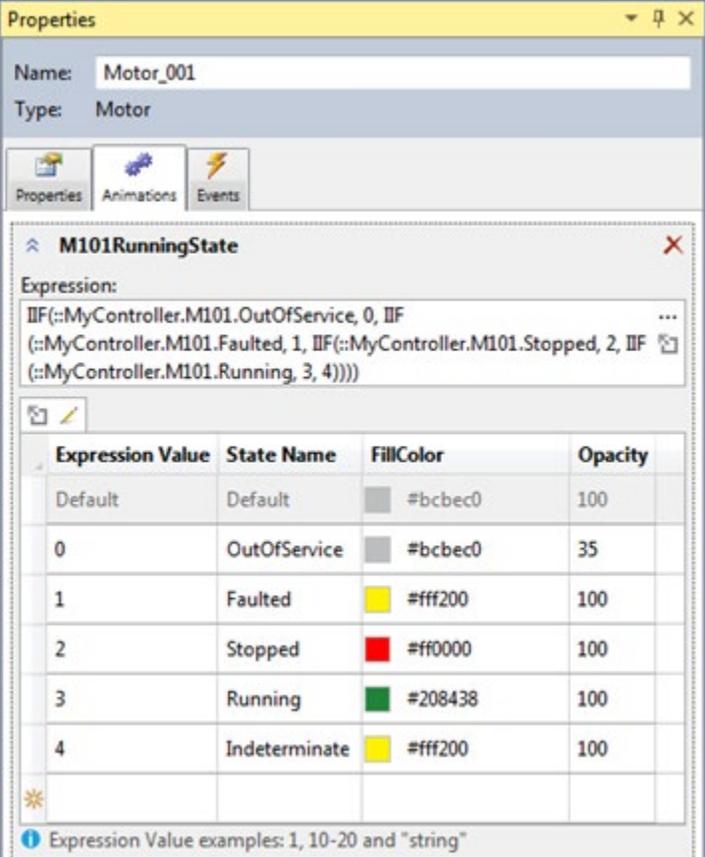


Click the **Animations** tab and select **State Table** to create a state table to manipulate the properties in a state-wise fashion.

Use the **State Table Definition** dialog box to define a name for the state table, the number of states, and the properties to manipulate. Change these values at any time. For example, to show the running state of the motor, select **Opacity** and **FillColor**.



Use the **Animations** tab to further define the State Table. In this example, read bits out of the controller to determine the state of the motor and change the color and opacity of the motor to represent those states. Name the states to make the state table easier to understand.



Expression Value	State Name	FillColor	Opacity
Default	Default	#bcbec0	100
0	OutOfService	#bcbec0	35
1	Faulted	#fff200	100
2	Stopped	#ff0000	100
3	Running	#208438	100
4	Indeterminate	#fff200	100

Expression Value examples: 1, 10-20 and "string"

If desired, configure state tables to trigger commands when entering or leaving states. See the *Events* chapter.

See also

[Example 2: Use a color table to indicate level in a mixer](#) on [page 47](#)

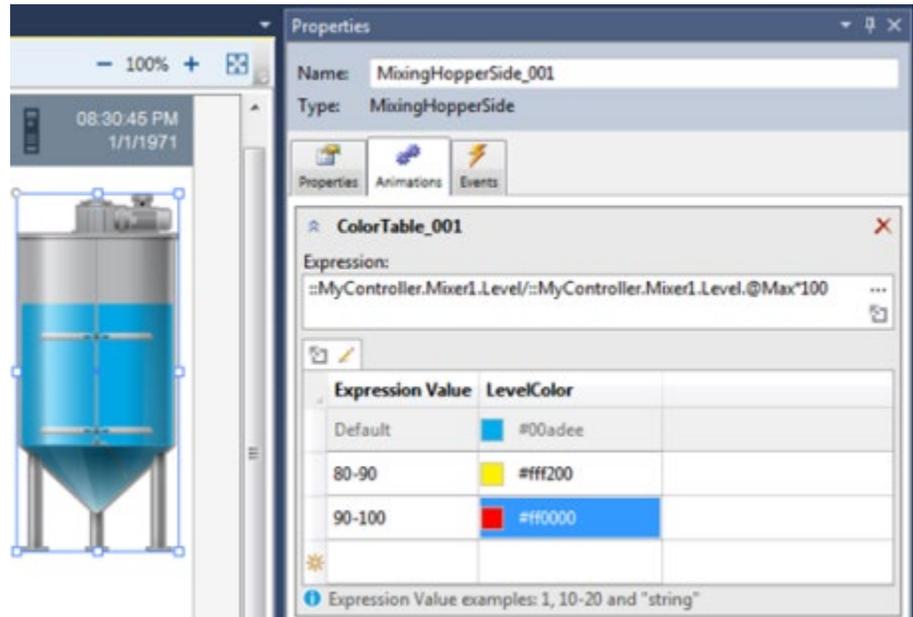
[Color and state tables](#) on [page 45](#)

[Events](#) on [page 51](#)

Example 2: Use a color table to indicate level in a mixer

Color tables are a subset of state tables that use only the color properties. Use color tables to create common color animations.

Color and state tables support applying ranges to expression values. For example, if changing the color of a mixer tank fill based on the level of the mixer, configure a color table so that when the level changes between 80-90% and 90-100%, the color changes from blue to yellow to red.



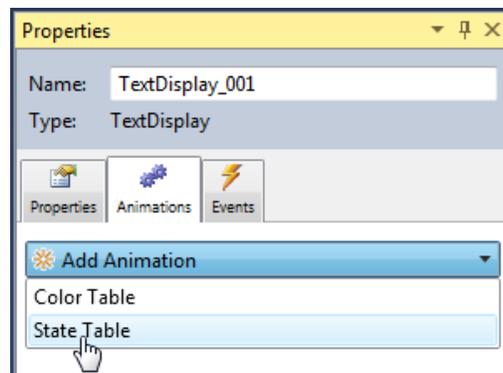
See also

[Events](#) on [page 51](#)

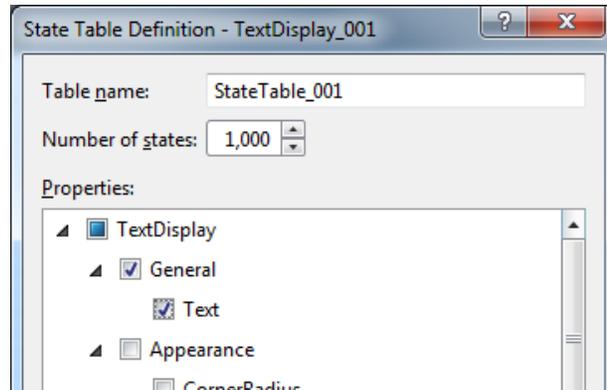
Example 3: Use a state table to display fault code text

State tables can support up to 2500 states. Use a text display with many states to convert a fault code into a text display for the operator. For example, if an integer tag in the controller represents 1000 different fault codes returned from a piece of equipment, then a state table on a text display can convert those fault codes into text.

To make a state table display fault code text, first select the Animations tab for the text display and under Add Animation, select State Table:

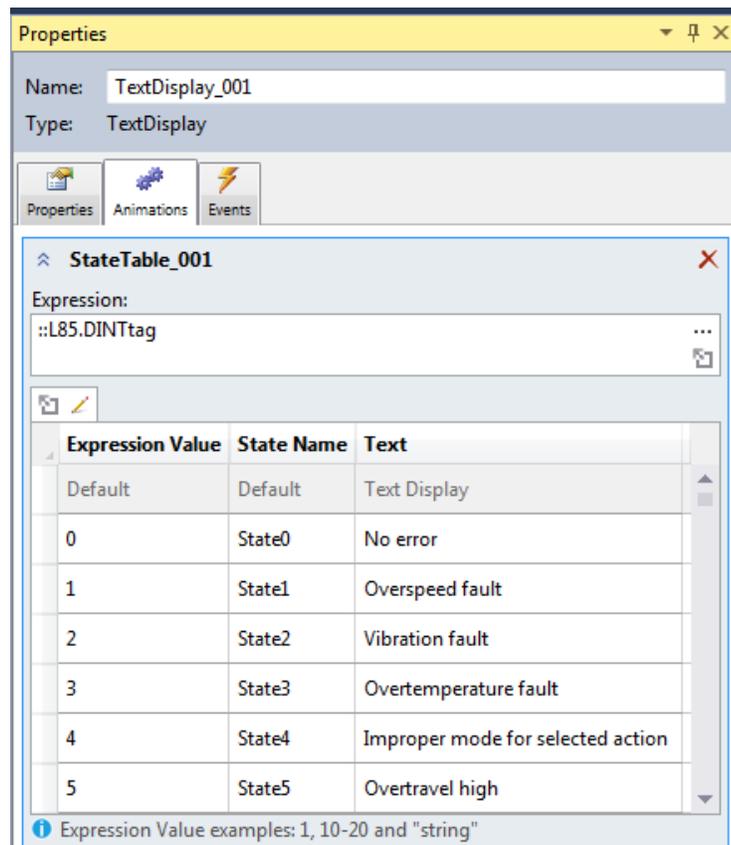


In the State Table Definition window, provide a name for the table, set the Number of states to 1000 and select the Text property:



Then select OK. A state table appears in the Animations tab.

Define a tag or expression that contains the error code and modify the Text column to match the error string to the error code:



Tip: You can copy and paste multiple cell values from Microsoft Excel into a state table in View Designer:

1. Select a specific number of cells in a spreadsheet column and use the Copy command.
2. Select the same number of cells in the Text column of the state table and Paste.

Adding many properties to a large state table may cause slow performance of View Designer and increase the time of switching between screens at runtime.

Note that a single screen can support up to 19 large state tables. A larger number of large state tables can cause a fault in View Designer.

Events

An event occurs to trigger one or more commands. An example of an event is touching a key on a keypad. Apply events to any graphic element. Events are not limited to specific button objects. View Designer enables creating more intuitive screens, such as operators touching pieces of equipment for navigating screens or popups with more detail about that equipment.

View Designer enables creating more complex events and commands, such as defining multiple events for a graphic element. For example, define a command to execute when pressing a button. Then define a different command to execute when releasing the button. Define multiple commands for any event as necessary. For example, change the values of several tags when pressing or releasing a button.

These are the event types.

Event Type	Triggers	Common uses
Button Behavior	Touch or a key press.	Configure common events and commands on a graphic element. To quickly access Button Behavior , right-click a graphic element and select Button Behavior .
Touch Press	Touch a graphic element, Add-On Graphic, screen, popup, or System Banner on the HMI device.	Initiate commands when pressing a graphic element.
Touch Release	Release a graphic element, Add-On Graphic, screen, popup, or System Banner on the HMI device.	Initiate commands when releasing a graphic element. Use Touch Release events to change a course of action as necessary. If sliding a finger off the element when pressing the element, the Touch Release event does not trigger. To disable this behavior, select Always Trigger Release Event on the Touch Release confirmation.

Event Type	Triggers	Common uses
Key Press	Touch a key on the HMI device physical keypad or an external keyboard	Initiate commands when pressing a key on the keypad. Configure any of the L1 to L10 or R1 to R10 keys. Tip: On an external keyboard, shift F1 through Shift F10 corresponds to L1 through L10 and Ctrl F1 through Ctrl F10 corresponds to R1 through R10.
Key Release	Release a key on the HMI device physical keypad or an external keyboard	Initiate commands when releasing a key on the keypad. Configure any of the L1 to L10 or R1 to R10 keys. Tip: On an external keyboard, shift F1 through Shift F10 corresponds to L1 through L10 and Ctrl F1 through Ctrl F10 corresponds to R1 through R10.
State Enter	When a configured state table enters a configured state	Initiate commands when a State Table enters or leaves a configured state. For example, use State Enter and State Exit events to have animations on the screen set values in the Logix controller.
State Exit	When a configured state table leaves a configured state.	

See also

[Example 1: Configure button behaviors](#) on [page 52](#)

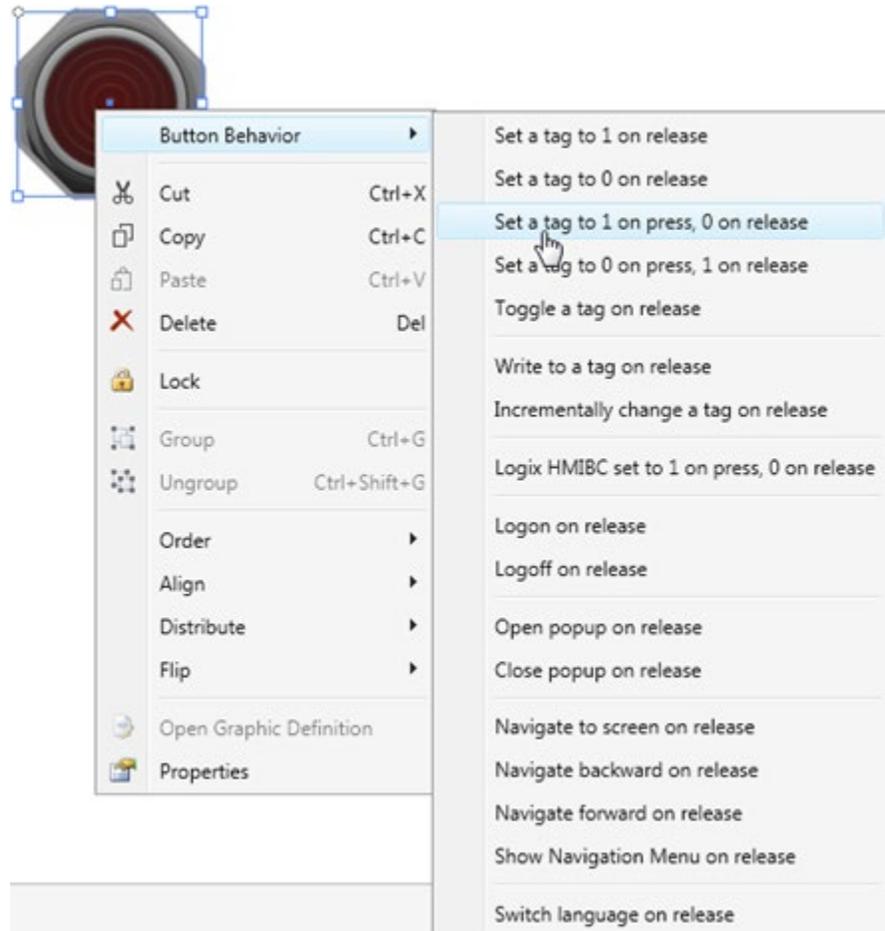
[Example 2: Configure multiple events on a graphic element](#) on [page 54](#)

[Example 3: Use State Enter and State Exit Events](#) on [page 59](#)

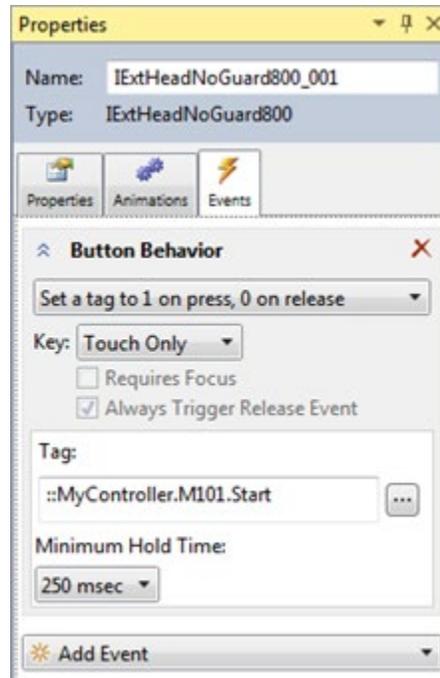
[Example 4: Use key press and key release events](#) on [page 62](#)

Example 1: Configure button behaviors

Button Behaviors are predefined combinations of events and commands that perform common actions. For example, place a button graphic element on the screen to work as a momentary button. When an operator presses the button, it sets a bit, and when the operator releases it, it resets the bit. Insert the button element on the screen. Right-click the button and select **Button Behavior**. When a list of button actions appears, select **Set a tag to 1 on press, 0 on release** to trigger a momentary behavior.



In the **Events** tab for the graphic element, configure the **Button Behavior**.



For this specific button behavior, the **Always Trigger Release Event** configuration is selected and read-only. The momentary action sets the M101.Start bit when pressing the button. The **Always Trigger Release Event** configuration ensures that the M101.Start bit resets when the operator stops pressing the button, or when a **State Enter, State Exit** or **Project Event** causes a navigation to a different screen.

See also

[Example 2: Configure multiple events on a graphic element](#) on [page 54](#)

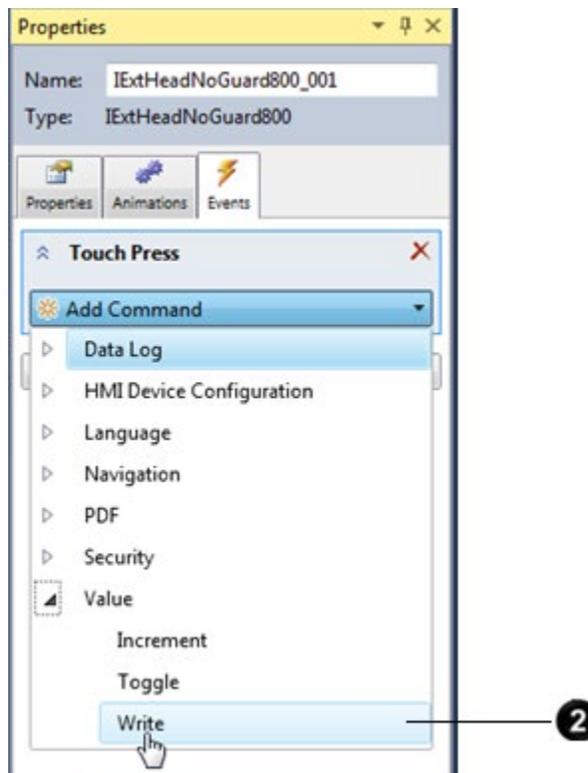
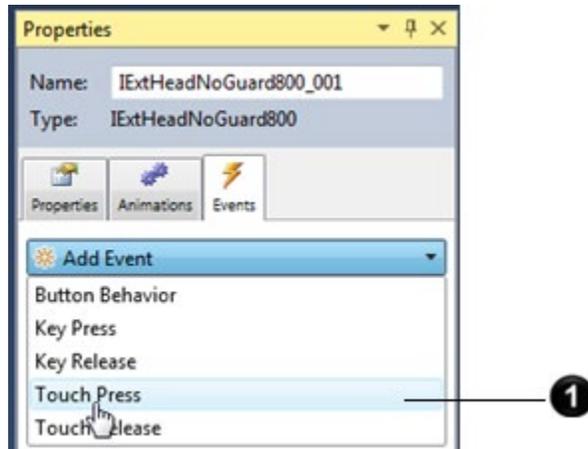
[Example 3: Use State Enter and State Exit Events](#) on [page 59](#)

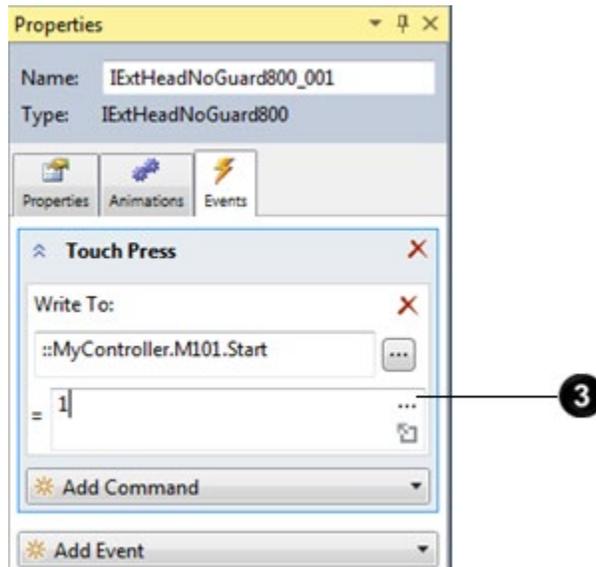
[Example 4: Use key press and key release events](#) on [page 62](#)

Example 2: Configure multiple events on a graphic element

View Designer enables defining multiple commands for any event. If necessary, create more complex events and commands, such as defining multiple events for a graphic element. For example, define a command to execute when pressing a button. Define a different command to execute when releasing the button.

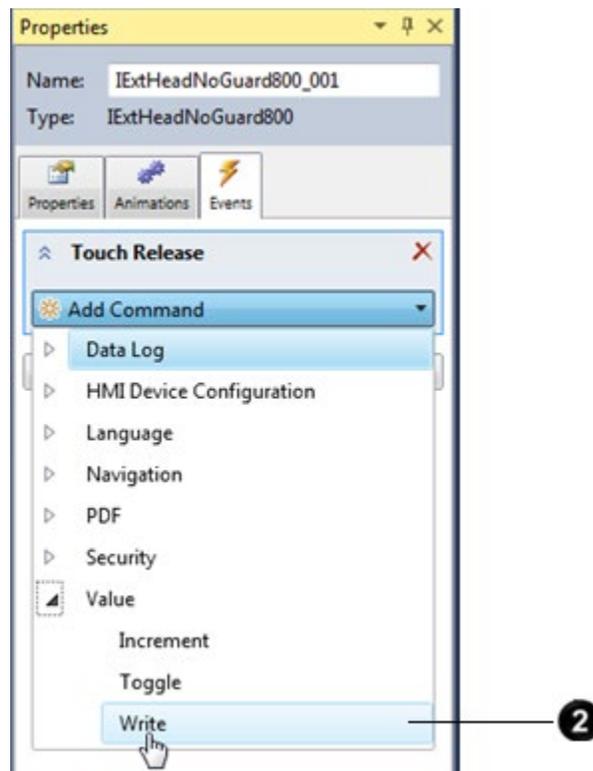
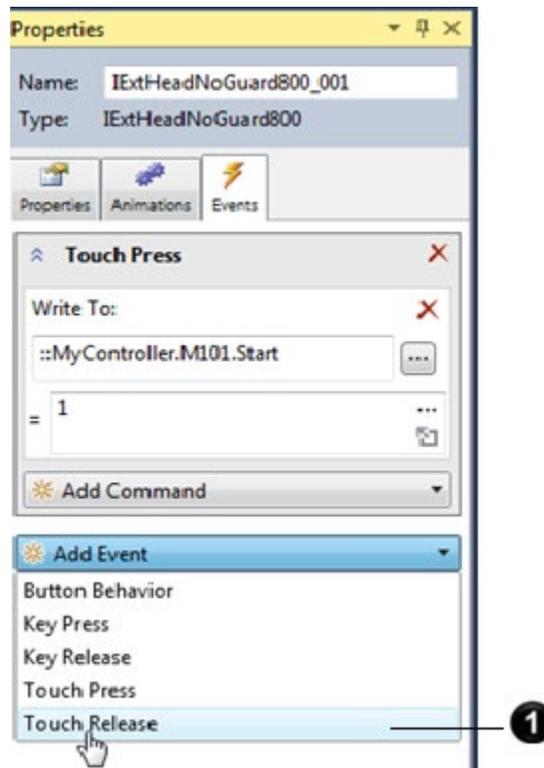
Configure a momentary button with press and release events. Define a **Touch Press** event and configure a **Write** command to set the Motor101.Start tag to 1.

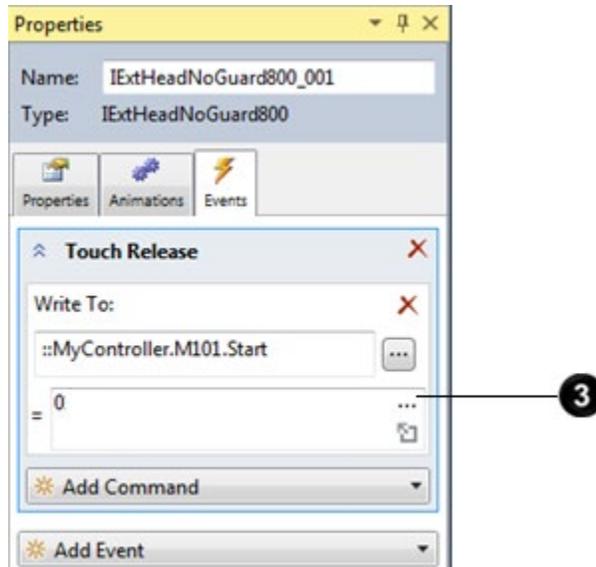




Item	Description
1	Select Touch Press event
2	Add Write command
3	Configure a command for a Touch Press event.

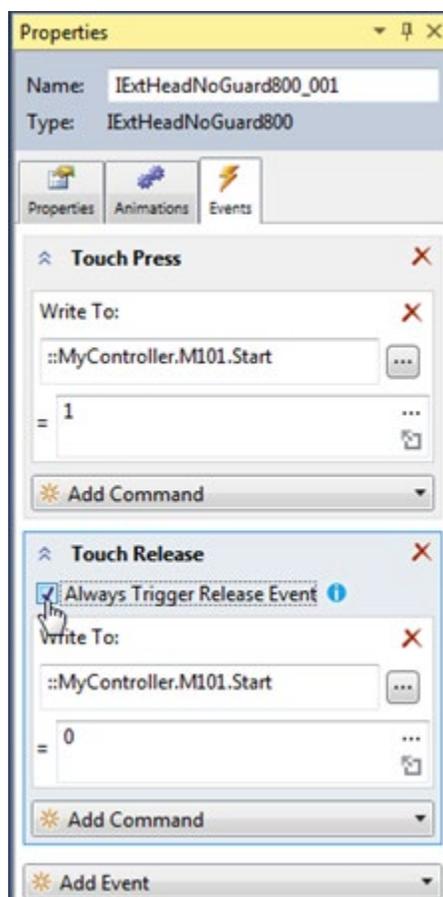
Press **Add Event** for the momentary button. Add a **Touch Release** event with a **Write** command to reset the M101.Start command back to 0.





Item	Description
1	Select Touch Release event
2	Add Write command
3	Configure a command for Touch Release event.

Select the **Always Trigger Release Event** check box to enable the button to work as a momentary button. The **Always Trigger Release Event** configuration ensures the M101.Start bit resets when the operator stops pressing the button, or when the screen navigates away because of a **State Enter** or **State Exit** event.



See also

[Reuse screens](#) on [page 77](#)

[Events](#) on [page 51](#)

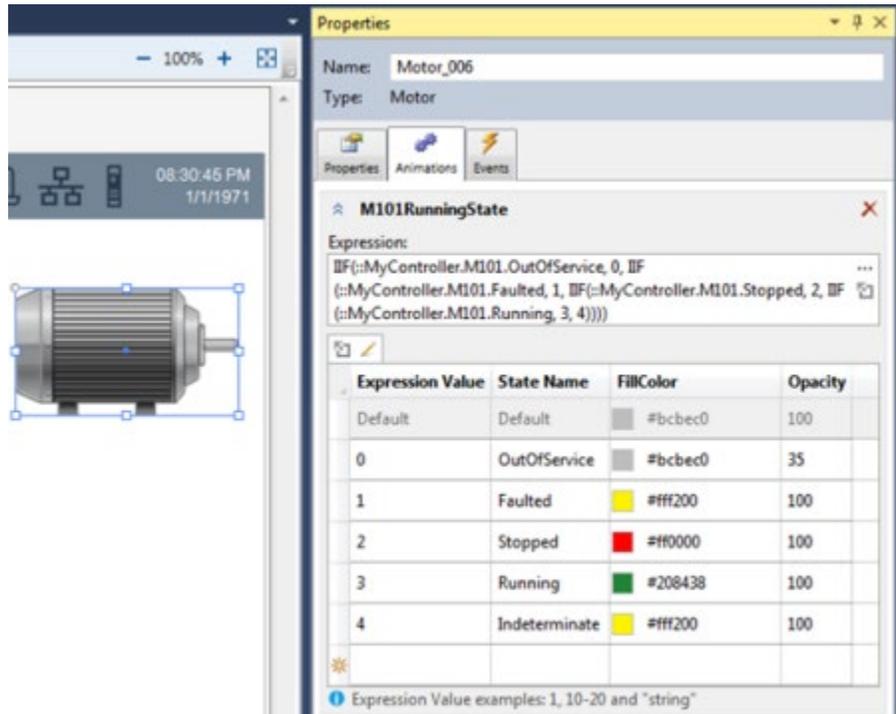
[Example 3: Use State Enter and State Exit Events](#) on [page 59](#)

[Example 4: Use key press and key release events](#) on [page 62](#)

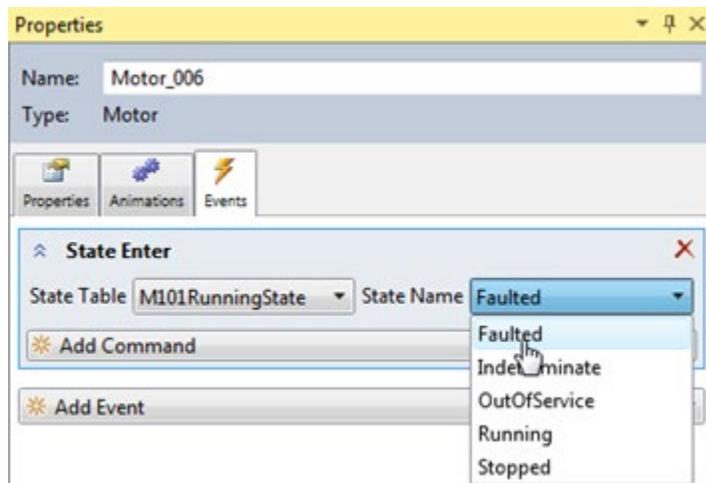
Example 3: Use State Enter and State Exit Events

Configure **State Enter** and **State Exit** events to a state table to trigger commands when changing states. For example, animate a motor by configuring a motor graphic element on the screen with a State Table that has multiple states.

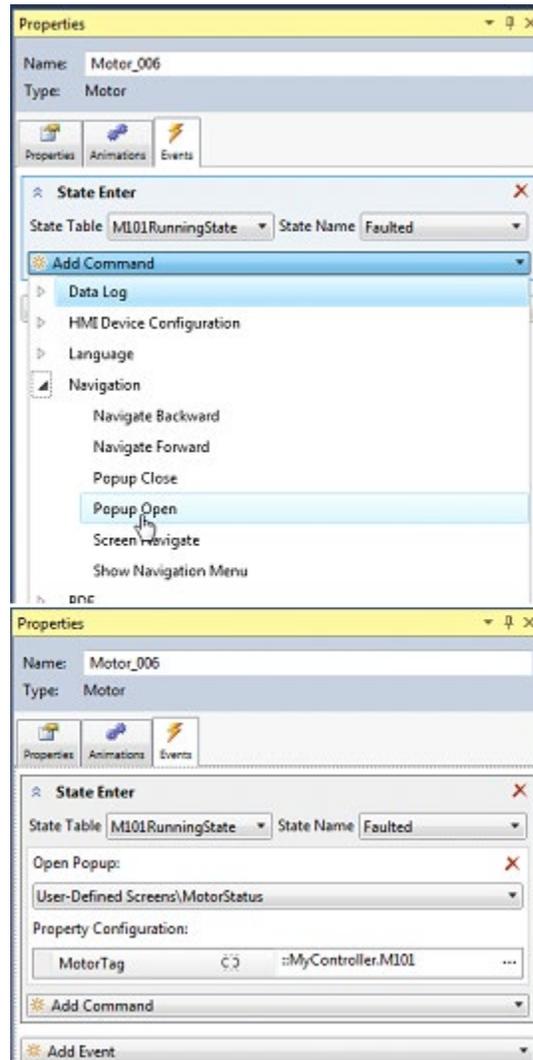
Connect State Enter and State Exit events to state tables for the events to operate. For example, assume that a motor graphic element has a M101RunningState state table where multiple states animate the motor running state.



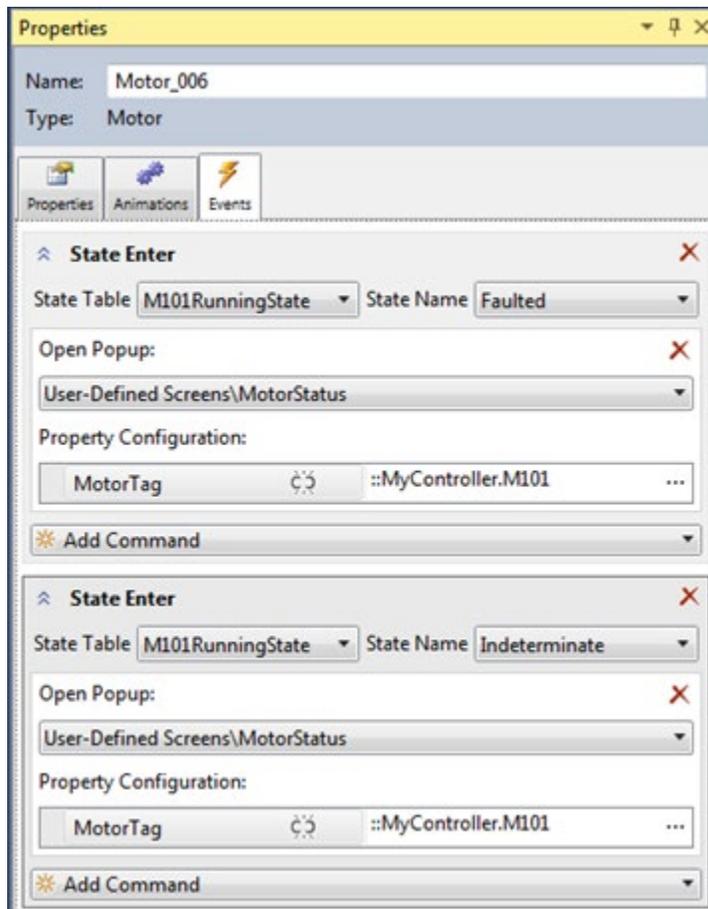
Define state events to trigger a Popup Open command to display a motor status detail popup screen when the motor enters a Fault or Indeterminate state. Configure the State Enter event for the motor entering the Faulted state. In the **Events** tab for the motor, create a **State Enter** event. Select the state table on the motor, and then select the **Faulted** state.



Add the command to open the **MotorStatus** popup. The **MotorStatus** popup is reusable, so give it a tag instance for the motor. The M101 tag is a user-defined structure (UDT) instance in the controller. For more information about creating reusable screens, see *Reuse Screens*.



Add the second event to also display the MotorStatus popup if the motor enters the Indeterminate state.



When displaying the screen with the motor graphic element, and the motor enters the Faulted or Indeterminate states, the MotorStatus popup overlays on the screen.

See also

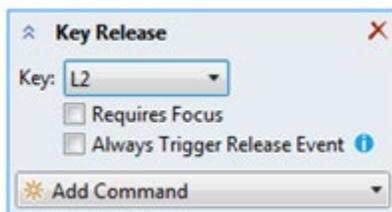
[Reuse screens](#) on [page 77](#)

[Events](#) on [page 51](#)

[Example 4: Use key press and key release events](#) on [page 62](#)

Example 4: Use key press and key release events

For Key Press and Key Release events, there is an additional configuration to select the key and determine if the event needs the graphic element to have focus.



Select **Requires Focus** to use the same key press to perform an action on multiple graphic elements on the screen according to which element has focus. For example, there are five motors on the screen, and the L1 key must always execute a command to set a Start tag true for each motor. Select **Requires Focus** to enable setting only the Start tag for the selected motor.

In a different example, L1 must start Motor1, and L2 must start Motor2, and so forth. In this case, clear the **Requires Focus** checkbox to enable the L keys to always execute their commands to write to the Start tag for the motors. The L keys execute the commands regardless of what graphic element on the screen is in focus.

See also

[Events](#) on [page 51](#)

Project Events

Use **Project Events** to trigger event commands regardless of the open screen. Define **Project Events** from the **Project Explorer** to trigger one or more event commands when an expression transitions from false to true or when a key is pressed on the keypad of a PanelView 5000 HMI device.

See also

[Create a project event](#) on [page 65](#)

[Example: Write the current user name or screen name to a controller using a project event](#) on [page 68](#)

Create a project event

Create a project event to trigger an action for an entire project. Events created for a specific item, such as a graphic element, Add-On Graphic, or a screen, trigger an action for that item only. A project event triggers regardless of the screen or popup open on the HMI device. Create up to 1000 project events for a project.

To create a project event

1. On the **Properties** tab, configure the properties for the project event:
 - **Enabled.** Select or bind this property to enable the evaluation of the event. If the **Enabled** property is false, the event does not trigger and commands do not execute.
 - **ExecuteWhen.** Bind this property to enter a tag or expression to trigger the event. The event triggers when the expression evaluates from false to true.
 - **ExecuteOnKey.** Select the key to trigger the event. Select **None** if not using a key to trigger an event. Any **ExecuteOnKey** project event configured for a specific key does not trigger on the HMI device if the open screen or a graphic element on that screen on the HMI device has a Key Press or Key Release event configured for the same key.
 - **Evaluation Period.** Select a period to define how often View Designer evaluates the **ExecuteWhen** property for the event.
2. On the Events tab, add command cards the event executes. Expand a command category to list the commands in that category:

- **Data Log.** Command that exports a data log to a USB or SD card or cancels the export.
- **HMI Device Configuration.** Commands that configure screen calibration, load a runtime application from removable media to the HMI device, reboot the HMI device, and safely remove media.
- **Language.** Command that changes the language that displays on the HMI device.



Tips:

- Import the desired languages before changing the language that displays on the HMI device.
- If a translatable string is left blank, View Designer defaults to the download language for runtime applications downloaded to the HMI device. If the translated string for the download language is also blank, View Designer defaults to the language used for designing the project.

- **Navigation.** Commands that navigate to screens on the HMI device.



Tip: Multiple screen navigation commands can exist for an event. All other types of commands for the event are executed before the navigation command. Only one navigation command is executed for an event. This is generally the first navigation command. The remaining navigation commands are ignored on the HMI device. To ensure the navigation command is the navigation command executed, Rockwell Automation recommends adding only one navigation command to an event.

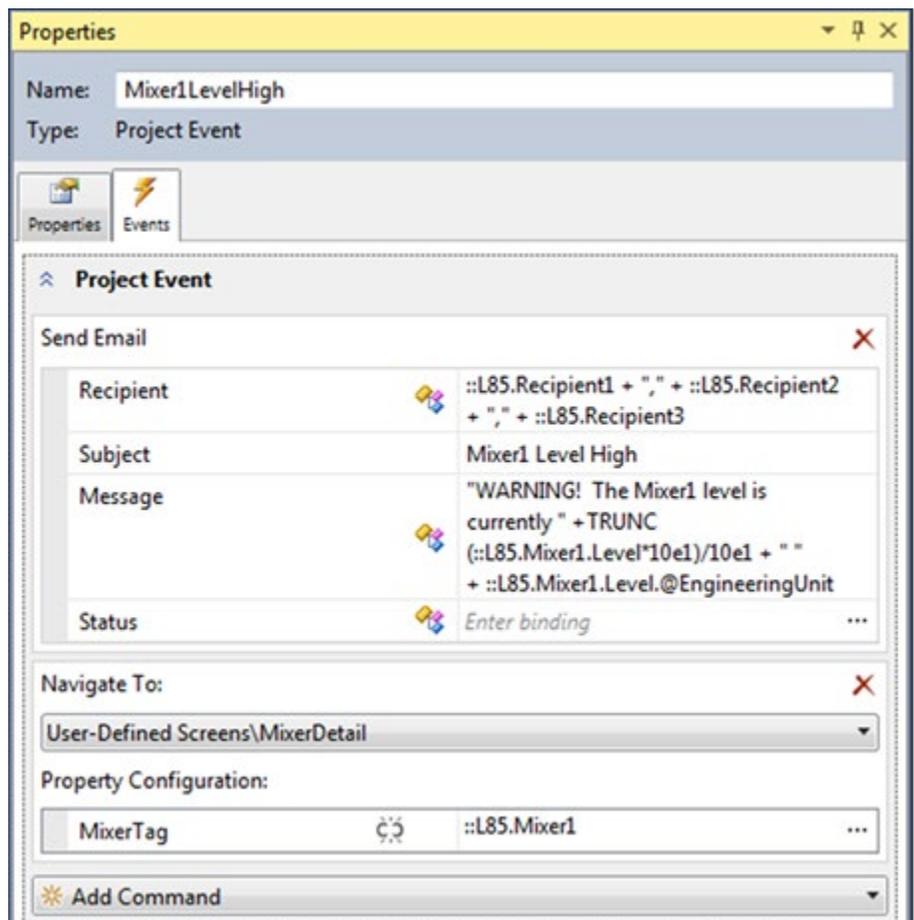
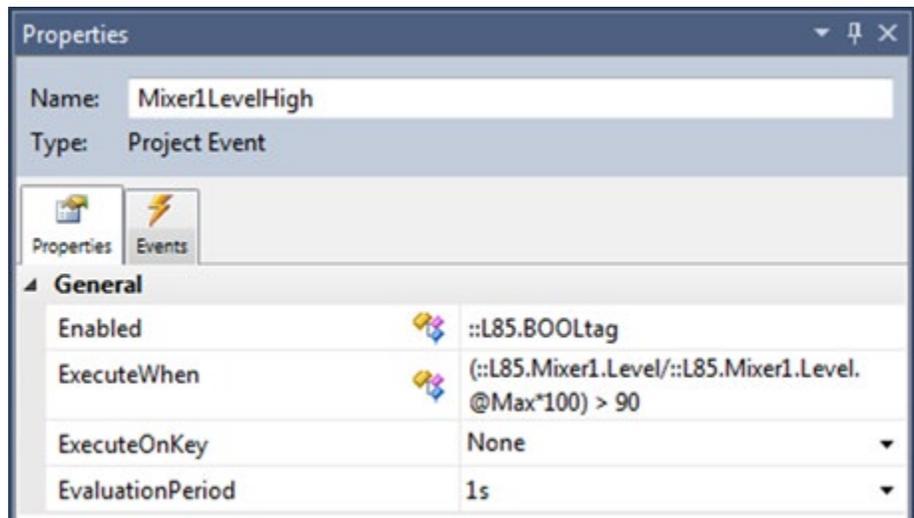
- **Notification.** Commands that send an email notification.
- **Security.** Commands that log on or log off the HMI device.
- **Value.** Commands that increment, toggle, or write a value to the selected tag.



Tips:

- Create multiple commands for each event. The order of execution may not correspond with the order of commands listed on the Event card.
- Creating many tag write commands for an event may pause screen updates while the writes occur.
- When an event performs multiple commands that write to multiple tags, all of the write commands occur sequentially. Therefore, you may not see all of the tag values in the Logix controllers at the same time.
- For best performance, configure commands to evaluate no more than 20 tags. For example, configure a **Write** command that has an expression that contains no more than 20 tags. Have a **Send Email** command insert no more than 20 tags in a message.
- Creating over 100 tag **Write** commands for a project event may pause screen updates while the **Write** commands occur.
- Configure 10 or fewer commands for an event that references many tags. Using larger numbers of tag references in many commands may cause slow screen performance while commands evaluate.
- Reference 1000 or fewer tags across all project events in a project. Using larger numbers of tag references may cause delays when a PanelView 5000 recovers from controller disconnects.
- Executing a large number of Project Events quickly can overload the project queue. If this happens, the HMI device may skip project evaluation periods.
- If you define multiple navigation commands in a Project Event, the HMI device executes only one navigation command for the event. For consistent results, use one navigation command for a Project Event.
- The HMI device may execute commands in a different than order than on the Events tab.

This example project event sends an email notification and navigates to a screen when a mixer level exceeds 90%.



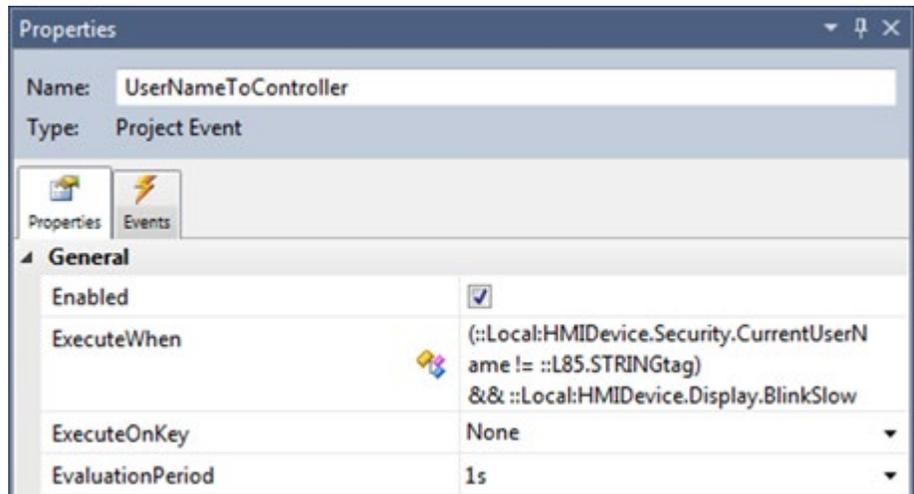
See also

[Project Events](#) on [page 65](#)

[Example: Write the current user name or screen name to a controller using a project event](#) on page 68

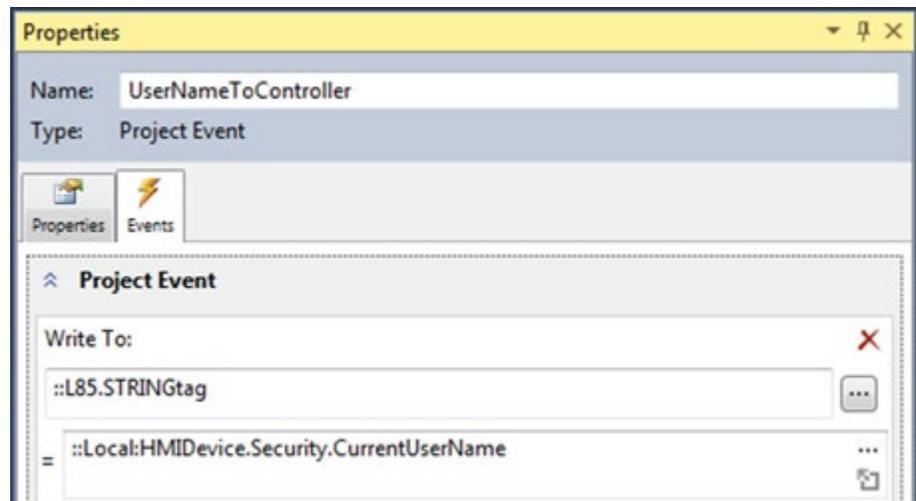
Example: Write the current user name or screen name to a controller using a project event

Use **Project Events** to write PanelView 5000 information to the controller. For example, if the controller needs to know the current user, create a project event named *UserNameToController* and configure the **ExecuteWhen** property of the project event:

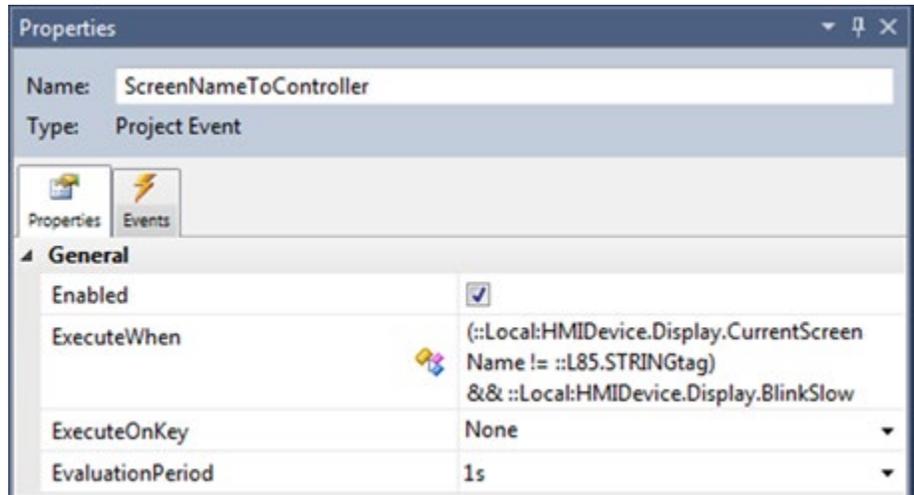


When the **CurrentUserName** local HMI tag is not equal to the value of a string tag in the controller, **ExecuteWhen** becomes true. The AND operator with the **BlinkSlow** local HMI tag forces a false to true transition for **ExecuteWhen** when starting the PanelView 5000.

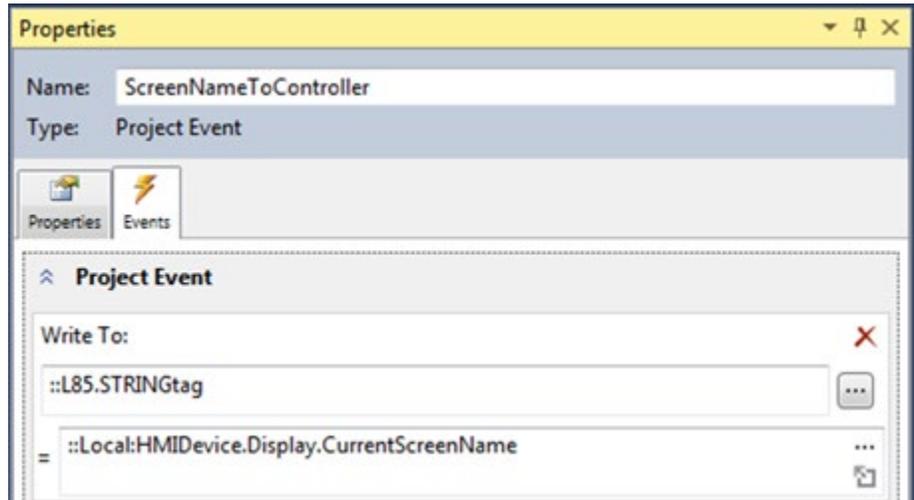
On the **Events** tab, configure a **Write To** command. Write the value of the **CurrentUserName** to the string tag when the *UserNameToController* project event triggers.



To send the current screen name to the controller, use the **CurrentScreenName** local HMI tag in a project event:



Have the project event write the current screen name to the controller:



See also

[Project Events](#) on [page 65](#)

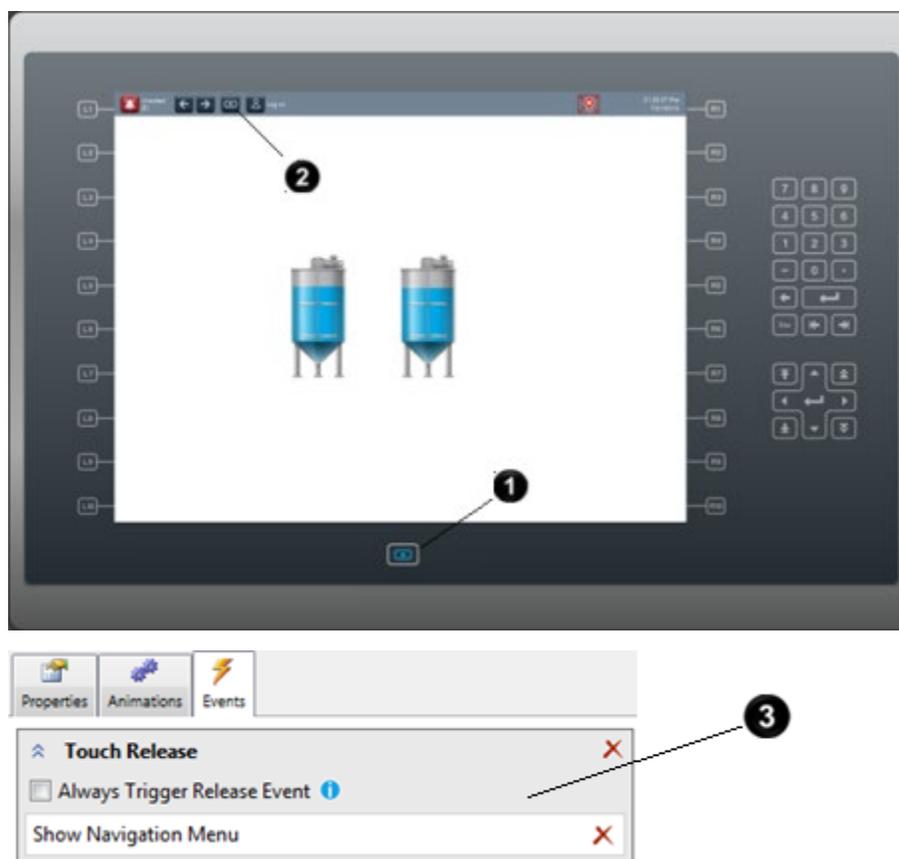
Navigation menu

Use the navigation menu on the PanelView 5000 HMI device to set up navigation for operators. The navigation menu:

- Enables configuring navigation in the **Navigation** folder of **Project Explorer** to eliminate configuring navigation buttons on every screen.
- Saves screen space by sliding up when pressing the Navigation button and sliding down when selecting a screen.
- Automatically hides navigation shortcuts to screens that are not applicable for the security access level of an operator.

To display the Navigation menu at runtime:

- Press the physical button at the bottom of a PanelView 5510 HMI device.
- Touch the navigation button on the system banner or configure a **Show Navigation Menu** command to use with an event on a graphic element.



Item	Description
------	-------------

Item	Description
1	Physical button on a PanelView 5510 terminal
2	Soft navigation menu on the system banner
3	Show Navigation Menu command used with an event on a graphic element

See also

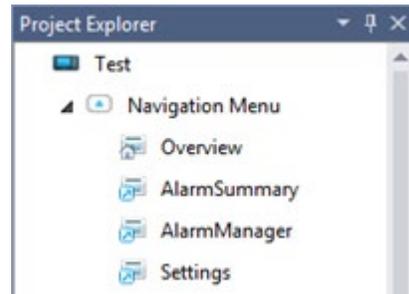
[Define shortcuts to setup Navigation Menu](#) on [page 72](#)

[Example: Configure Navigation Menu properties for runtime](#) on [page 73](#)

[Events](#) on [page 51](#)

Define shortcuts to setup Navigation Menu

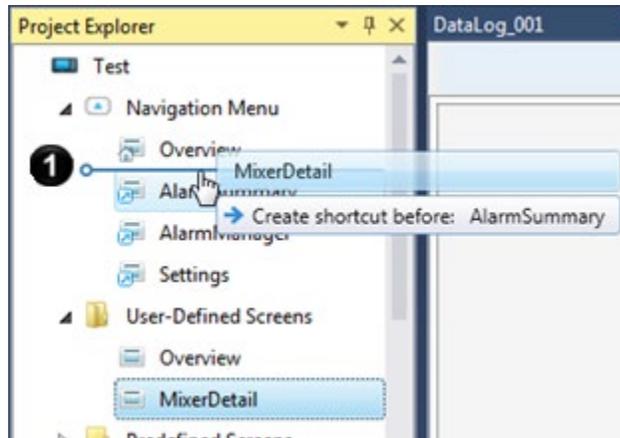
To set up the navigation menu, define shortcuts in the **Navigation Menu** folder in the **Project Explorer**.



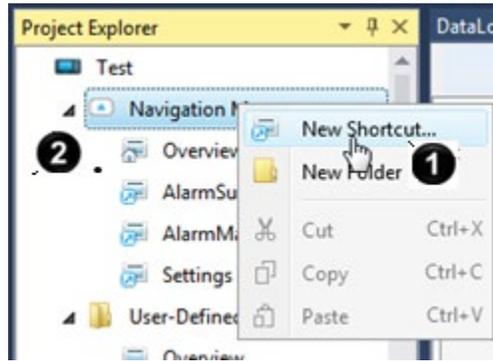
View Designer refers to items in the **Navigation Menu** folder as Shortcuts. Shortcuts are navigation links to the user-defined or pre-defined screens in a project.

To create shortcuts, use one of these methods:

- Drag a screen into the **Navigation Menu** folder.
- Right-click on the **Navigation Menu** folder and **Select New Shortcut**.



Item	Description
1	For example, drag the MixerDetail screen into Navigation Menu folder



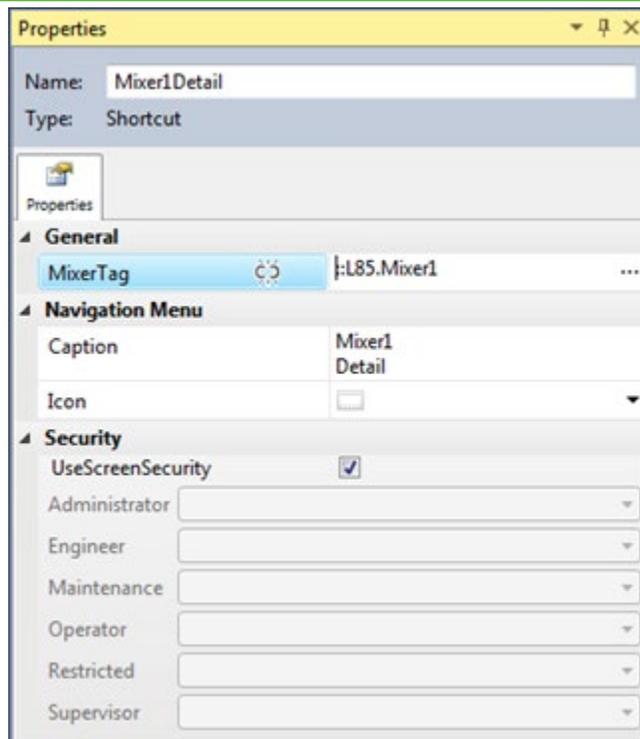
Item	Description
1	New Shortcut
2	Navigation Menu folder

See also

[Example: Configure Navigation Menu properties for runtime](#) on [page 73](#)

Example: Configure Navigation Menu properties for runtime

After creating a shortcut, change shortcut properties in the **Properties** pane as necessary.



In this example, the **MixerDetail** screen is a reusable screen that has a custom **MixerTag** property. The **MixerTag** property needs a tag context to display the desired mixer information. Custom screen properties appear on the shortcut **Properties** pane to provide that tag information. Refer to *Reusable Screens* chapter for more information.

The **Caption** property:

- Displays on the navigation menu icon for the shortcut
- Text defaults to the shortcut name
- Enables entering spaces
- Uses the Enter key to make a two-line caption
- Enables entering Unicode characters as necessary
- Supports language switching to enable operators to see the **Navigation Menu** items in the applicable language.

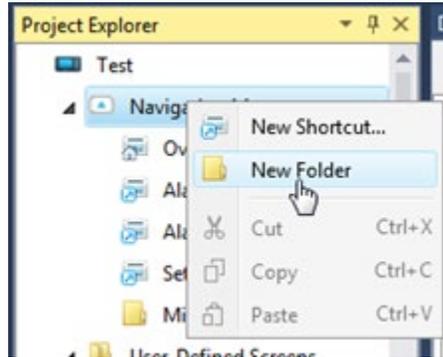
Use the **Icon** property to define the icon to use for the shortcut on the **Navigation Menu**. To select an icon, click the list and select a predefined icon or select a user-defined icon.

Use the **Security** property to define different security access for the shortcut. Typically, select the **UseScreenSecurity** checkbox to enable the shortcut to use the same security configuration as the screen it references.

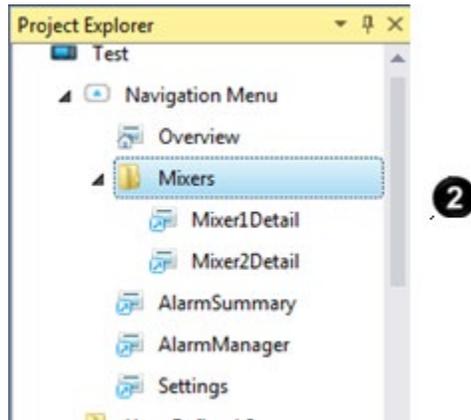
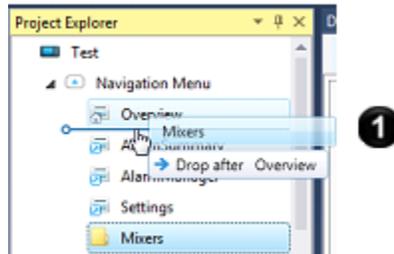


Tip: Specify security access for a user role when a reusable screen needs user roles to access different instances of a shortcut.

The **Navigation Menu** supports defining up to three levels of folders. Each level appears above the lower level at runtime. For example, create a Mixer folder with shortcuts to different mixer instances by right-clicking on the **Navigation Menu** icon and selecting **New Folder**.

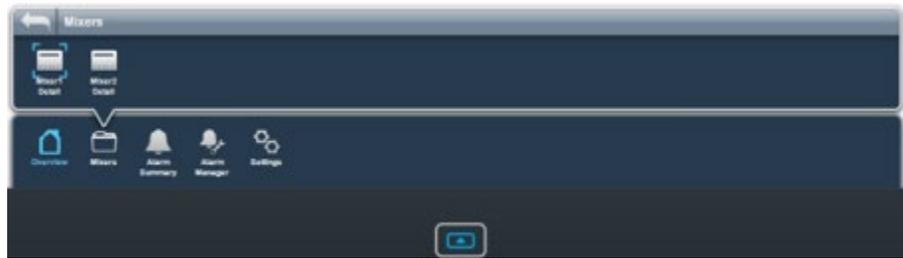


View Designer supports renaming the folder and dragging it to a desired place in the **Navigation Menu**. You can also drag other shortcuts into the folder or create new shortcuts in the folder.



Item	Description
1	Drag the folder to a desired location in the Navigation Menu .
2	Drag shortcuts into the folder or create new shortcuts in the folder.

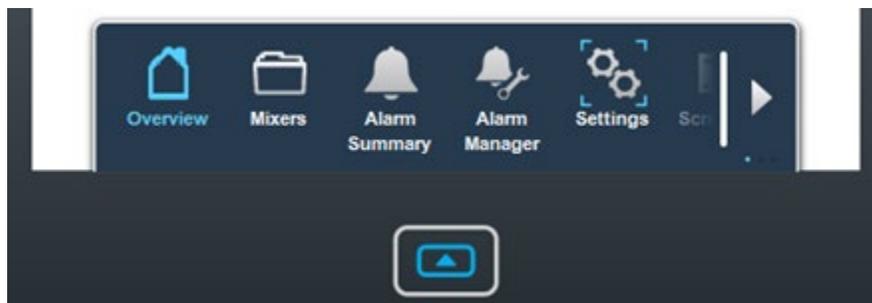
At runtime, the Navigation Menu then appears as illustrated:



Complex navigation menu configurations support:

- Referencing a reusable screen multiple times with different tag instances.
- Referencing the same screen multiple times in the Navigation Menu for quick access to the screen from multiple menu levels.

The **Navigation Menu** also supports many shortcuts. If not all the shortcuts fit on the screen, use the left and right arrows that appear on the Navigation Menu to access all the shortcuts. This generally occurs on smaller PanelView 5000 HMI device screens.



See also

[Navigation menu](#) on [page 71](#)

[Define shortcuts to setup Navigation Menu](#) on [page 72](#)

[Reuse screens](#) on [page 77](#)

Reuse screens

Reusable screens support creating a screen once and using it repeatedly with different contexts.

Create custom properties for the screen and then add different values to those custom properties when navigating to the screen. Custom properties link to properties of the graphic elements on the screen or the properties of the screen.

Create reusable screens or popups using one of these methods:

- When changing a small number of properties, create a custom property using an alias of a property of a screen element.
- If changing multiple properties, create a custom property bound to a Logix data type. Do this when using the same screen repeatedly to show different pieces of identical equipment. The data of the equipment typically appears as a user-defined data type (UDT) or an Add-On Instruction data type in Logix Designer.

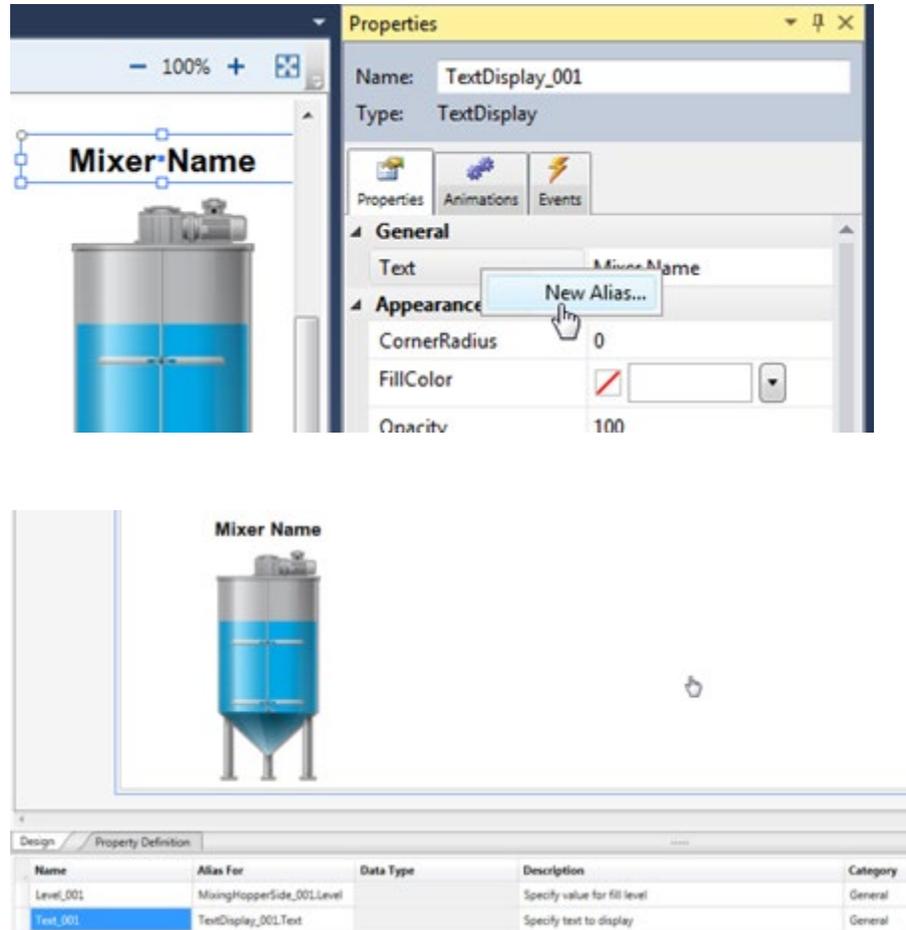
See also

[Example 1: Use alias properties](#) on [page 77](#)

[Example 2: Use data type properties to create a reusable screen](#) on [page 80](#)

Example 1: Use alias properties

In this example, create a reusable screen with the name *MixerDetail* to show the level in a mixer and the name of the mixer. Use the screen for multiple mixers. Create an alias for the **Level** property of mixer and the **Text** property of a text display. The **Alias** properties appear in the **Property Definition** pane under the screen.



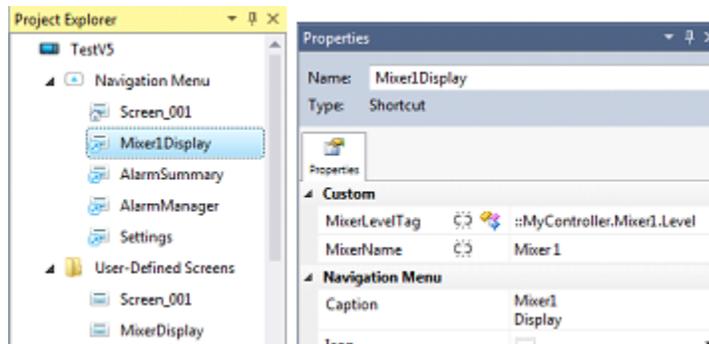
Use the **Property Definition** pane to:

- Create and view custom properties for the screen.
- View which element and property link to the Alias.
- Rename custom properties
- Change descriptions

- Change the category for custom properties to appear in a different category in the **Properties** pane.



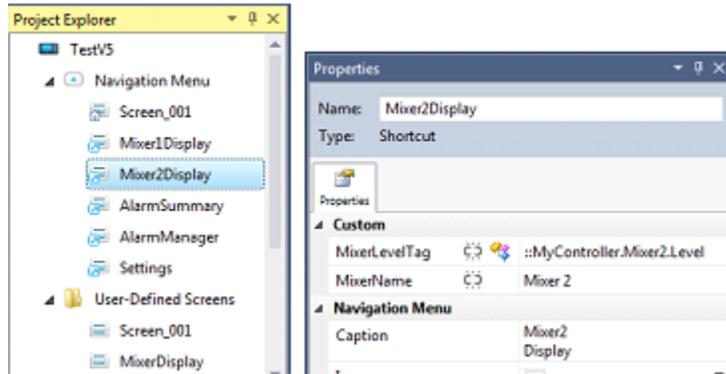
Set up a navigation to the **MixerDisplay** screen by dragging the **MixerDisplay** screen into the **Navigation Menu** folder. This creates a shortcut to the **MixerDisplay** screen on the runtime **Navigation Menu**. Rename the shortcut *Mixer1Display*. The properties of the shortcut display the configurable custom properties for the screen.



In this example, bind the **MixerLevelTag** property to a tag in the controller having the value of the level of the mixer. Then enter *Mixer1* for the value of the **MixerName** property. At runtime when the operator selects the **Mixer1** display shortcut, the **MixerDisplay** reusable screen appears.



Repeat the steps for multiple mixers. For *Mixer 2*, configure another shortcut as:



The **Navigation Menu** has two shortcuts to the same **MixerDisplay** screen, but has different data displays for each shortcut.



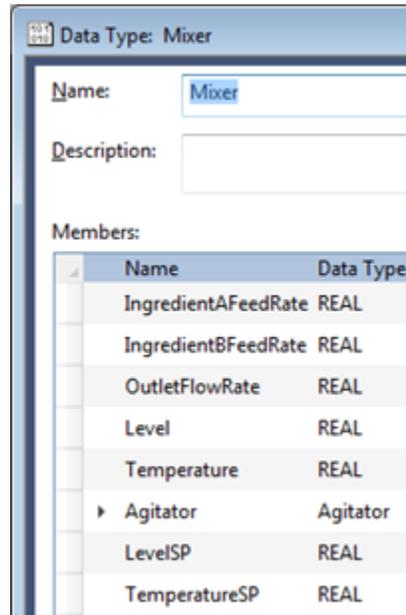
See also

[Example 2: Use data type properties to create a reusable screen](#) on [page 80](#)

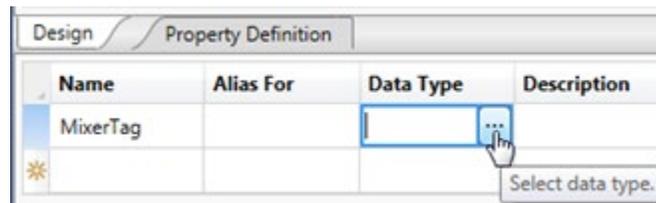
[Reuse screens](#) on [page 77](#)

Example 2: Use data type properties to create a reusable screen

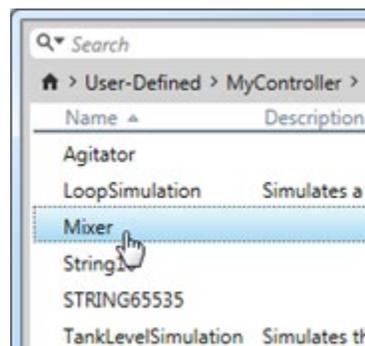
If Logix 5000™ represents equipment data as a data type, use instances of that data type to animate graphic elements on a screen. Use a custom screen property tied to the Logix data type. Create a *MixerDetail* screen that shows the status of a mixer represented by a user-defined *Mixer* data type in Logix 5000:



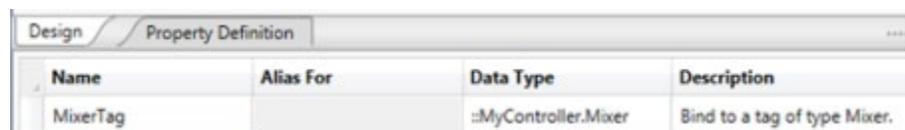
Create a custom property named MixerTag for the **MixerDetail** screen.



Browse the **Data Type** field to display all the data types in the Logix 5000 project and select the user defined Mixer data type.

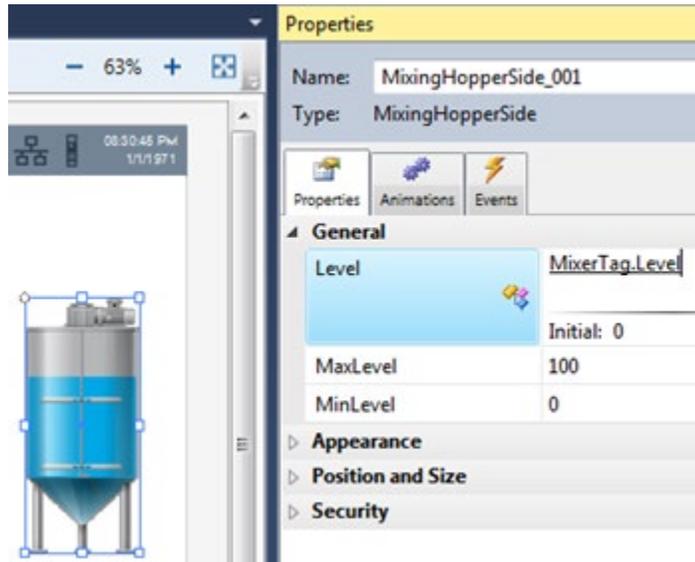


Enter an optional description for the custom MixerTag property.

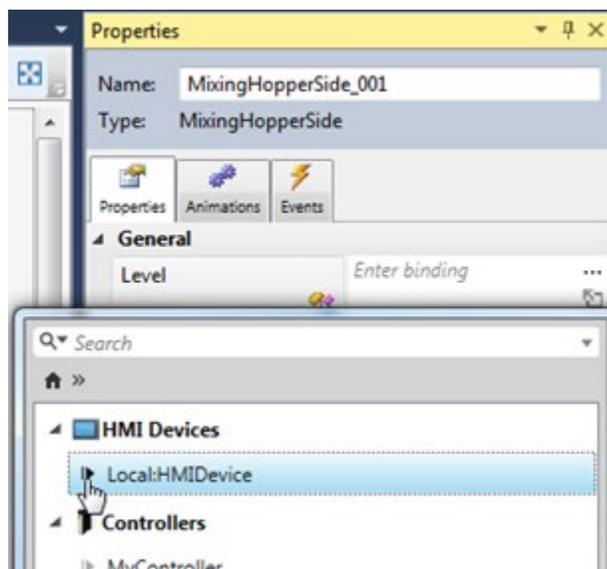


The custom MixerTag property appears as a property of type Mixer. View Designer supports using the definition of the data type to create animation

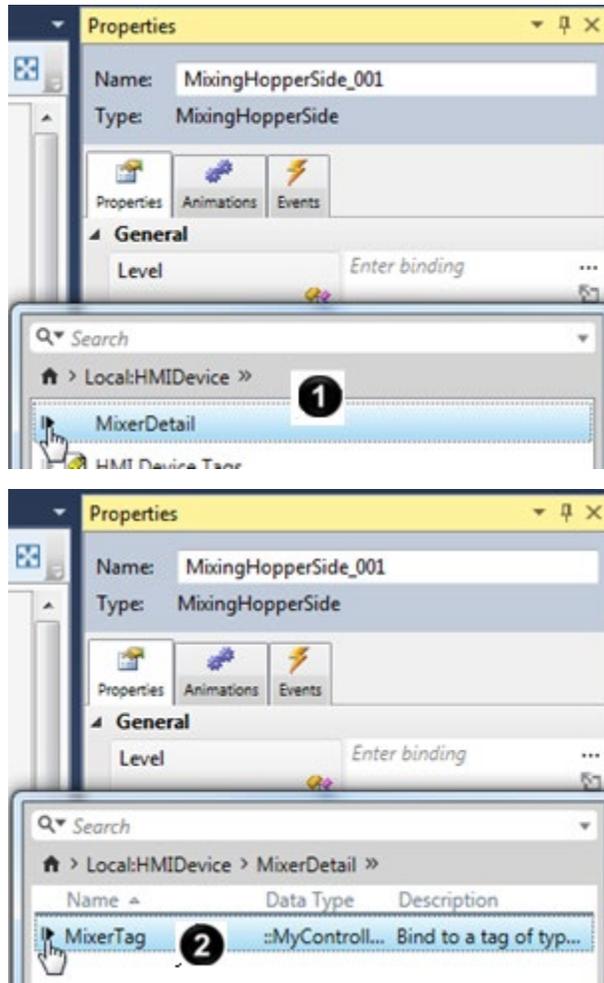
bindings for graphic elements on the **MixerDetail** screen. For example, add a Mixer graphic element to the screen and bind the **Level** property to the **Level** member of the **Mixer** data type. When defining the bindings, use the **Tag Browser** to browse the **MixerTag** property.



Browse the **MixerTag** property by navigating the **MixerDetail** screen property:

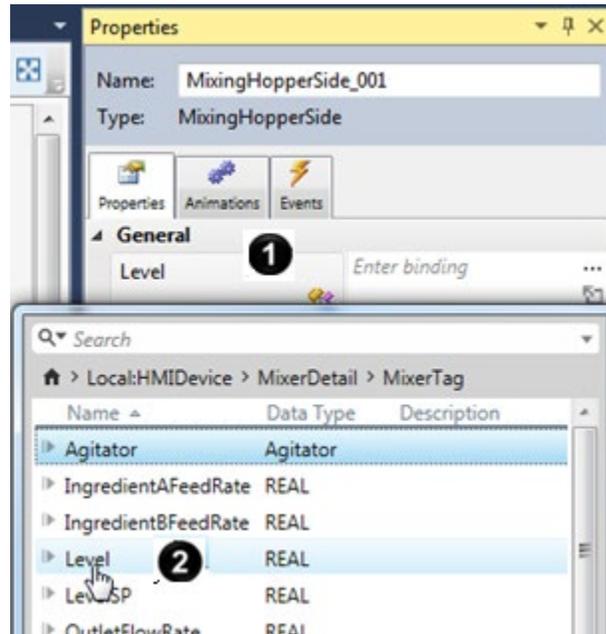


In this example, select the drill-in arrow next to **MixerDetail** to access the **MixerDetail** screen to view the custom MixerTag property.



Item	Description
1	MixerDetail
2	MixerTag property

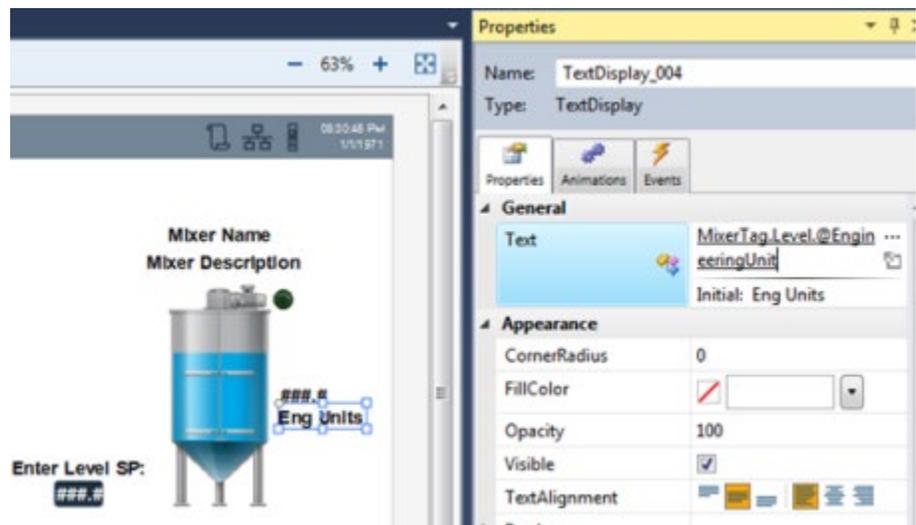
This example defines the **MixerTag** property as a data type Mixer. Select the drill-in arrow next to the **MixerTag** property to view all the members of the Mixer data type definition. Then select the **Level** member of the **MixerTag** to bind to the **Level** property of the Mixing tank.



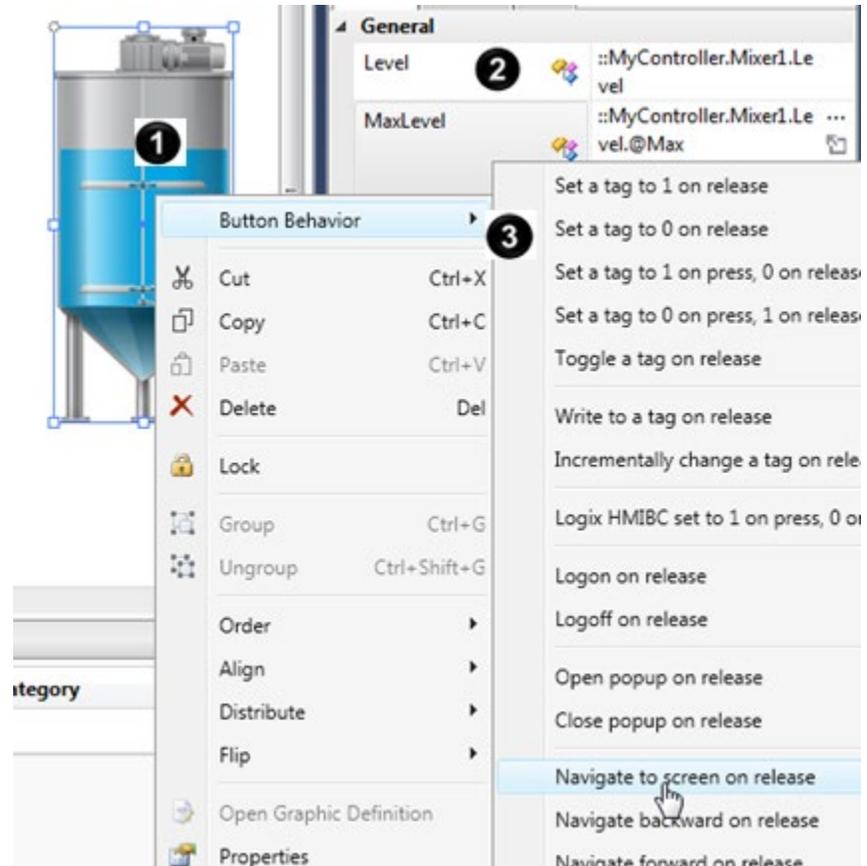
Item	Description
1	The Level property of the mixing tank
2	The Level member of the MixerTag property

Browse the data type definition to configure all the bindings on the **MixerDetail** screen to reduce time and ensure accuracy instead of manually typing data type member names.

Continue configuring the property bindings for the graphic elements by browsing the **MixerTag**. Bind any members of the Mixer data type including extended tag properties.

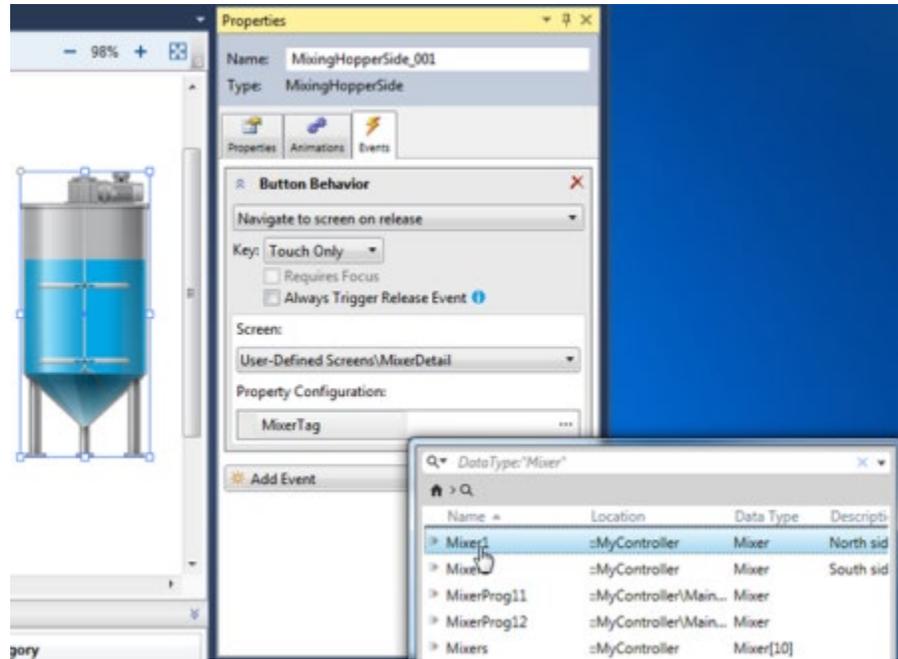


After creating and configuring the **MixerDetail** screen, access the screen using a **Navigation Menu** shortcut or a Navigation command. In this example, use a Mixer element on an overview screen to display the **MixerDetail** screen with the correct context. Add a Mixer element on the overview screen and bind it to the desired Mixer tags. Then create a **Button Behavior** to configure the element to act as a button.



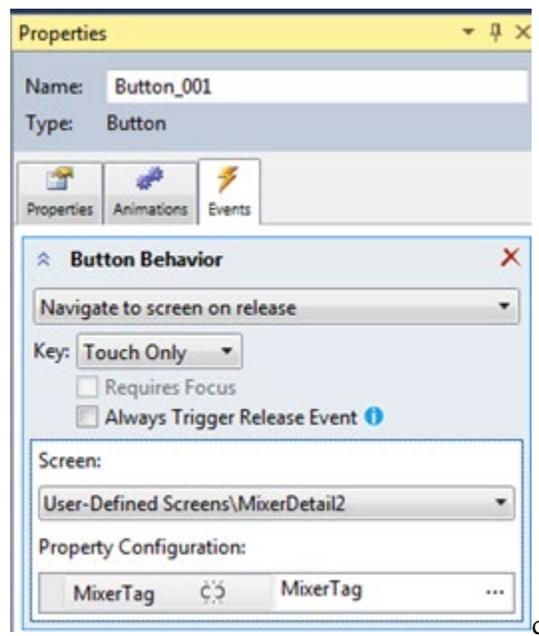
Item	Description
1	Mixer element on overview screen
2	Bind properties
3	Configure a button behavior

When selecting **Navigate to screen on release**, the **Events** tab in the **Properties** pane opens. Select the **MixerDetail** screen. Open the tag browser to select a Mixer tag instance for the MixerTag property. The tag browser filters and displays the controller tags of type Mixer. Select a **Mixer** tag for this instance of the mixer.



To use the **MixerDetail** screen again, add another navigation command or shortcut to the **MixerDetail** screen and select a different Mixer tag instance.

View Designer supports passing custom property definitions from screen to screen. For example, to have the **MixerDetail** screen communicate with another **MixerDetail2** screen to display more information about the mixer, configure a button on the **MixerDetail** screen as shown.



This takes the MixerTag instance used by MixerDetail and passes it to the **MixerTag** property for **MixerDetailz**.

See also

[Reuse screens](#) on [page 77](#)

Active Directory Authentication

Configure network-based security logon using Active Directory

Network-based security enables managing user accounts and user groups centrally. It allows network users to log on to an HMI device, authenticate with the Active Directory, and be granted authorization rights based on their role.

When network-based logon is used for a project, local users still may be configured and used on the terminal (for example for back-up). You can switch to local users on the logon pop-up in runtime.



Tip: Leaving the **Connection** fields blank will have the panel use local users at runtime.

Active Directory properties may vary depending on your company and the complexity of system architecture. If you find any problems during configuration, contact your Active Directory administrator.

Prerequisites:

Ensure that the logon method is set to use the user name and password:

1. From the main menu, select **Project > Project Properties**.
2. In the **Project Properties** window, select the **Log On Method** tab.
3. Select the **Use user name/password for logging on** radio button.

To configure network-based security logon:

1. From the main menu, select **Tools -> Security Administration**.
2. In the Security Administration window, select the **Connection and Groups** tab.
3. In the **Connection** section, fill in the required fields.
 - a. In the **Domain Name** field, enter your corporate domain.

IMPORTANT During the Active Directory logon process, **user name** and configured **Domain name** are combined to define the **User Principal Name**, in the format <username>@<domain>.com. It is sent with a password to the domain controller for authentication.

The username used for the **User Principal Name** must be used during Active Directory logon.

EXAMPLE Correct format: jamesmith@domain.com,
Incorrect format: domain.com\jsmith

- b. In the **Domain Controller** field, enter the full address of the Active Directory server.
- c. In the **Port** field, enter the number of the port.



Tip: Typical default ports are: no encryption - 389, StartTLS - 389, LDAPS - 636.

- d. In the **Domain Distinguished Name** field, enter your domain distinguished name.

EXAMPLE DC=domain,DC=com

- e. If the connection requires encryption, select the encryption method.

IMPORTANT Not using encryption for network-based authentication results in sending username and password as plaintext over the network. This creates a security risk.

To use StartTLS or LDAPS encryption for logon, you are required to import the proper security certificate to the terminal by using the Certificates setting screen.

- 4. In the Groups and Roles section, click **+** next to Add new User Group.



Tip: You can add and define multiple groups.

- 5. In the new section that appears, enter the following information:
 - a. In the **User Group Distinguished Name** field, enter a distinguished name of a group.

EXAMPLE CN=Operators,OU=Groups,DC=domain,DC=com

- b. In the dropdown menu next to the **User Group Distinguished Name** field, select the role you want to assign to the group.



Tip: You can remove a group by clicking **X**.

At runtime, when someone logs in, the Active Directory returns the group membership of that user. Now this group is mapped to a role, so the user gets the security role mapped to the Active Directory group they belong to. If a user belongs to multiple groups with different roles, they will get the first role configured in this list. If a user does not belong to any group mappings, the user will receive an error message on the PanelView 5000 when they log on.

- 6. Click **OK**.



Tip: Active Directory properties may vary, depending on your company and the complexity of the system architecture. If you encounter any problems during configuration, contact your Active Directory administrator to obtain guidance or this configuration.

Find the Active Directory Logon Information

See also

Security roles

Use the Windows Active Directory Users and Computers tool to find the connection information. You can find this tool in the Windows Start menu under Windows Administrative Tools.



Tip: To see all the attributes of selections, you may need to turn on viewing of **Advanced Features** under the **View** menu of the **Active Directory Users and Computers** tool.

To find the Active Directory logon information:

1. From the Windows start menu, select **Windows Administrative Tools**.
2. Select the **Active Directory Users and Computers** tool.
3. In the **Active Directory Users and Computers** tool, select the **People** folder, and on the list find a person who will log on the PanelView 5000.
4. Double-click the user name to open the **Properties** for that user.
5. Navigate to the **Account** tab, and note the following information:
 1. User logon name. This is the User Name that the user will enter on the Logon dialog on the PanelView 5000.
 2. The domain after the “@” is what you will enter as the **Domain Name** on the **Connection and Groups** dialog.
6. In the **Active Directory Users and Computers** tool, select the **Domain Controller** folder, and find the domain controller you use.
7. Double-click the domain controller, and note the DNS name.

That name is what you will enter for **Domain Controller** in View Designer.

8. In the **Active Directory Users and Computers** tool, right-click on the **Domain** folder, and select **Properties**.



Tip: The **Domain** folder is the high-level folder named the same as your domain in the left pane of the Active Directory Users and Computers tool.

9. Select the **Attribute Editor** tab, and note the value for the `distinguishedName` attribute.

This is the value that you enter in the **Domain Distinguished Name** field in View Designer.

10. In the **Active Directory Users and Computers** tool, display the properties of the user, and select the **Member Of** tab to see their assigned groups.

11. Double-click the group that the user belongs to, in the **Attribute Editor** tab find the `distinguishedName` attribute and note the value.

This is the value that you enter in the **User Group Distinguished Name** field in View Designer.

Log on using network-based security

To log on using network-based security:

1. To log on to the application using your network-based security configuration, on the top of the screen, select the log on button .

IMPORTANT The user name used for the **User Principal Name** must be used during Active Directory logon.

2. Ensure that you are logging on as a domain user, enter your Active Directory user name and password, and Select **Log on**.

IMPORTANT When StartTLS or LDAPS encryption method is configured, you are required to import the proper security certificate to the terminal by using the Certificates setting screen.

A caching mechanism is used for users who successfully logged on with Active Directory authentication. If the PanelView 5000 loses connection to the Active Directory, the user can still log on with cached domain credentials for 24 hours. After this period, logon is no longer possible without an Active Directory connection. The Cache is only cleared on project download, and not on reboot or restart.



Tip: You can switch between a local user and a domain user by selecting the link at the bottom of the **Logon** screen.

If the **Connection** information is left blank in View Designer, only local user logon will be used by the logon dialog at runtime.

See also

Import a digital certificate from removable media

Adding a Certificate for Encrypted

To use StartTLS or LDAPS for encrypted connections, you must load a certificate onto the PanelView 5000.

Find the certificate used by your company to securely log into the Active Directory server. Import this certificate to the PanelView 5000 as a .cer file using base-64 encoded X.509.

To copy the certificate from your PCs:

1. From the search field on the Windows start menu, run **Manage User Certificates**.
2. Under **Trusted Root Certification Authorities**, select **Certificates**.
3. In the pane on the right, double-click your company certificate to open the certificate information.
4. In the **Details** tab, select **Copy to File...**
5. Select **Base-64 encoded X.509 (.CER)** file format.
6. Copy the certificate file to an SD card or USB drive.
7. Plug the SD card or USB drive with the certificate into your PanelView 5000.
8. In PanelView 5000, use the predefined **Certificates** screen to import the certificate onto the device.

Add-On Graphics

Use Add-On Graphics to reuse items throughout a project to save development time. Create graphics once and then reuse them as often as needed.

Use Add-On Graphics to create reusable graphics by combining multiple Toolbox elements into a single graphic. Add-On Graphics are in their own Add-On Graphics folder in the **Toolbox**. Reuse Add-On Graphics by adding instances of the graphic to screens throughout the project. Add-On Graphic changes apply to the Add-On Graphic definition in a central location and propagate to all instances of the graphic contained in the project.

See also

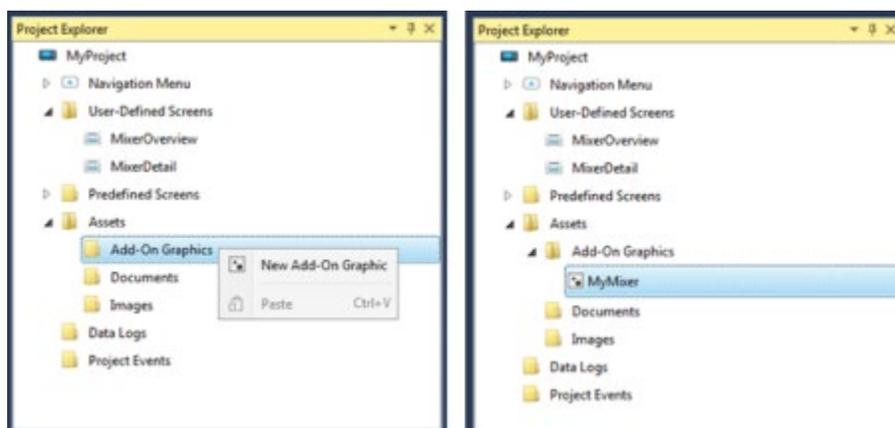
[Create an Add-On Graphic](#) on [page 95](#)

[Example: Use custom properties for an Add-On Graphic](#) on [page 97](#)

[Create an instance for the Add-On Graphic definition](#) on [page 99](#)

Create an Add-On Graphic

As an example, create an Add-On Graphic to represent a mixer on a mixing line. In **Project Explorer**, select **Assets**. Right-click the **Add-On Graphics** folder and select **New Add-On Graphic**. In this example, name the Add-On Graphic **MyMixer**.



To open the definition of the Add-On Graphic, select **MyMixer**. The definition opens and displays an outline of the target HMI device screen size.

Adding graphic elements to Add-On Graphics and configuring the graphic elements is the same process for screens and popups. For example, add a graphic element to the add-on graphic to represent the mixer. In the **Toolbox**,

select the **Mixing Hopper Side** graphic element. The Add-On Graphic adds the graphic element to the center.



Add elements to display the name of the mixer, the level of the mixer, and a numeric input to specify the level set point. When adding elements to the add-on graphic definition, the checkerboard pattern shows the size of the graphic.



See also

[Example: Use custom properties for an Add-On Graphic](#) on [page 97](#)

[Create an instance for the Add-On Graphic definition](#) on [page 99](#)

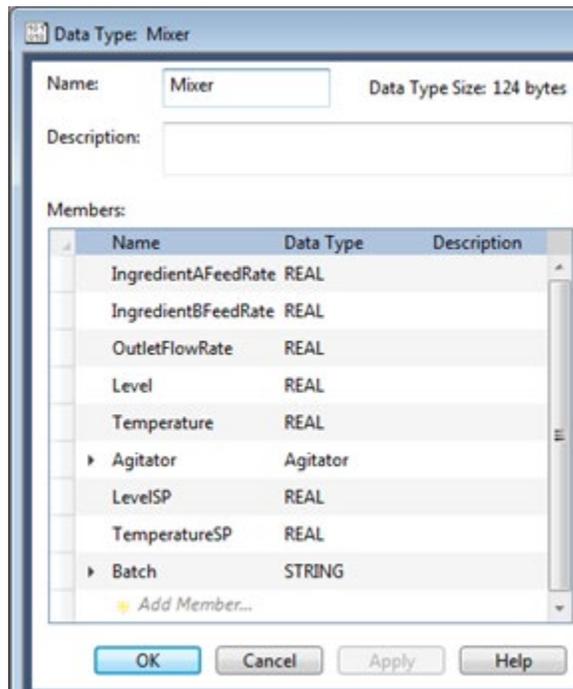
Example: Use custom properties for an Add-On Graphic

Create custom properties for an add-on graphic and pass different values to the custom properties when navigating to a screen having instances of the add-on graphic. The custom properties link to properties of the graphic elements contained within the add-on graphic definition.

These methods create custom properties for an Add-On Graphic.

- When changing a few properties of an Add-On Graphic, create a custom property using an Alias of a property of an element in the Add-On Graphic.
- When changing many properties of an Add-On Graphic, create a custom property tied to a Logix data type. For example, when using an Add-On Graphic repeatedly with different pieces of identical equipment, and the data of the equipment is a user-defined data type (UDT) or an Add-On Instruction data type in Logix Designer.

In this example, tie the **MyMixer** add-on graphic definition and its contents to a user-defined data type using the name Mixer in the Logix controller project. The Mixer data type in Logix Designer has the following members.

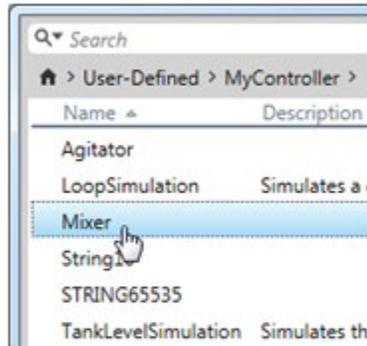


Pass in data from the Mixer data type instances in the controller to animate instances of the **MyMixer** Add-On Graphic used in the HMI. Create a custom property in the Add-On Graphic definition tied to the Mixer data type in the controller project.

Create the custom property in the **Property Definition** pane below the Add-On Graphic editor. In the **Name** box enter **MixerTag**.

Next, click the ellipsis button in the **Data Type** field to launch the data type browser to browse the data types in the controller project. Select the user-defined **Mixer** data type.

Name	Alias For	Data Type	Description
MixerTag			



Name	Alias For	Data Type	Description
MixerTag		::MyController.Mixer	

Enter a description and category to specify where the property appears in the properties pane.

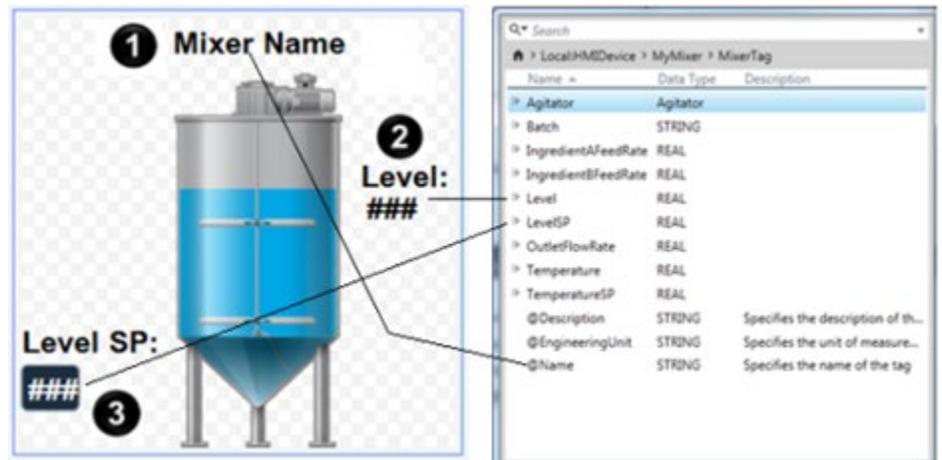
The design environment shows a graphic of a blue mixer tank. The graphic includes labels: 'Mixer Name' at the top, 'Level: ##' on the right side, and 'Level SP: ##' on the left side. Below the graphic is the 'Property Definition' table:

Name	Alias For	Data Type	Description	Category
MixerTag		::MyController.Mixer	Select Mixer Instance	Mixer

Use the property to bind elements within the Add-On Graphic to members of the Mixer data type in the controller. In this example, bind the level property of the mixer element to the level member of the Mixer data type in the controller. Select the mixer graphic element in the Add-On Graphic definition and bind the level property.



Click the ellipsis button to launch the tag browser and view the **MixerTag** property. Drill into the **MixerTag** property to browse the Mixer data type definition and select the Level member. Use the same method to tie the rest of the elements in the add-on graphic to members of the Mixer data type.



Item	Description
1	Mixer Name tied to @Name extended tag property
2	Level tied to Level
3	LevelSP tied to Level SP

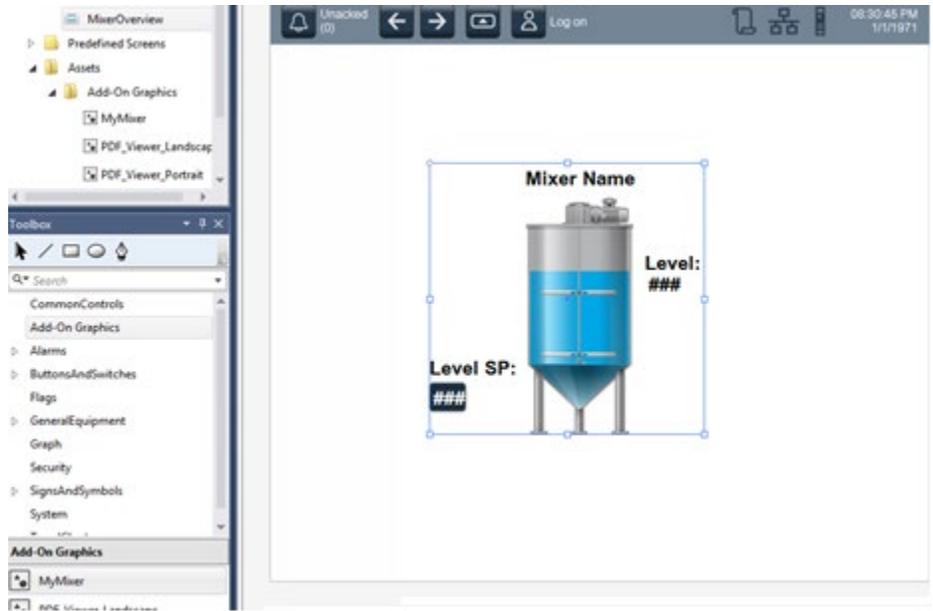
See also

[Create an instance for the Add-On Graphic definition](#) on [page 99](#)

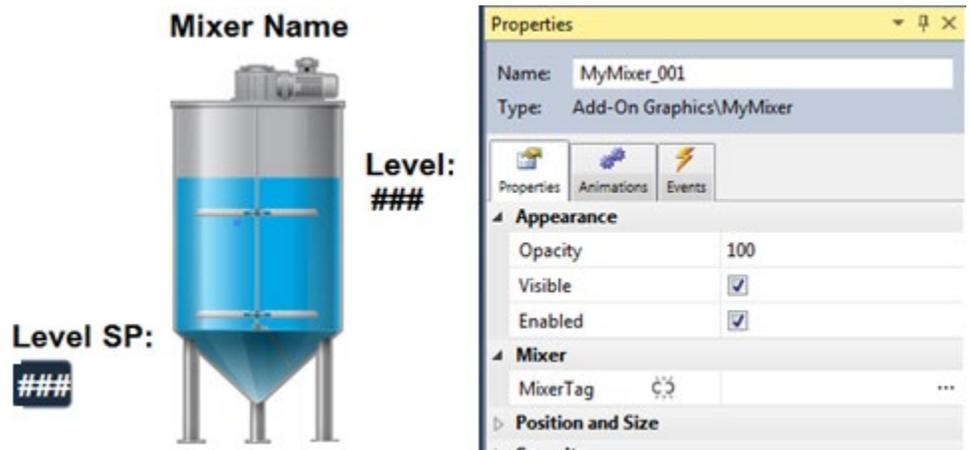
Create an instance for the Add-On Graphic definition

Add an Add-On Graphic instance to the Mixer Overview screen. Add-On Graphics appear in the **Toolbox** to add to a screen. In the **Toolbox Add-On Graphics** folder, select the **MyMixer** add-on graphic to add it to the **Mixer**

Overview screen.



Use the **MixerTag** custom property appearing in the **Mixer** area in the **Properties** tab to specify a specific instance of the mixer data type to pass into the add-on graphic at runtime. In this example, the instance of the Add-On Graphic to show data from Mixer1 in the controller.

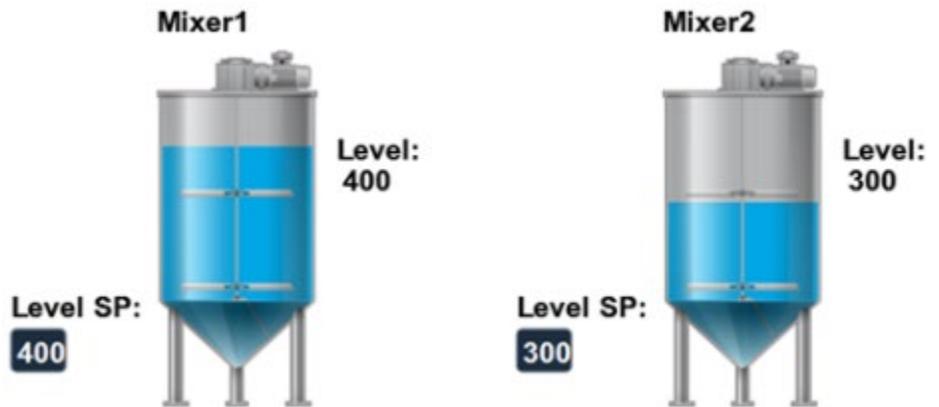


Click the ellipsis button to launch the tag browser and then bind the **MixerTag** property. The tag browser automatically filters by the **Mixer** data type and displays the tags in the controller of the type. This enables finding the Mixer1 instance of the data type by selecting **Mixer1**.

The required tag data for the add-on graphic instance passes through a single tag reference to create development savings through reuse.

Add another Add-On Graphic instance to the Mixer **Overview** screen to display data from Mixer2 in the controller. In the **Properties** tab, click the ellipsis button to launch the tag browser to bind the **MixerTag** property. Select **Mixer2**.

At runtime, the mixer Add-On Graphic instances on the Mixer **Overview** screen displays data for Mixer1 and Mixer2.



Changes to the definition of an Add-On Graphic automatically propagate to all instances. Open the mixer Add-On Graphic definition and add an indicator to display when the agitator motor of the mixer is running and then change the color of the mixer level.



Return to the **Mixer Overview** screen and verify that the changes exist in the instances of the mixer Add-On Graphic.



The Add-On Graphic definition has properties to manage revision changes. Select the **MyMixer** Add-On Graphic definition in **Project Explorer**. In the **Revision** area, change the major revision property to 2, and add a note in the definition to document changes to the Add-On Graphic.

View Designer enables reusing Add-On Graphics across projects using copy and paste to reduce development time. This also helps ensure consistent representation and operation of the graphic across a plant or facility.

In the source project, in the **Project Explorer**, right-click the **MyMixer** Add-On Graphic and select **Copy**.

In the target project, in **Project Explorer**, right-click the Add-On Graphics folder, select **Paste**. The **MyMixer** Add-On Graphic appears in the target project.

View Designer enables copying and pasting Add-On Graphic instances between projects. Copying and pasting the Add-On Graphic adds both the Add-On Graphic and the definition to the source project. This is useful when copying a screen between projects. Any Add-On Graphics used by the screen also includes their definitions.

If the source project already has an Add-On Graphic with the same name, you can easily manage name conflicts with options to keep both add-on graphics, update/replace an existing add-on graphic or cancel the operation.

See also

[Add-On Graphics](#) on [page 95](#)

Alarms

The PanelView 5000 HMI device terminals work with Logix controller alarms to display alarms on the PanelView 5000 HMI device. The PanelView 5000 HMI device displays Logix tag-based alarms using the PanelView 5000 HMI device version 5 and later with Logix controllers version 32 and later.

See also

[Add alarms to the system](#) on [page 103](#)

[View Designer alarm guidelines](#) on [page 104](#)

[Alarm Summary and Alarm Manager](#) on [page 106](#)

Add alarms to the system

To add an alarm to the system, open Studio 5000 Logix Designer and add an alarm instruction or a tag-based alarm.

Add tag-based alarms in Logix Designer by right-clicking on a tag in the tag editor and selecting Add Alarm. For more information on adding and configuring alarms in Logix Designer, see the online help in Logix Designer.

Configure alarms similarly for tag-based and instruction-based alarms. Logix Designer supports many alarm configuration options, such as the conditions for when the alarm occurs, the alarm message, and any associated tags sent along with the alarm.

Alarms configured in Logix Designer automatically appear on the PanelView 5000 HMI device for any Logix controller defined as a controller reference in **Project Properties** in View Designer. Any alarms added as an online edit in Logix Designer also appear on the PanelView 5000 HMI device.

See also

[View Designer alarm guidelines](#) on [page 104](#)

[Alarm Summary and Alarm Manager](#) on [page 106](#)

View Designer alarm guidelines

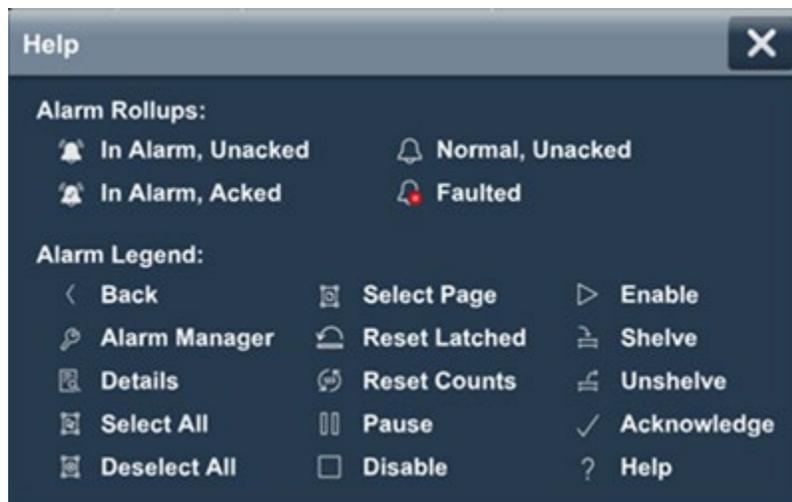
When alarms are active, the system banner displays an active alarm button with the total number of unacknowledged alarms. The alarm button blinks red to indicate active unacknowledged alarms. If all alarms are acknowledged, the system banner displays a solid red button. The alarm button returns to the default color when there are no active alarms.



Pressing the alarm button navigates to the predefined Alarm Summary screen that displays all the active or unacknowledged alarms.

In Alarm, Unacked (3)			In Alarm, Acked (0)			Normal, Unacked (0)			Faulted (0)		
#	Alarm	Inhibit	Event Time	Condition	Alarm Name	Message					
1			... 03-19-22 PM	HIHI	...r2 @Alarms.Level_HIHI	Mixer level HI-HI					
2			... 03-19-15 PM	HI	...ixer1 @Alarms.Level_HI	Mixer Level HI					
3			... 03-19-05 PM	HI	...ixer2 @Alarms.Level_HI	Mixer Level HI					

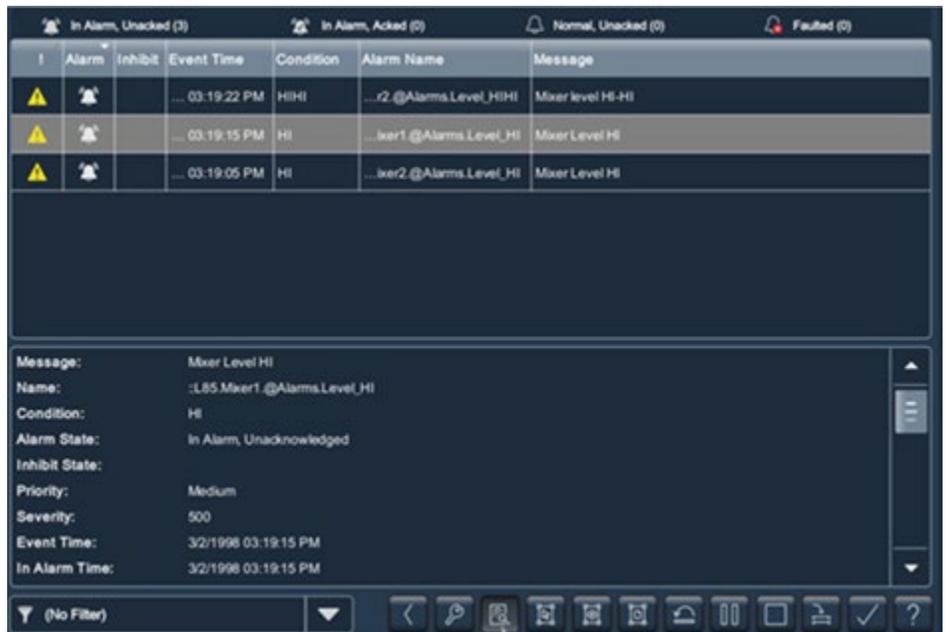
From the bottom of the Alarm Summary, use the buttons to perform operations on alarms, such as acknowledgment or shelving. Filter alarms using the filter list, and sort alarms by clicking on the alarm column headers. Pressing the button in the lower right shows a legend describing the operation of the buttons in the dialog box.



Press the **Alarm Manager** button to navigate to the predefined **Alarm Manager** screen. The **Alarm Manager** screen displays all the alarms configured in the Logix controllers and their current state.



Unshelving and re-enabling alarms from the Alarm Manager is available. The alarm summary and alarm manager have a **Details** pane that shows detail about an alarm, such as its severity, state transition time stamps, and associated tag values.

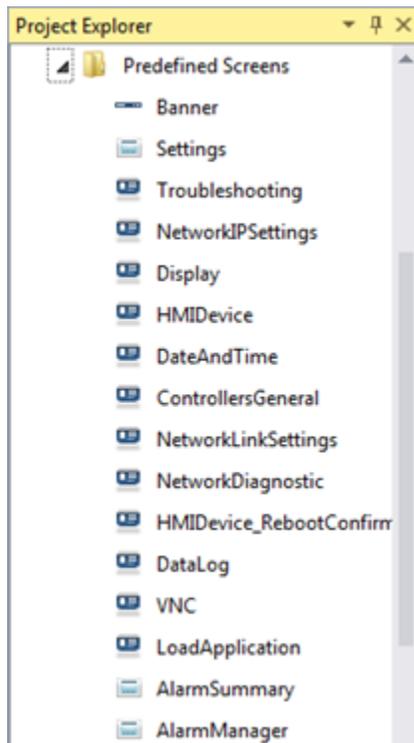


See also

[Alarm Summary and Alarm Manager](#) on [page 106](#)

Alarm Summary and Alarm Manager

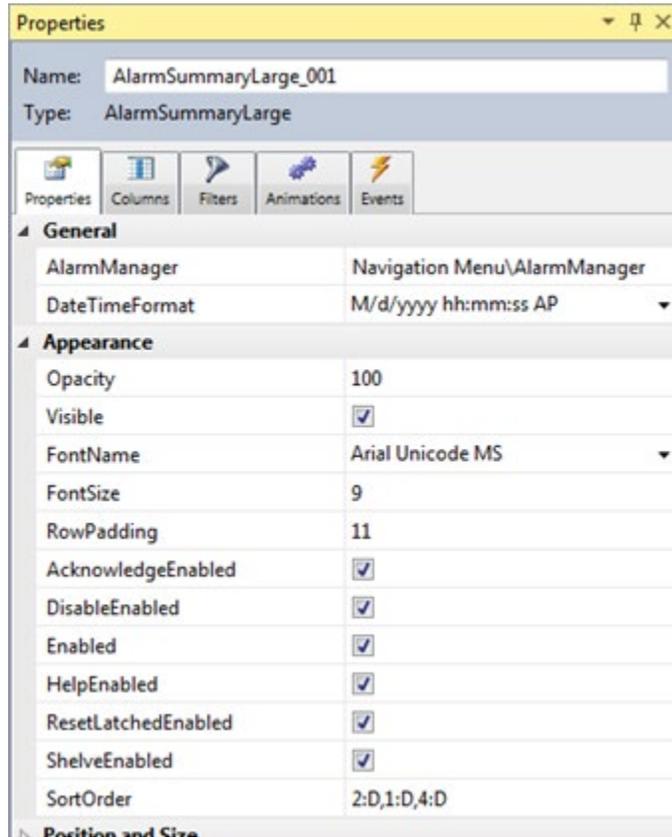
View Designer enables customizing the predefined Alarm Summary and Alarm Manager or creating new Alarm Summary and Alarm Manager screens. To customize the predefined screens, open the **AlarmSummary** or **AlarmManager** screens in the **Predefined Screens** folder in **Project Explorer**.



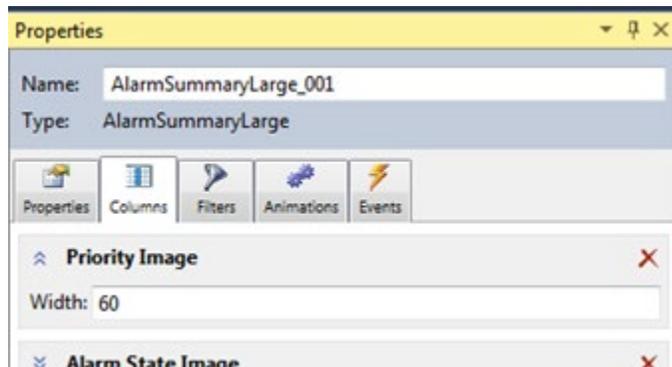
If opening the **AlarmSummary** screen, select the Alarm Summary table on the screen to configure the properties:

- **AlarmManager.** Navigates to the shortcut of the **AlarmManager** screen. Modify this property to move the **AlarmManager** shortcut to a different location in the navigation menu. This shortcut opens when pressing the **Alarm Manager** button on the **Alarm Summary**.
- **DateTimeFormat.** Specifies the date/time format used by fields on the alarm summary. Binding this property enables changing the date/time representation based on the current language.
- **RowPadding.** Changes the height of the rows. Modify the height of the rows to fit more rows on a page. Avoid making the value of the height too small. Reducing the height too much makes it difficult for an operator to press a specific row.
- **AcknowledgedEnabled** through **ShelveEnabled.** Turn these properties on and off to change whether a specific button is enabled on the alarm summary. Binding these properties allows enabling based on the current user role.

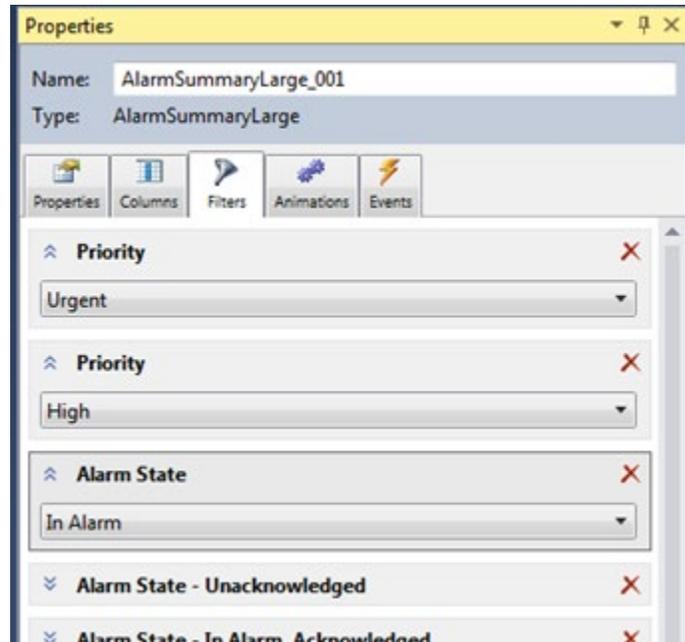
- **SortOrder.** Defines the sort order used when the alarm summary appears. Define the order by entering a valid column number, colon, and a D for descending or A for ascending, followed by a comma if adding a secondary or tertiary sort.



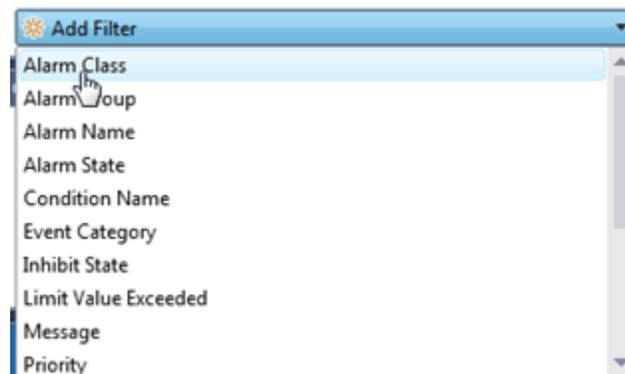
On the **Columns** tab, specify the visible columns and column widths:



On the **Filters** tab, define multiple filters to display specific alarms on the Alarm Summary table:



Add a new filter at the bottom of the **Add Filter** list:



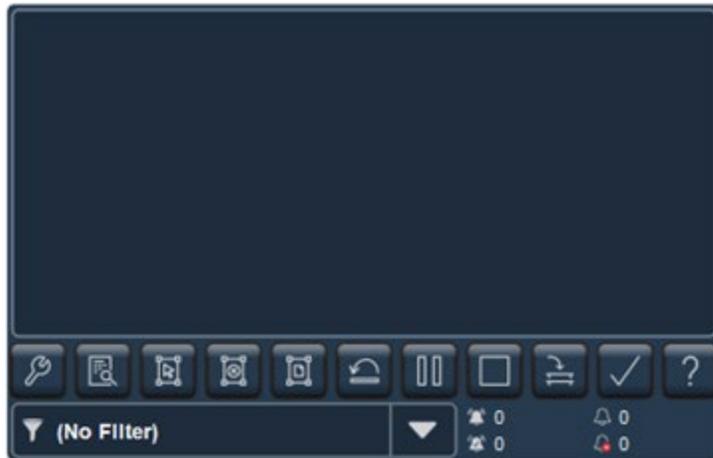
When applying a filter, the **Filter** field on the Alarm Summary table displays the applied filter and turns solid orange. This indicates that not all alarms are always visible.

The **FilterIndex** property defines the default filter applied to the alarm table when the screen with the table opens. Bind to the **FilterIndex** property or create a custom **Alias** property aliased to the **FilterIndex** property of the Alarm Summary.

When the screen opens, the numeric value in the **FilterIndex** property displays the filter that corresponds to the alphabetical order of filters on the **Filters** tab. For example, the FilterIndex 3 applies the third filter when the screen containing the Alarm Summary tables opens. The operator can then change the filter at runtime as required. 0 in the **FilterIndex** property applies no filter to the Alarm Summary table.

If necessary, add the **AlarmSummary** and **AlarmManager** graphic elements to user-specified screens. Select the **Alarms** category in **Toolbox** to select the Large and Medium Manager and Alarm Summary graphic elements.

Use the large elements for ten inch and larger PanelView 5000 terminals. Use the medium elements for smaller terminals. Following is an example of an Alarm Summary Medium:



The **Alarm Summary Medium** graphic element:

- Displays the same information as the large element
- Has fewer rows of alarms
- Does not have a back button
- Uses icons to represent the alarm roll-up information

See also

[Alarms](#) on [page 103](#)

Alarm History

Use the Alarm History Viewer to view up to 40,000 of the most recent alarm transitions. Every time an alarm transitions to a new state, such as **In Alarm** or **Acknowledged**, the HMI device stores a record in the alarm history. Once the number of records reaches 40,000, the HMI device overwrites the oldest records.

Place the **Alarm History Viewer Large** or the **Alarm History Viewer Medium** graphic elements from the **Toolbox** onto a user-defined screen to display the alarm history. Use the large element on 10-inch or larger PanelView 5000 HMI devices. Use the medium element with smaller HMI devices.

Alarm	Inhibit	Event Time	Condition	Alarm Name	Message
⚠	🔔	12/17/2019 08:40:14 PM	HIHI	::L85.Mixer1.@Alarms.Level_HIHI	Mixer level HI-HI
⚠	🔔	12/17/2019 08:40:11 PM	TRIP	...rray.@Alarms.ConveyorSta3Jam	Jam on conveyor station 3
⚠	🔔	12/17/2019 08:40:09 PM	TRIP	...5.BOOLarray.@Alarms.M101Trip	Motor 101 Trip condition
⚠	🔔	12/17/2019 08:39:59 PM	TRIP	...5.BOOLarray.@Alarms.M101Trip	Motor 101 Trip condition
⚠	🔔	12/17/2019 08:39:57 PM	HI	::L85.TestAnalog	Motor Overspeed warning
⚠	🔔	12/17/2019 08:39:45 PM	HIHI	::L85.Mixer1.@Alarms.Level_HIHI	Mixer level HI-HI
⚠	🔔	12/17/2019 08:39:33 PM	HI	::L85.TestAnalog	Motor Overspeed warning
⚠	🔔	12/17/2019 08:39:29 PM	TRIP	...5.BOOLarray.@Alarms.M101Trip	Motor 101 Trip condition

The Alarm History supports sortable columns and filtering:

- To configure columns, select the **Column** tab in View Designer,
- To configure filters, select the **Filters** tab.

Properties

Name: AlarmHistoryViewerLarge_001

Type: AlarmHistoryViewerLarge

Properties Columns Filters

- Priority Image (Width: 60)
- Alarm State Image (Width: 60)
- Inhibit State Image
- Event Time
- Condition Name
- Alarm Name
- Message
- Add Column

For more information, refer to configuration of Alarm Summary and Alarm Manager.

Alarm History allows also to filter the records by time. Select the Time Filter field and enter a date and time according to the displayed format.



Select the checkbox on the right to apply the filter. The Time Filter field turns orange. The Alarm History shows only the alarm transitions that occurred after the applied time filter. The filter count shows the number of visible records and the total alarm records available.



Tips:

- To change the time format, edit the value of DateTimeFormat property on the **Properties** tab of the Alarm History .
- If you only enter a date, the filter uses a time of 12 AM or 0:00.
- Empty minutes or seconds value starts the filter at a full hour or minute.

To show the most recent alarm transitions, select the Refresh button .

The alarm history records persist on the HMI device through reboots, power cycles and downloads of new projects. The update of HMI device's firmware clears the alarm history records stored on the device. To clear the alarm history, configure a custom button or a project event with the Clear Alarm History command.

Export Alarm history

Export the alarm history to perform external analysis of the alarm transitions from your machine or process. To export, navigate to the predefined **Data Export** settings screen:

On the screen:

1. In **What to export**, select **Alarm history**.
2. Enter a name for the export file.
3. Choose a destination for the export file.
4. Select **Export**.

The system saves the Alarm history as a .zip file containing a .csv file. You can open the .csv file in tools such as Microsoft Excel for further investigation.

Automatic Diagnostics

The Automatic Diagnostics feature works with Logix controllers at revision V33 and later to automatically provide fault information about your Rockwell Automation control system. If devices connected to the Logix controller experience fault conditions, the Logix controller will automatically send this information to the HMI device, making it easier for you to notice and then troubleshoot fault conditions. The View Designer does not require any additional configuration if you use the predefined screens and system banner indication.

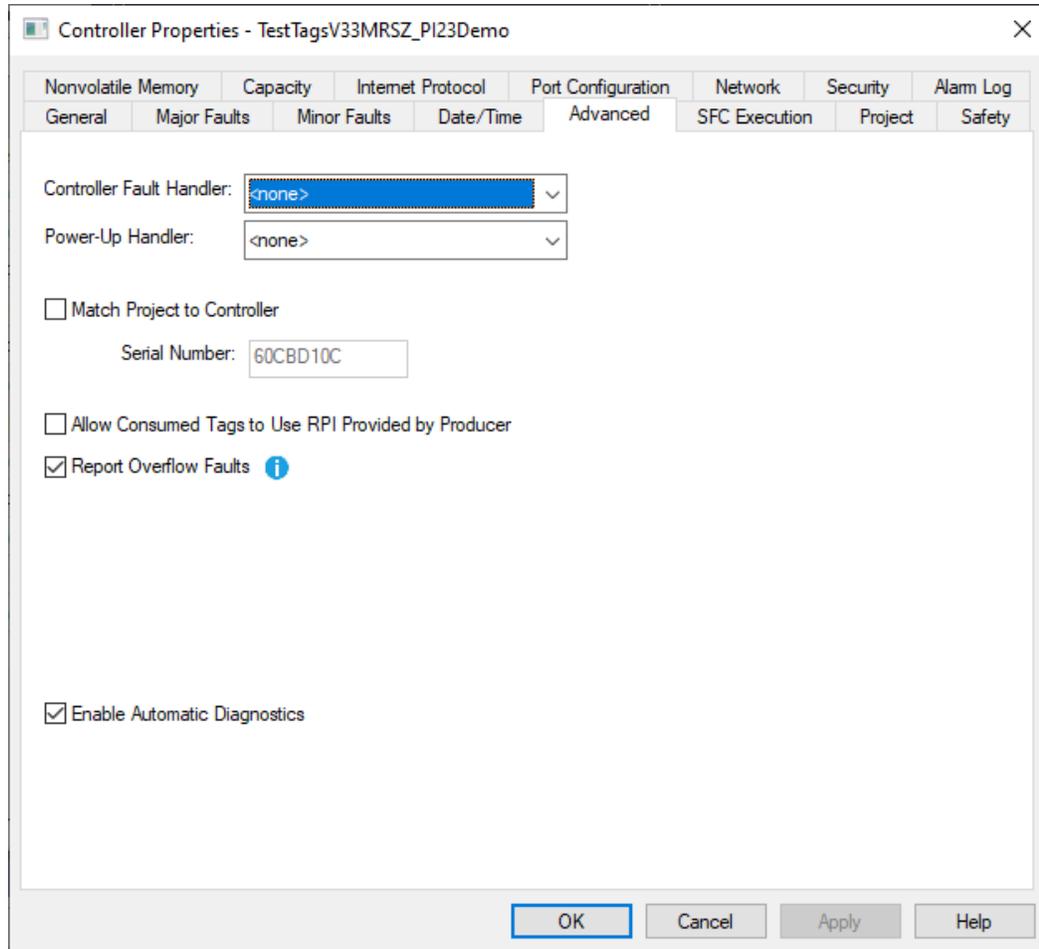
Automatic Diagnostics messages are automatically available from the controller in English, Portuguese, French, Italian, German, Spanish, Chinese, Japanese, and Korean. When you switch the language on the HMI device, the messages will be presented in that language.

Enable Automatic Diagnostics

To enable Automatic Diagnostics, first enable the feature in your Logix controller:

1. Open Studio 5000 Logix Designer.
2. Open the Advanced tab on the Controller Properties dialog.

3. Select Enable Automatic Diagnostics.



This information is now available on the HMI device.

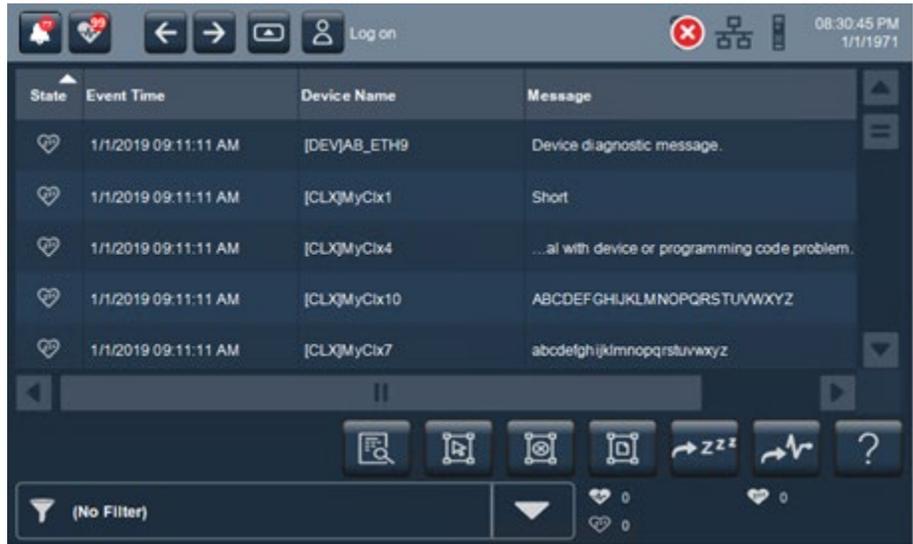
The Automatic Diagnostics displays on the HMI device on the System Banner using the Automatic Diagnostics Status Indicator button:



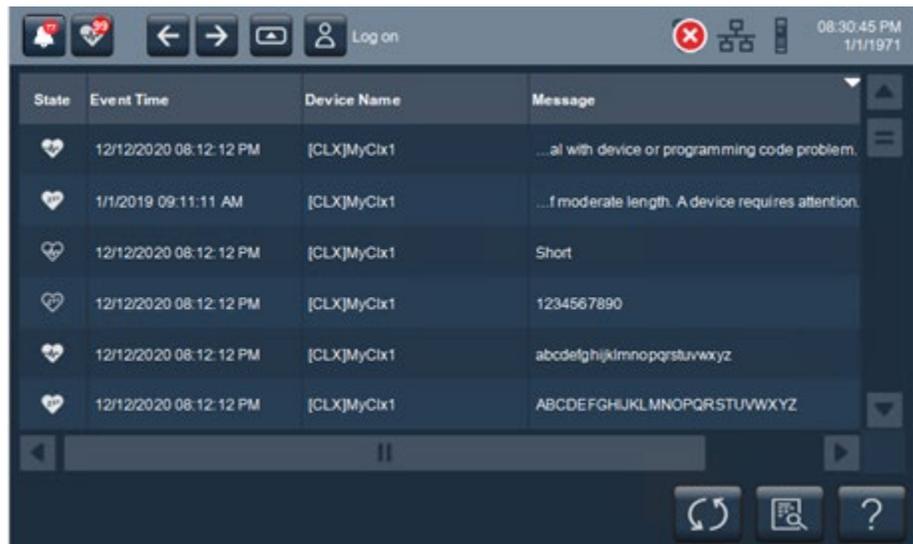
The button shows a notification icon with the total number of Active, Unsuppressed diagnostics messages. The numbered notification disappears if there are no Active, Unsuppressed diagnostics. Press the button to navigate to the Predefined Automatic Diagnostics screen.

Automatic Diagnostics and Automatic Diagnostics History

There are two predefined screens for Automatic Diagnostics and Automatic Diagnostics History in View Designer. The Automatic Diagnostics screen shows current diagnostics and allows you to suppress diagnostics if, for example, a piece of equipment is undergoing maintenance, and you temporarily do not want to see the diagnostic message from that equipment. To customize the predefined Automatic Diagnostics screen, open the **AutomaticDiagnostics** screen under Predefined Screens in the Project Explorer.



The Automatic Diagnostics History screen shows the most recent 1000 diagnostic events. The history remains through power cycles and project downloads. The history is cleared during a firmware update or by executing the **Clear Diagnostic History** command. To customize the predefined Automatic Diagnostics History screen, open the **AutomaticDiagnosticsHistory** screen under Predefined Screens in the Project Explorer.



If you want to make your own screen to view diagnostics, place an Automatic Diagnostics Large or Automatic Diagnostics Medium graphic element on your

screen. To make your own diagnostics history screen, place an Automatic Diagnostics History Large or Automatic Diagnostics History Medium element on your screen. Use the large elements on 10 inch or larger HMI devices. Use the medium elements with smaller HMI devices.

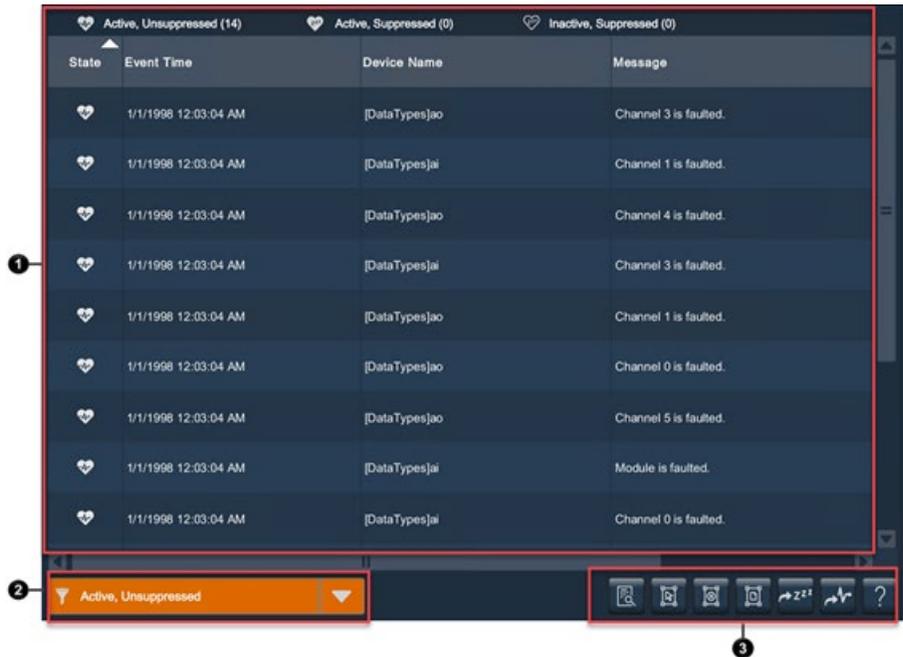
You can modify the Automatic Diagnostics and Automatic Diagnostics History graphic elements by changing their properties such as DateTimeFormat or SortOrder and by configuring what columns are available. Refer to the chapter Alarms for more information since these configurations are very similar to the alarm graphic element configurations.

See also

[Alarm Summary and Alarm Manager](#) on [page 106](#)

Automatic Diagnostics at Runtime

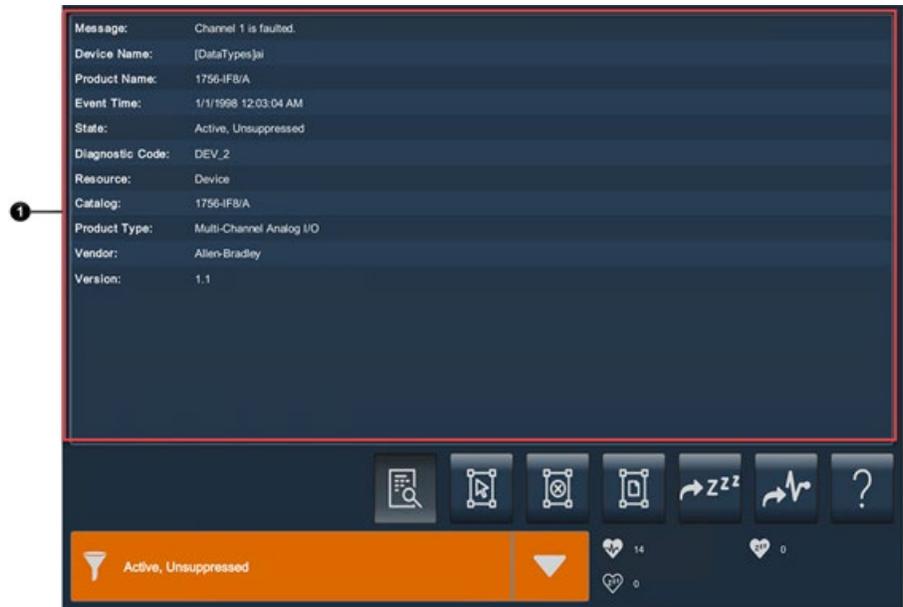
At Runtime, the Automatic Diagnostics element provides the following indications and interactions:



Item	Description
1	The list of automatic diagnostics notifications.
2	Filter list. Select a filter to show diagnostics in certain states. Select (No Filter) to show all diagnostics regardless of their status. Select Active, Unsuppressed to display only diagnostics showing active faults which the operator has not suppressed. Select Suppressed to display diagnostics which the operator previously suppressed.
3	Buttons to perform actions available for the Automatic Diagnostics Viewer: <ul style="list-style-type: none">  Details hides or shows the Details pane, which contains details of the currently selected diagnostic.  Select All selects all of the diagnostics in the diagnostics table, including those not displayed on the current page of diagnostics.  Deselect All deselects all diagnostics in the diagnostics table, including those not displayed on the current page of diagnostics.

Item	Description
	Select Page selects all of the diagnostics displayed on the current page of diagnostics.
	Suppress set selected diagnostics event to the suppressed state.
	Unsuppress set selected diagnostics event to the unsuppressed state.
	Help opens the Help popup. This popup displays a legend of icons and the task each button on an diagnostics table performs.

The **Details** pane appears on the bottom half of large diagnostics tables. The **Details** pane replaces the list of diagnostics in medium automatic diagnostics tables. If no record is selected, the Details pane is empty.

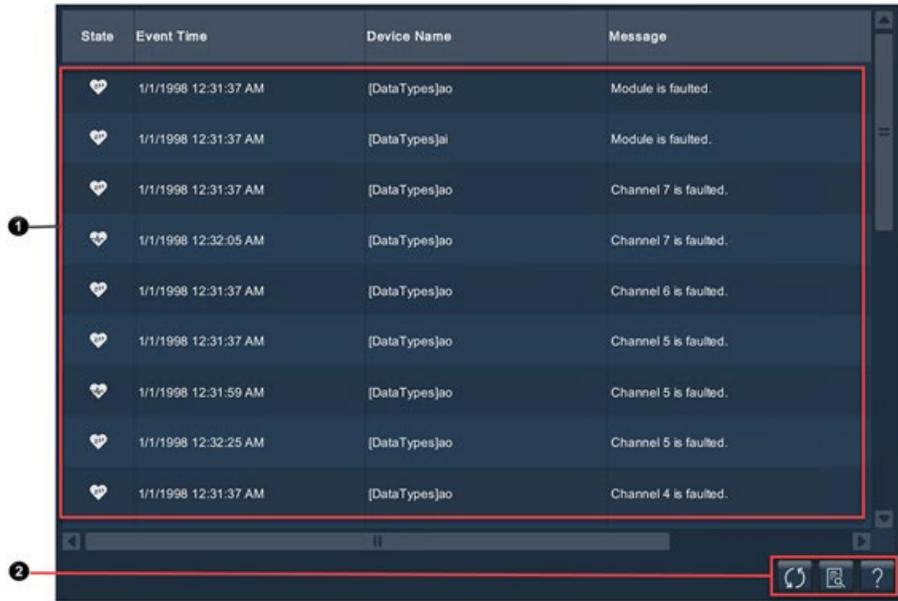


The **Help** popup displays a legend of icons and the task each button on an diagnostics table performs.



Automatic Diagnostics History at Runtime

At Runtime, the Automatic Diagnostics History element provides the following indications and interactions:



Item	Description
1	The list of automatic diagnostic state change records from the HMI device.
2	Buttons to perform actions available for the Automatic Diagnostics History Viewer:
	Details hides or shows the Details pane, which contains the details of the currently selected diagnostics state change record. The Details pane appears on the bottom half of large automatic diagnostics history tables. If no record is selected, the Details pane is empty.
	Refresh loads the latest diagnostics state changes from the automatic diagnostics history. The operator has to refresh the table to see the latest changes. The operator can also exit and enter the screen with the Automatic Diagnostics History again to refresh.
	Help opens the Help popup.

The **Help** popup displays a legend of icons and the task each button on an automatic diagnostics table performs.



Use images

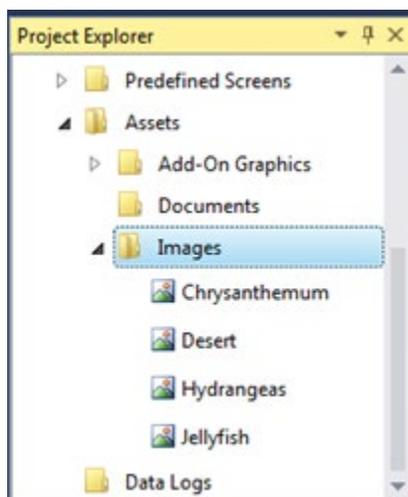
Add images to projects and place them on screens to enhance the appearance of the screens.



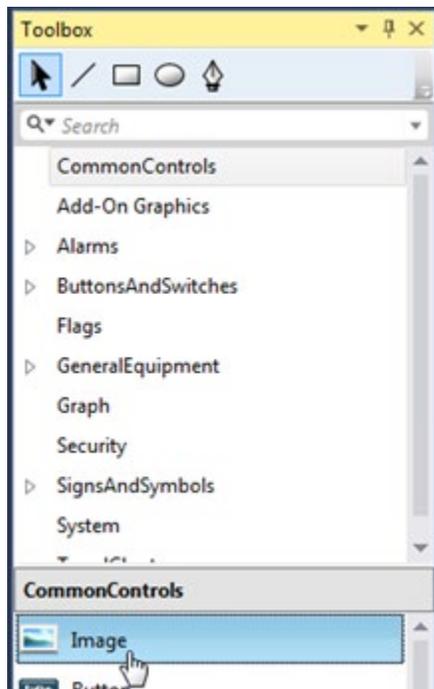
Tips:

- View Designer supports JPG, BMP, PNG, and SVG image types
- For best screen switching performance, do not add images to a project with more resolution than needed. For example, if using a JPG image as a screen background on a 10-inch PanelView 5510 HMI device with 800x600 resolution, save the image as an 800x600 pixel JPG and then add it to the project.
- View Designer gives an error message if you add an image which uses too much memory. Decrease the resolution of the image and try to add it again.
- There are some special guidelines you must follow when using SVG images:
 - Supported SVG files include Tiny 1.1 and Tiny 1.2 SVG files.
 - View Designer does not support the following within the SVG image definition:
 - Scripting
 - Animation
 - Units of %, pica, or cm
 - Linked images
 - To ensure the Image graphic element displays the SVG image, adhere to the following guidelines when creating an SVG file:
 - Use Adobe Illustrator CS6 to create the image and save it as an SVG type.
 - In the SVG Options dialog box, select the following:
 - SVG Profiles box: SVG Tiny 1.2
 - Location box: Embed
 - Avoid using text in the SVG image. If including text in the SVG image, in the **Fonts Type** box, select **Convert** to outline.

To add images to a project, in **Project Explorer**, right-click the **Images** folder and select **New Images**. Select one or more images on the computer and select **Open**. The Images appear in the **Images** folder.



To add an image to a screen, double click or drag the **Image** toolbox element onto the screen. The **Select Image** dialog box opens.



On the **Select Image** dialog box, select an image to appear on the **Image** element screen. Dragging an image onto the screen from the **Images** folder also adds the image to the screen.

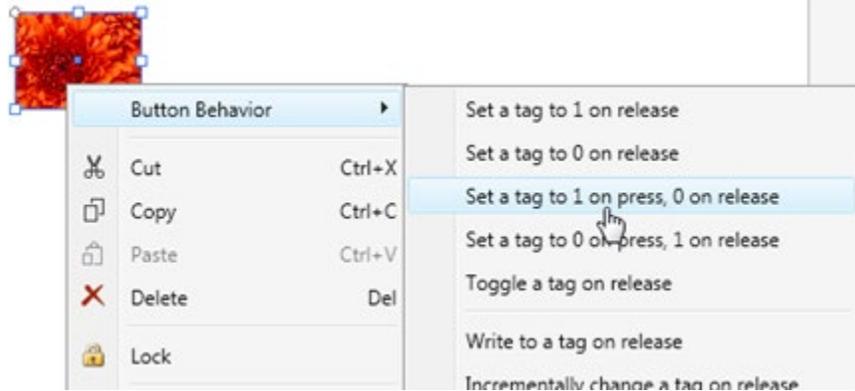
See also

[Example 1: Configure the image to work as a momentary button](#) on [page 120](#)

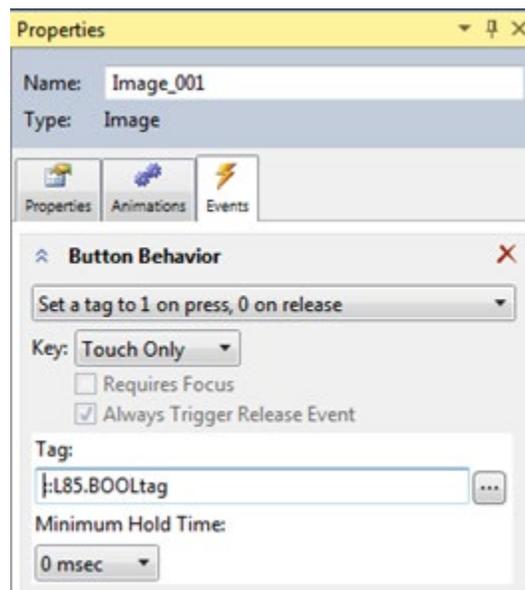
[Example 2: Change images displayed in an image element at runtime](#) on [page 121](#)

Example 1: Configure the image to work as a momentary button

Images support state tables and events. In this example, turn the image into a button by right-clicking the image and selecting **Button Behavior**. Then select the desired button behavior. For this example, the image is a momentary button.



Update the **Events** card to configure the tag value that changes when pressing the graphic element.



See also

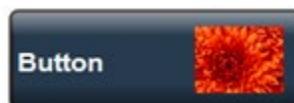
[Example 2: Change images displayed in an image element at runtime](#) on [page 121](#)

Example 2: Change images displayed in an image element at runtime

View Designer enables changing the images displayed in an Image element at runtime by either:

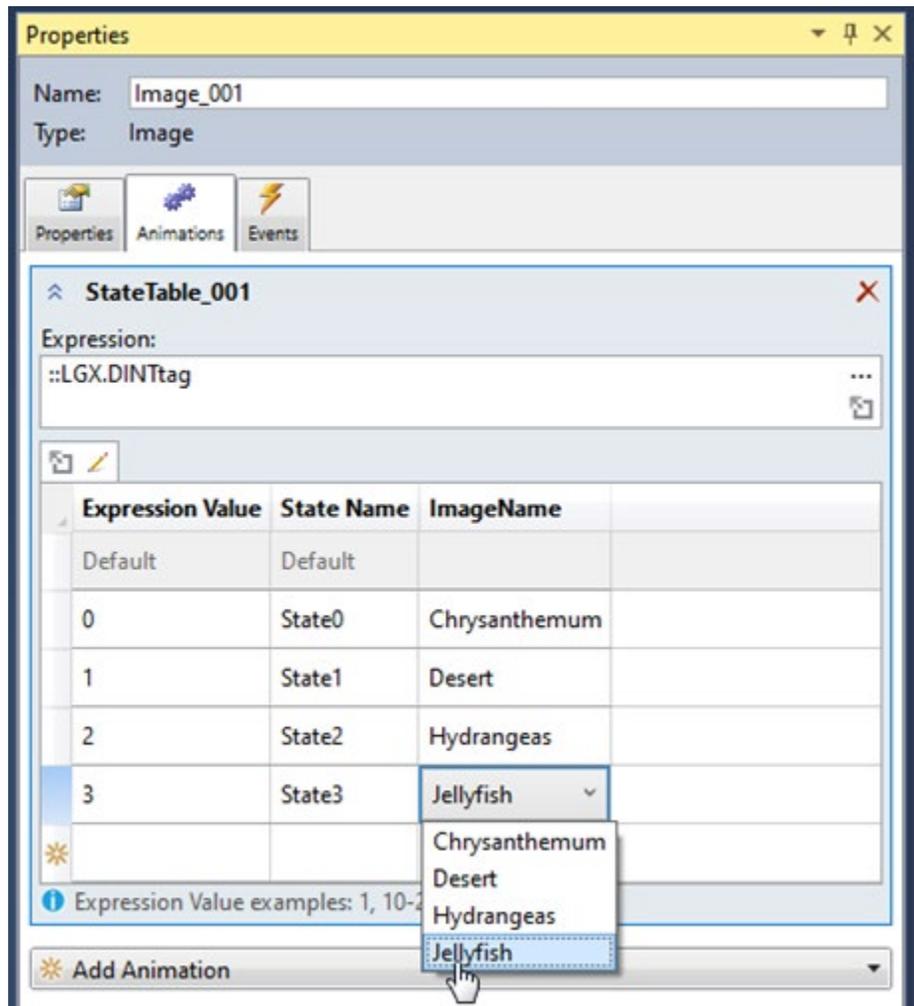
- Layering multiple image graphic elements and use the visibility properties of the graphic elements to display the images.
- Modify the **ImageName** property of the **Image** graphic elements. This example uses this method.

For example, add an image decoration to a button.



In this example, configure the image to change according to the state of a machine. A DINTtag represents the state of the machine.

Create a state for the image graphic element using an expression of DINTtag and use four states to represent the different pictures to use.



The state table uses the **ImageName** property to change the image displayed in the image graphic element. Select the names of the image files added to the Images folder in the Project Explorer.

In runtime, as the value of DINTtag cycles between the values of zero through three, different image decorations appear on the button display.



Alternatively, bind the **ImageName** property of the Image element to a string tag in the controller. As the string tag changes to different image names, the Image element displays the matching image file. Note that the text strings must exactly match (including case) the image names in **Project Explorer**.

See also

[Use images](#) on [page 119](#)

Language switching

Use Language switching on the PanelView 5000 HMI device to view text in a user-specified language at runtime. View Designer supports up to 20 languages.

When using language switching with a PanelView 5000 HMI device, content from the Logix controller also displays in the selected language. This includes tag descriptions, engineering units, Boolean state identifiers, and alarm messages. For more information about defining multiple language strings for tag descriptions and alarm messages in Logix Designer, access the Documentation Languages topics in the Logix Designer help.

See also

[Configure language switching](#) on [page 125](#)

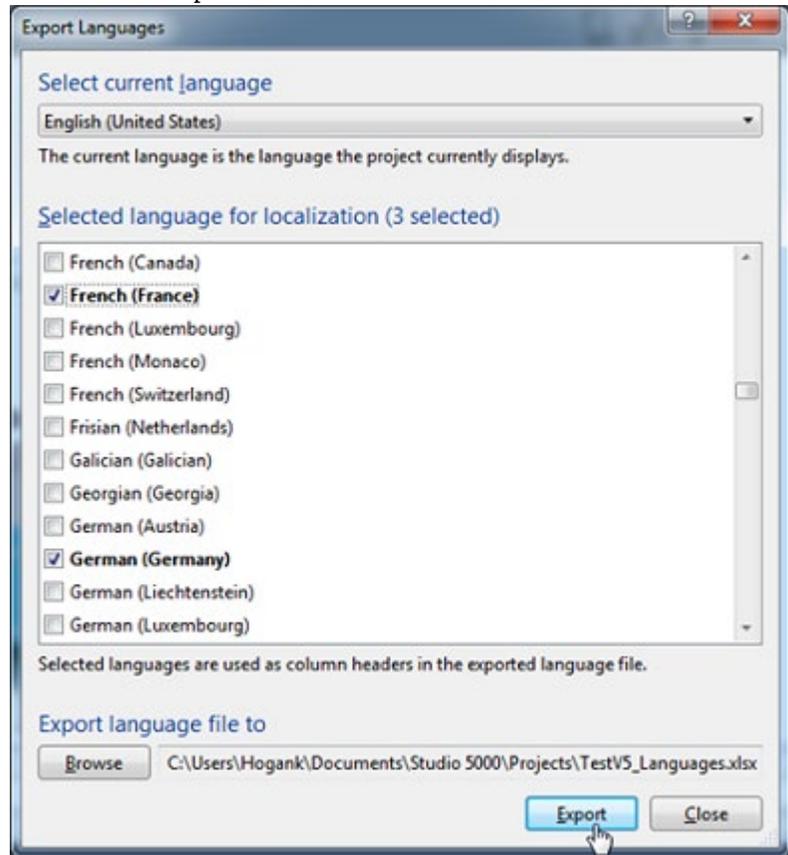
Follow this procedure to select the language to use on the PanelView 5000 HMI device.

Configure language switching

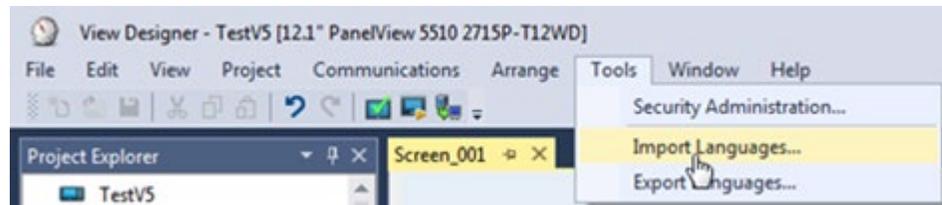
To configure language switching

1. After creating the HMI project, select **Tools > Export Languages** from the main menu.
2. On the **Export Languages** dialog box, configure the language used for developing all screens. This language appears in the first language

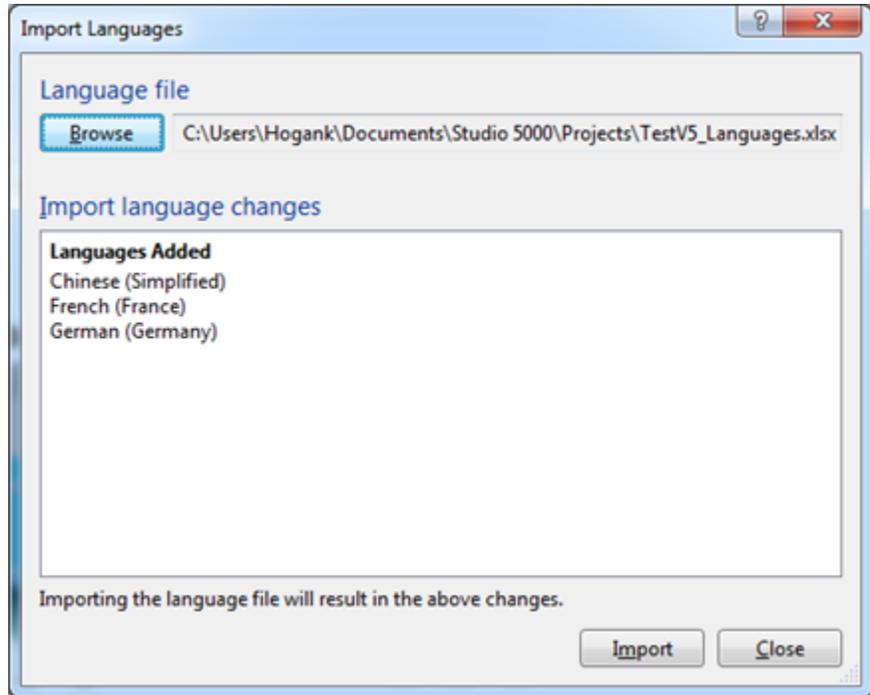
column in the spreadsheet.



3. Select other languages to use on the PanelView 5000 HMI device. The languages appear in the additional language columns on the spreadsheet.
4. Determine the location to export the language spreadsheet to and select **Export**. The default languages spreadsheet xlsx file format is compatible with Microsoft Excel.
5. Add the translations for the different languages in the spreadsheet.
6. After entering all the translations, from the main menu, select **Tools > Import Languages** to import the languages to View Designer.



- In the **Import Languages** dialog box, select **Import** to import the languages spreadsheet. The spreadsheet displays the changes made to languages previously imported to the project. The **Import Languages** dialog box displays languages added, changed, and deleted. A deleted language does not appear in the import spreadsheet. The View Designer project has the added languages.

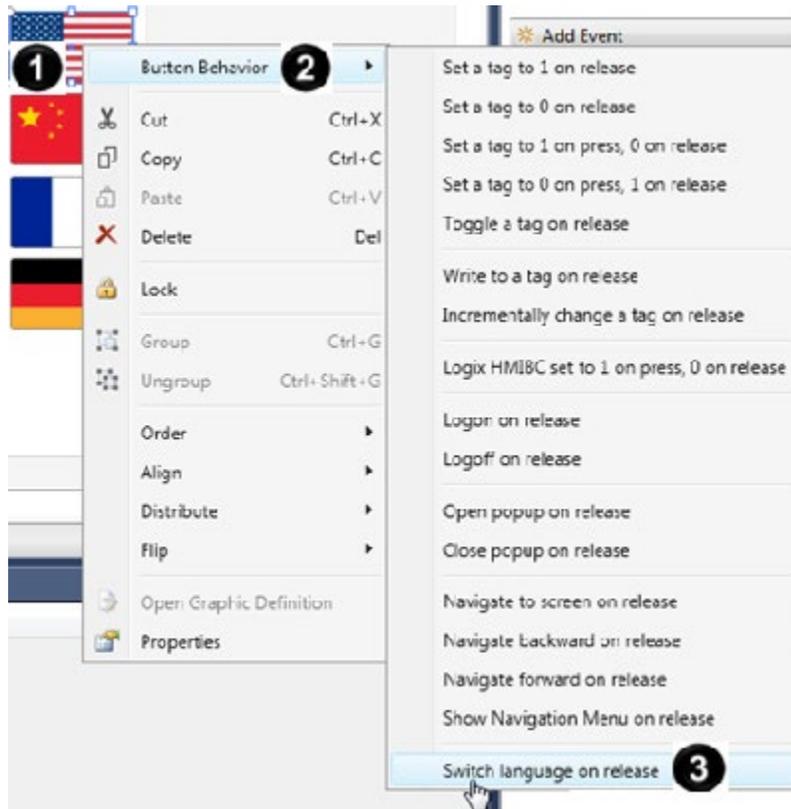


See also

[Example: Configure language commands to switch language](#) on [page 127](#)

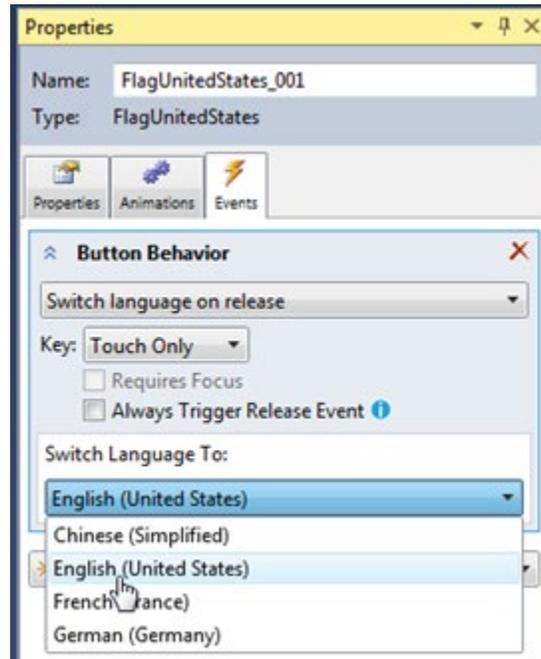
Example: Configure language commands to switch languages

The **Switch Language** command is compatible with any event type. This example shows configuring a group of flags on a screen as buttons to switch to the different languages. First, right-click on the flag and select **Button Behavior > Switch language on release**.

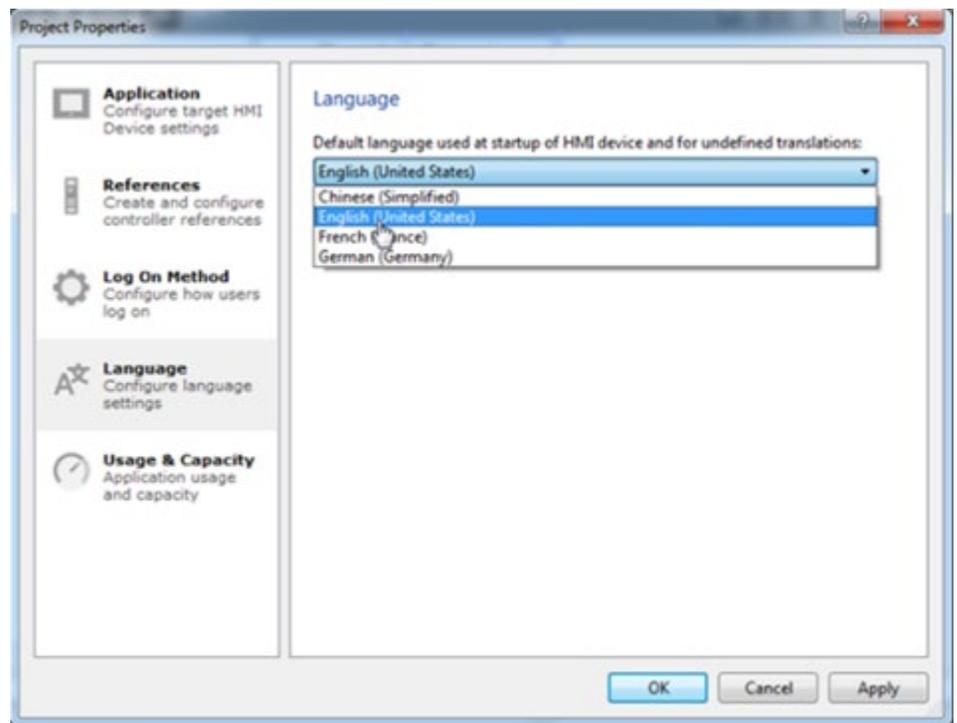


Item	Description
1	Flags
2	Button Behavior
3	Switch language on release

Next, in the **Events** tab in the **Properties** dialog box, select the language for the flag. Repeat this step for additional flags. When configuring the language command, only languages imported to the View Designer project are available.



Define the default language for starting the PanelView 5000 HMI device and for the scenario of missing a language translation. On the **Project Properties** dialog box, select **Language**.



Download the project.



Tip: View Designer enables selecting a different default language when downloading the project to the PanelView 5000 HMI device.

Use the **ActiveLanguage** HMI Device tag to display the selected language at runtime.

Name	Data Type	Description
ActiveLanguage	STRING	Displays the active langua...
AmbientTemp	DINT	Displays the ambient tem...
ApplicationName	STRING	Displays the name of the...
ControllerCount	DINT	Displays the number of co...

PanelView 5000 HMI devices support displaying unicode text strings using STRING tags in a Logix controller. The PanelView 5000 supports UTF-8 encoding to read and write string tags. UTF-8 encodes multi-byte character sets by breaking down Unicode characters into multi-byte representations. The Logix controller also uses UTF-8 encoding to store complex characters in the STRING data type for display on the PanelView 5000 HMI device. For example, entering the bytes d094 in a STRING tag in Logix Designer displays the Cyrillic character Д.



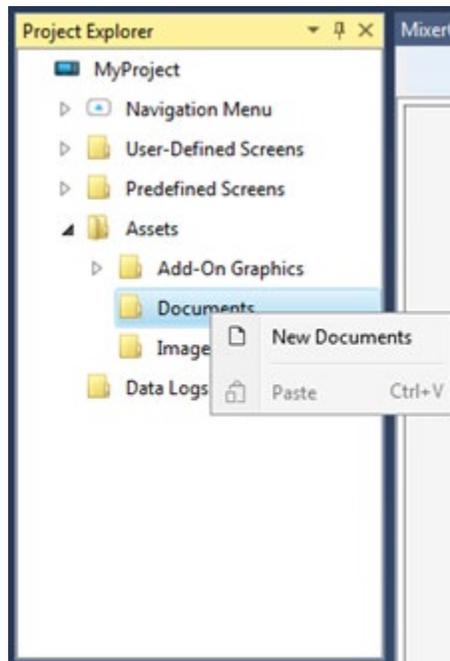
The Cyrillic character Д appears on the PanelView 5000 Text Display bound to STRINGtag.

See also

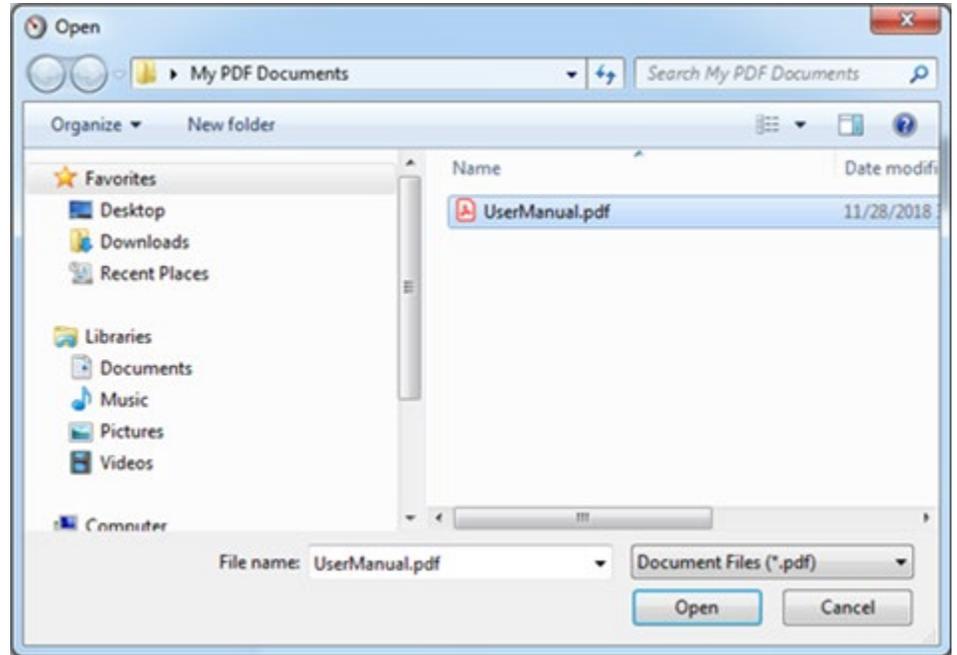
[Language switching](#) on [page 125](#)

PDF Viewer

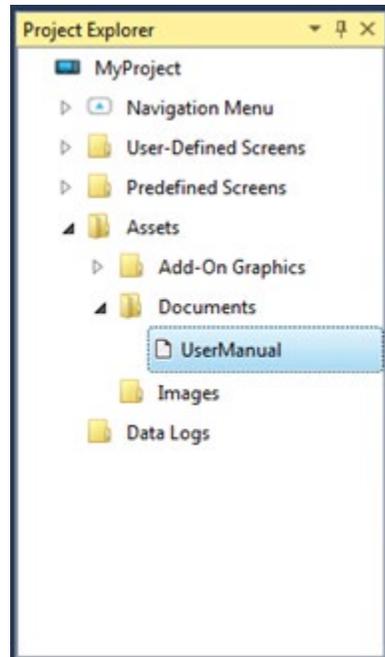
The PDF Viewer displays a PDF document on a PanelView 5000 HMI Device to view documentation, operating procedures, and machine setup instructions. First, add a PDF document to view on an HMI device in the **Assets > Documents** folder of the **Project Explorer**. In the **Project Explorer**, expand the **Assets** folder, right-click the **Documents** folder, and click **New Documents** to browse for PDF files.



Select the PDF document and click **Open**.



The PDF document appears in the **Documents** folder.



Add the PDF document on a screen or popup using either a:

- PDF Viewer Add-On Graphic. A Pre-configured Add-On Graphic with buttons that have event commands to display, navigate, and zoom the referenced PDF document on the HMI device. A PDF Viewer Add-On Graphic does not require customization.
- PDF graphic element. Supports touch support for navigating and panning pages within the referenced PDF document. A PDF graphic element has no pre-configured buttons to navigate or zoom the PDF document. Customize a PDF graphic element to add buttons and event

commands to navigate and zoom the PDF document on the HMI device.

See also

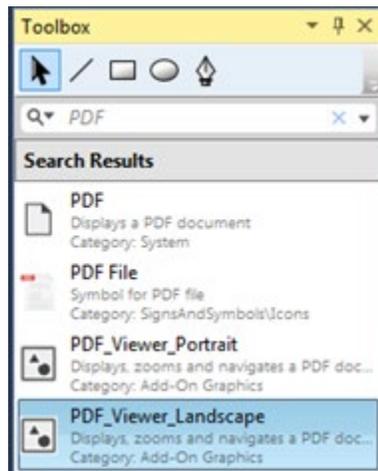
[Example 1: Add a PDF Viewer Add-On Graphic to a screen](#) on page 133

Example 1: Add a PDF Viewer Add-On Graphic to a screen

Add a PDF Viewer Add-On Graphic to a screen by searching for *PDF* in the **Toolbox**. Select a PDF Viewer Add-on Graphic that corresponds to the target HMI device screen size.

- `PDF_Viewer_Landscape`. Occupies the entire screen on a 10-inch HMI device and a portion of larger HMI devices.
- `PDF_Viewer_Portrait`. Occupies half of the screen on a 10-inch HMI device.

In this example, select the `PDF_Viewer_Landscape` Add-On Graphic.



Double-click the `PDF_Viewer_Landscape` Add-On Graphic in the **Toolbox**.



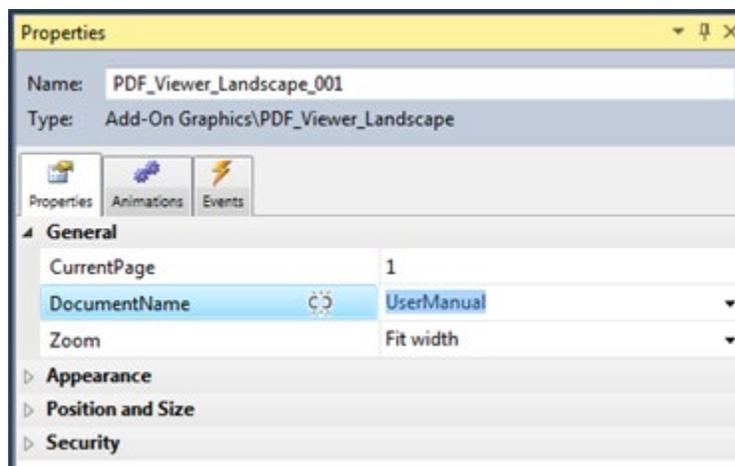


Tip:

Expanding the **Assets > Add-On Graphics** folders and then dragging the PDF_Viewer Add-On Graphic or right-clicking the PDF_Viewer Add-On Graphic and selecting Add Add-on Graphic to (screen name) also adds a PDF Viewer Add-On Graphic to a screen or popup from Project Explorer.

In the **Properties** window, expand the **General** category and configure the required properties:

- **DocumentName.** Select the PDF document that the Add-On Graphic displays on the HMI device.
- **PageNumber.** Type the page number of the PDF document that initially opens on the HMI device. The page displays in View Designer.
- **Zoom.** Select the initial zoom factor for the HMI device. The zoom setting also appears in View Designer.



The User Manual PDF displays in the PDF Viewer.



See also

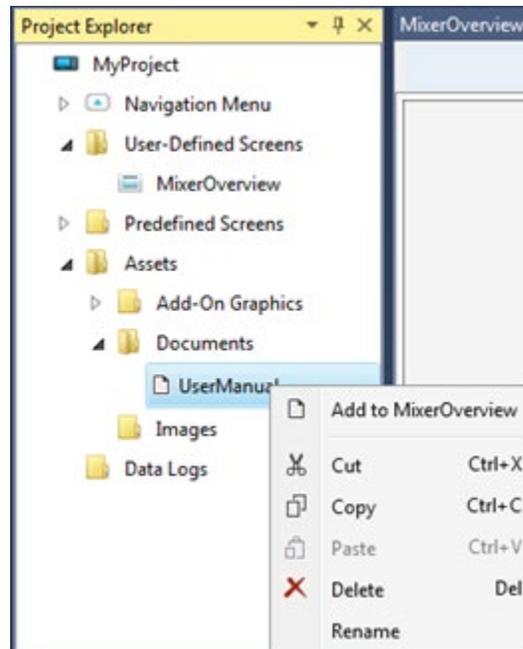
[Bind bookmark properties of a PDF graphic element](#) on [page 143](#)

[Example 2: Drag a PDF document onto the screen](#) on [page 135](#)

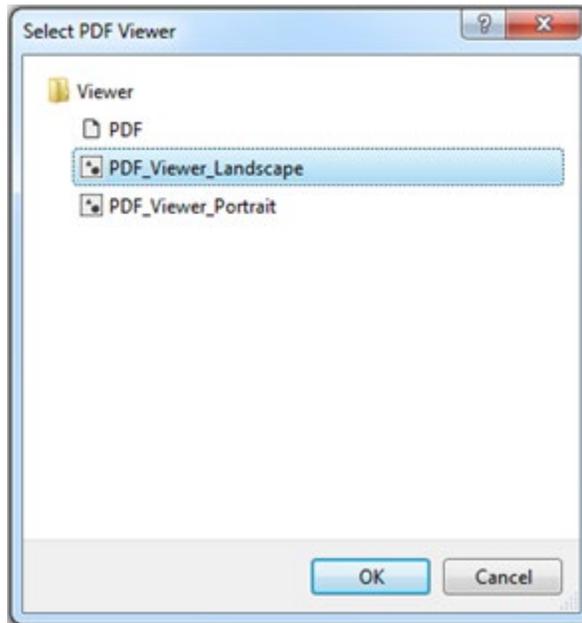
[Example 3: Create custom PDF Viewer](#) on [page 138](#)

Example 2: Drag a PDF document onto the screen

Add a PDF document to a screen or popup from the Project Explorer by expanding the **Assets > Documents** folder and then dragging the document or right-clicking the document and selecting **Add** to (screen name).



When the **Select PDF Viewer** dialog box appears, select the PDF Viewer element type. This example will use the PDF_Viewer_Landscape Add-On Graphic.



At runtime, the PDF_Viewer_Landscape Add-On Graphic displays buttons to control viewing the PDF document:



Item	Description
1	The PDF document added to the PDF Viewer Add-On Graphic through the DocumentName property.
2	 Bookmarks displays the bookmarks on the left side of the PDF document on the HMI device.
3	 Previous displays the last page open in the PDF document. This button appears unavailable on the HMI device when on the first page of the PDF document.
4	 Next opens the next page of the PDF document. This button appears unavailable when on the last page of the PDF document on the HMI device.
5	The currently displayed Page Number of the PDF on the HMI device and the total Page Count or number of pages of the PDF document.
6	The value in the Text property of the PDF Viewer Add-On Graphic that displays the name of the PDF document on the HMI device.
7	 Fit Width expands the width of the PDF document.
8	 Fit Window expands the entire page of the PDF document to fit the PDF graphic element.
9	 Zoom In increases the PDF document magnification by the percentage specified in the ByPercent property of the Zoom In event command.
10	 Zoom Out decreases the PDF document magnification by the percentage specified in the ByPercent property of the Zoom In event command.

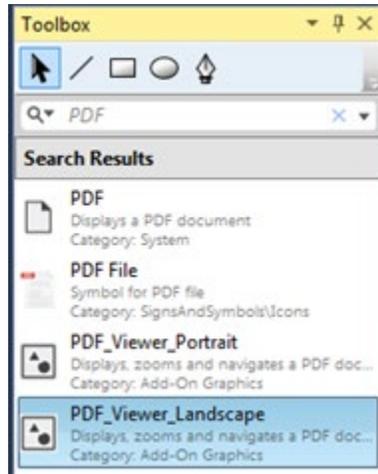
See also

[Bind bookmark properties of a PDF graphic element](#) on [page 143](#)

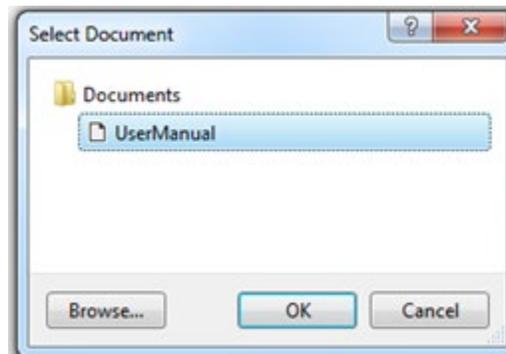
[Example 3: Create custom PDF Viewer](#) on [page 138](#)

Example 3: Create custom PDF Viewer

To create a custom PDF Viewer, change one of the pre-configured PDF Viewer Add-On Graphics or add the PDF Viewer graphic element to a screen or popup from the **Toolbox**. This example illustrates adding a PDF Viewer graphic element to a screen from the **Toolbox**. Drag and drop the PDF element on a screen or double-click the element in the toolbox.



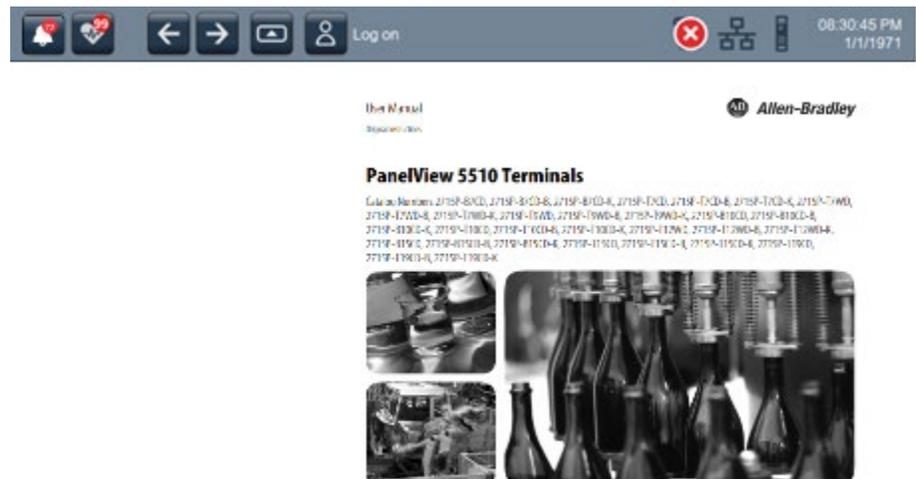
In the **Select Document** dialog box, select the PDF document to add to the screen, or select **Browse** to select a new PDF document.



The PDF graphic element appears on the screen and references the PDF document.



Next, size and position the PDF element on the screen as needed.

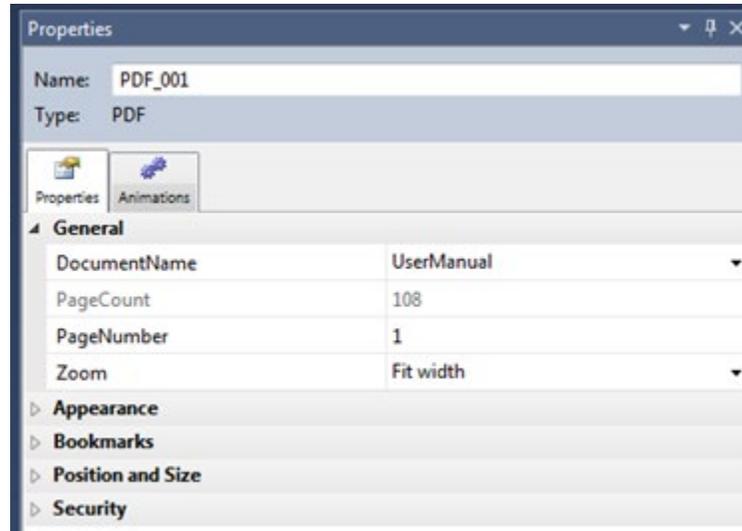


In the **Properties** window, select the **Properties** tab to configure the properties for the PDF Viewer graphic element.

In the **Properties** window, expand the **General** category and configure the required properties:

- **DocumentName.** The name of the PDF document to appear on the HMI device for the Add-On Graphic.
- **PageNumber.** Type the page number of the PDF document to initially open on the HMI device. The page also displays in View Designer.

- **Zoom.** Select the initial zoom factor for the HMI device. Also use the zoom setting in View Designer.



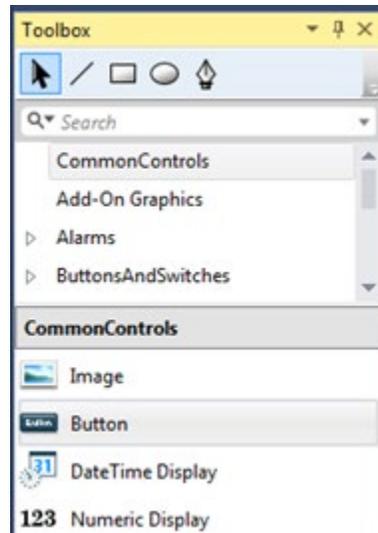
To control the properties dynamically at runtime, bind the properties to an expression or tag in a controller.

The PDF graphic element has touch support for navigating and panning pages within the referenced PDF document. To further customize a PDF graphic element, add buttons with event commands to navigate and zoom the PDF document on the HMI device. The following commands are available to use with the PDF Viewer graphic element:

- **Next Page.** Navigates to the next page in the PDF when the command executes.
- **Previous Page.** Navigates to the previous page in the PDF when the command executes.
- **Zoom In.** Increases the document magnification by a percentage of the preset level in a PDF document. In the **ByPercent** box, enter a value to increase magnification.
- **Zoom Out.** Decreases the document magnification by a percentage of the preset level in a PDF document. In the **ByPercent** box, enter a value to decrease magnification.
- **Zoom To.** Adjusts the magnification of the PDF to fit an entire page, fit the width of the page, or show the page at 100% in the PDF Viewer. In the Zoom list, select a magnification.

As an example, add a button to the screen that navigates to a specific page in the PDF document when the operator presses the button.

Add a button to the screen from the toolbox.

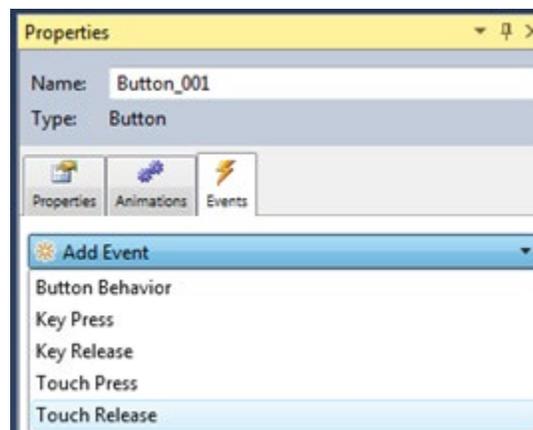


PanelView 55 10 Terminals

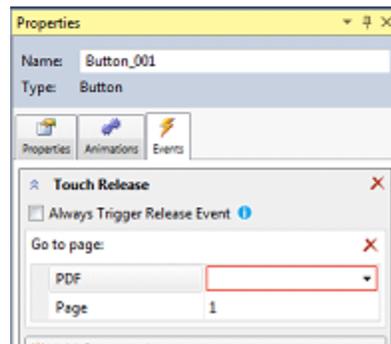
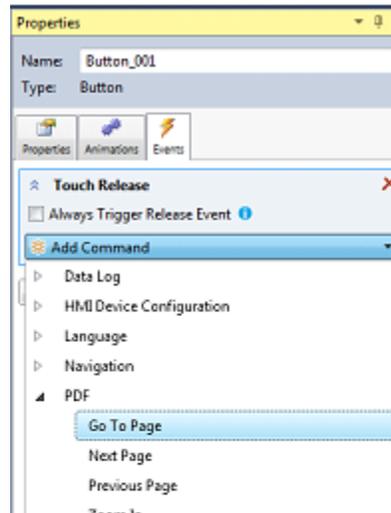
Catalog Numbers 2715-T7CD, 2715-T7CD-B, 2715-T7CA,
2715-B7CA-B, 2715-T9WD, 2715-T9WD-B, 2715-T9WA
2715-T10CA-B, 2715-B10CD, 2715-B10CD-B, 2715-B10D
2715-T12WA-B, 2715-T15CD, 2715-T15CD-B, 2715-T15C
2715-B15CA-B, 2715-T19CD, 2715-T19CD-B, 2715-T19C



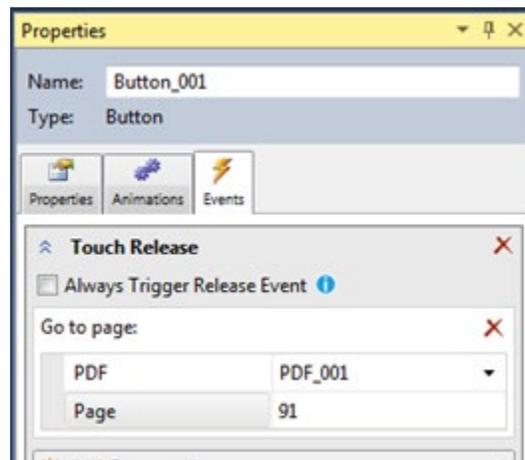
Next, access the **Events** tab in the Properties pane and add a **Touch Release** event to interact with the PDF Viewer graphic element.



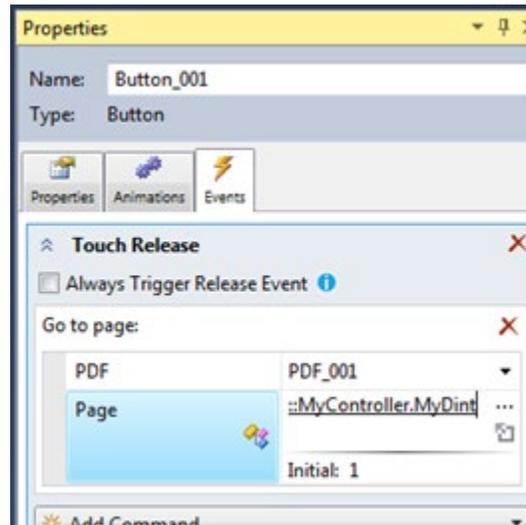
Next, add a **Go To Page** command to the event.



Select the PDF element to work with the command. Then enter the page number that opens when the command executes.



Binding the **Page** property to an expression or tag in the controller also controls the page dynamically at runtime based on the state of the machine or process.



Label the button to identify the page or section of the document that appears when pressing the button.



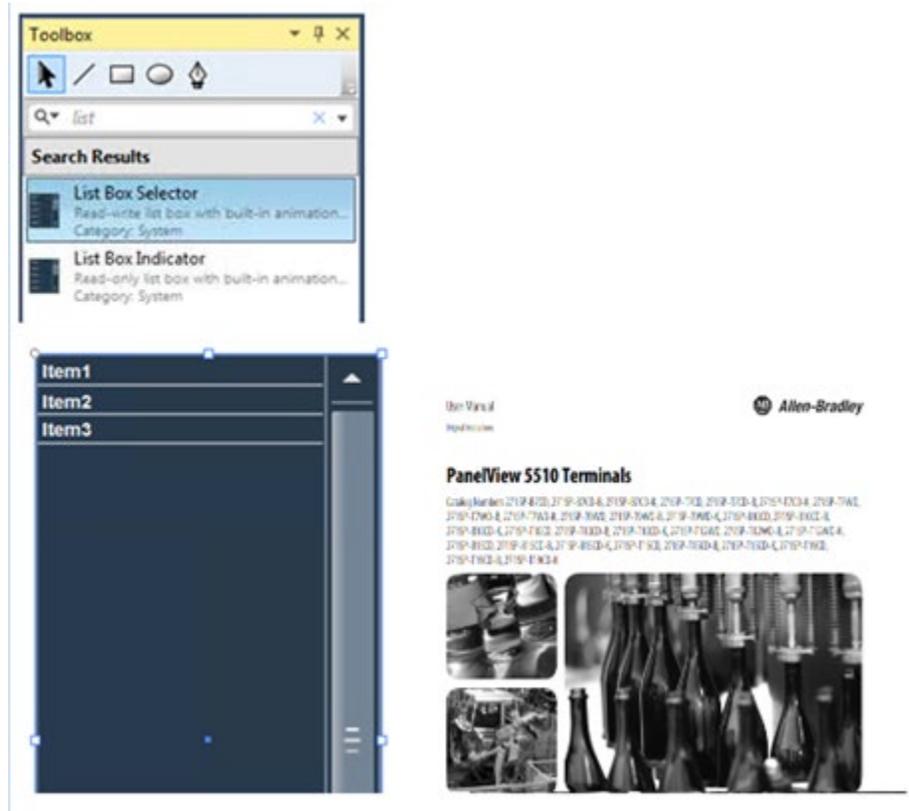
See also

[Bind bookmark properties of a PDF graphic element](#) on [page 143](#)

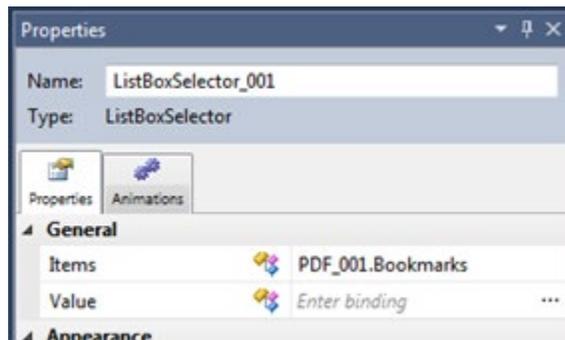
Bind bookmark properties of a PDF graphic element

Add other buttons with PDF commands to the screen to interact with the custom PDF Viewer graphic element. The PDF Viewer graphic element has bookmark properties to display and navigate bookmarks in a PDF document. The Bookmarks property displays the bookmarks in a PDF document. The **BookmarkIndex** property navigates to a specific bookmark in a PDF document.

Use a **List Box Selector** to display and navigate bookmarks in a PDF document. Drag a List Box Selector onto the screen.



Next, use property-to-property binding to bind the **Items** property of the **List Box Selector** to the **Bookmarks** property of the PDF graphic element.



Use property-to-property binding to bind the **Value** property of the **List Box Selector** to the **BookmarkIndex** property of the PDF graphic element.

At runtime, the List Box Selector displays all bookmarks in the PDF document.



At runtime, selecting a bookmark opens the bookmark in the PDF document.

See also

[PDF Viewer](#) on [page 131](#)

Guidelines for working with PDF documents

Follow these guidelines for performance when working with PDF documents:

- Each screen has a limit of displaying two PDF documents. This includes PDF Viewer Add-On Graphics and PDF graphic elements.
- PDF documents on the HMI device support internal links for navigation within the PDF document. PDF documents on the HMI device do not support external links. PDF links cannot navigate outside of the PDF document on the HMI device.
- PDF files with complex drawings take longer to render. For example, if a PDF document has an image with individual shapes such as lines, circles, and rectangles, that image takes longer to render than a drawing that is a single image.
- Do not rotate the PDF graphic element or a PDF Viewer Add-On Graphic. Rotating an element with a PDF document reduces the clarity of content in the PDF document and prevents zooming and navigating the PDF document on the HMI device.

See also

[PDF Viewer](#) on [page 131](#)

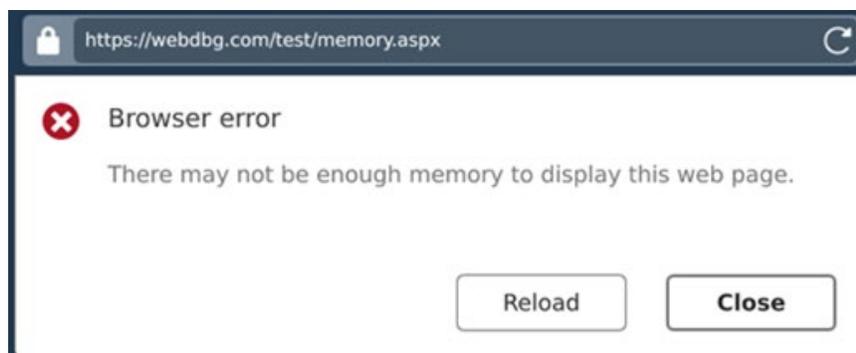
Web Browser

The web browser feature provides an HTML5-capable browser useful for displaying many types of information to operators:

- Webpages – including built-in webpages from many Rockwell Automation products
- Videos – support for mp4 video files
- IP cameras – support for H.264 or mjpeg streaming

Use the web browser to access content on the intranet, the Internet, remote file servers, devices with built-in webpages, removable SD or USB drives, or web cameras. The web browser displays this content in a window which overlays your screen similarly to a popup display. The web browser is available for the PanelView 5510 and disabled for the PanelView 5310.

If the web browser navigates to a webpage which starts consuming more memory than what is available on the HMI device, the webpage will close with an error:



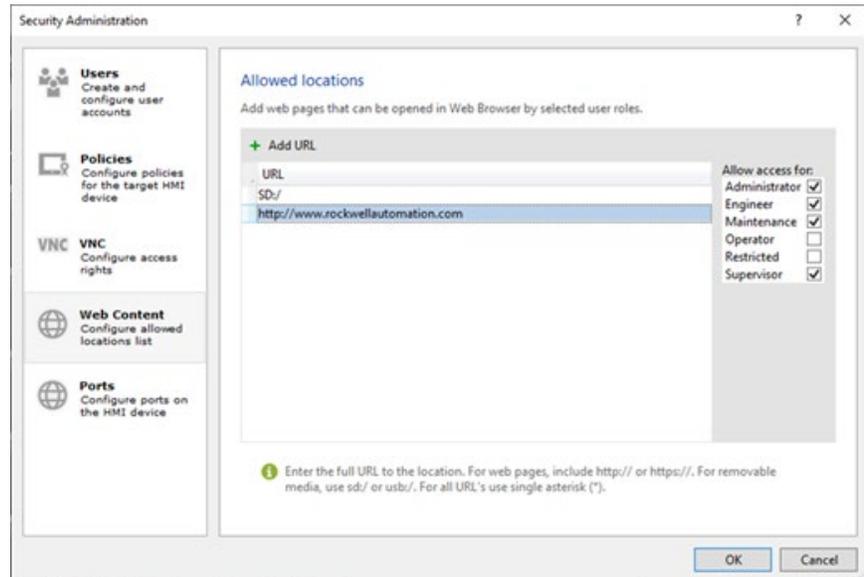
During emulation, the **Open Web Browser** command opens computer's default browser. This lets you test the URL links. The HMI Device web browser is not supported in the emulator.

The web browser does not support:

- PDF files. Use the PDF viewer instead.
- Downloads. This prevents malicious web sites from downloading to the PanelView 5000.

Web Browser Guidelines

To use the web browser, determine where you want to store the content and what user roles can access it. Use this information to fill out the **Web Content** tab for allowed locations on the **Security Administration** dialog in View Designer:



To allow access to a location, enter the location as:

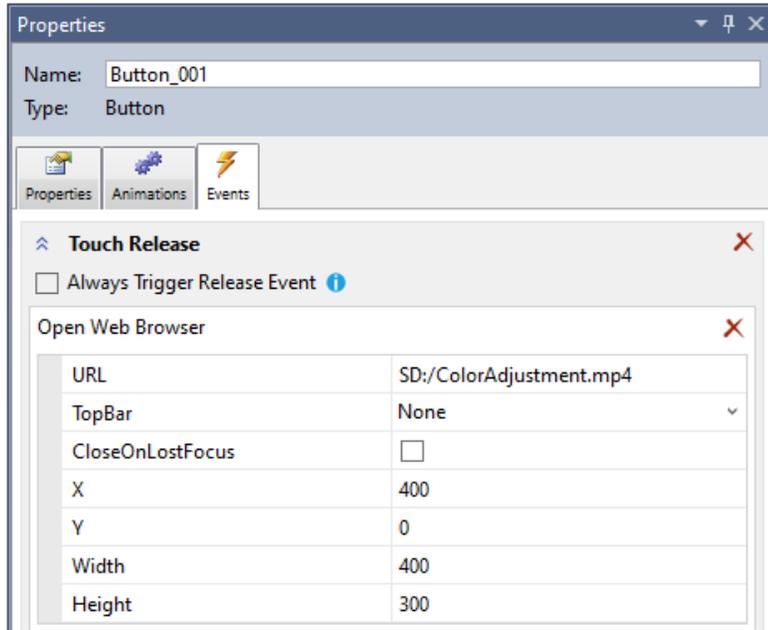
- SD:/ or USB:/ to allow access to the SD card or a USB drive on the HMI device
- An IP address in each format, for example *http://192.168.1.151*
- A remote webpage location, for example *https://www.rockwellautomation.com*

Use the asterisk * as a wildcard to substitute a character or a string of characters. Enter a single asterisk in the row to allow the security role to access all possible URLs.

For each location, select the security roles that are allowed to access that location. For example, you can have maintenance webpages that are accessible by your maintenance personnel but not by your operators.

You can also use the Ports tab on the Security Administration dialog to define any specific ports needed to communicate with equipment in your facility. Ports 80 and 443 are open by default and do not need to be added to the Ports tab. These are the ports typically used by http and https protocols.

An **Open Web Browser** command opens the web browser. You can execute this command from any type of event, such as a button press event or a project event.



Open Web Browser Command

The **Open Web Browser** command has several properties, all of which are bindable such that they could be modified at runtime using a binding to a tag or expression. The available properties are:

- **URL.** Enter the link that the browser will display. For example:
 - SD:/MyVideo.mp4 to display a video stored on the SD card
 - USB:/RestartMachineInstructions.html to display a webpage stored on the USB drive
 - 192.168.1.151 to display a webpage from a piece of equipment such as a Logix controller or IP camera
 - https://www.rockwellautomation.com to display the main Rockwell Automation webpage from the internet
- **TopBar.** Select the look and the behavior of the top bar of the web browser:
 - **None.** The address bar and navigation buttons are hidden.
 - **Slim** (read-only). Displays only the address bar.
 - **Normal** (read-only). Displays full top bar with navigation buttons: forward/back, zoom control and close. The address bar is read-only.
 - **Normal** (editable). Displays full top bar with navigation buttons: forward/back, zoom control and close. The operator can enter URLs in the address bar.
- **CloseOnLostFocus.** When selected, closes the web browser when the focus shifts to other screen elements. The browser will always close if you press the navigation button, open a popup, perform a screen navigation, or execute a **Close Web Browser** command. For example,

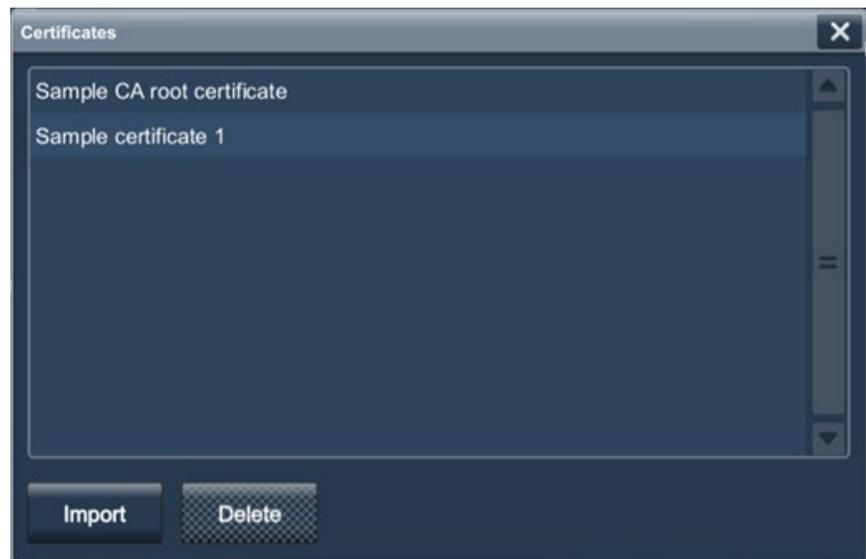
the web browser closes when you select an element outside the web browser. This property is selected by default.

- **X, Y.** Set the initial position of the top-left corner of the web browser canvas in pixels. Use 0,0 for these properties to display the browser as a full screen.
- **Width, Height.** Set the dimensions of the web browser canvas in pixels. Use 0,0 for these properties to display the browser as a full screen.

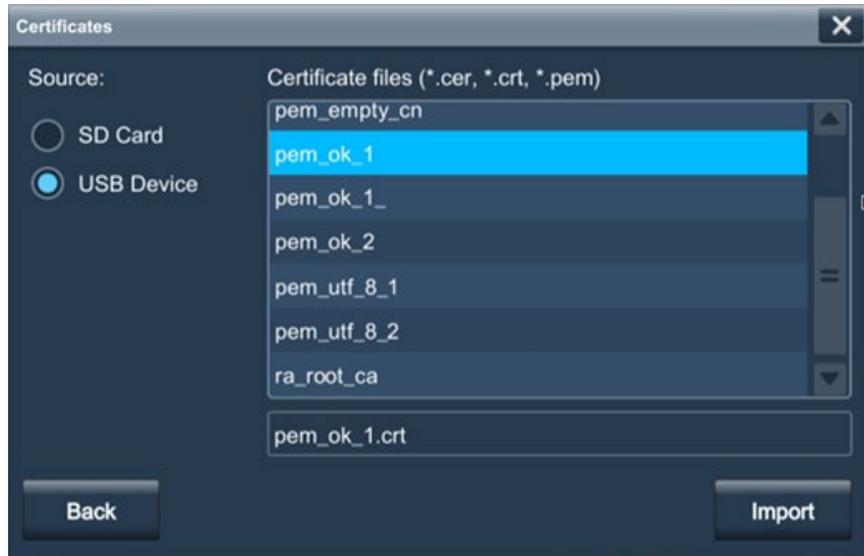
Browser Certificates

The browser certificates can be loaded at runtime into the HMI device. This is useful if the https access certificate for a website has expired. The updated certificate can be loaded without shutting down the HMI device. The HMI device comes with a set of common root certificates.

To see what certificates are currently loaded on the HMI device, go to the predefined Settings > Certificates screen. The screen will display a list of all currently loaded certificates:



To load a certificate, place a new or updated certificate in the root directory of an SD card or USB drive, then insert the SD card or USB drive into the HMI device. Navigate to the predefined Settings > Certificates screen and press the Import button. Select the certificate to load and press the **Import** button again. Certificate files must end with .cer, .crt, or .pem extensions.



Trend chart and data log overview

Log data to capture historical tag values of a process or machine. A data log is a time-series collection of values from specified tags. Export data logs from the HMI device to removable media, such as an SD card or a USB storage device. Exported data logs exist in a .zip file. The .zip file has a .csv file for each data log. Export data logs through an event that triggers a command to export the data logs to removable media.



Tips:

- Data logs can have tag values from all controllers in a project.
- Data logging requires a minimum 4 GB SD card in the PanelView 5000 HMI device.

Use a data log to:

- View historical data on a trend.
- Collect tag values at a specified sample rate.
- Find data patterns in a system.
- Troubleshoot a system.
- Audit data

A trend chart displays data over time in a graphical format. Trend charts can display values from tags in a controller, user-defined properties, or properties of a graphic element. For example, a trend chart monitoring tank pressure and tank temperature for an hour.

Use a trend chart to:

- Monitor processes in a manufacturing system
- Display data from a tag, user-defined property, or the property of a graphic element
- Display up to eight traces that plot data over time to graphically present data
- View, pan, and pause data on the HMI device

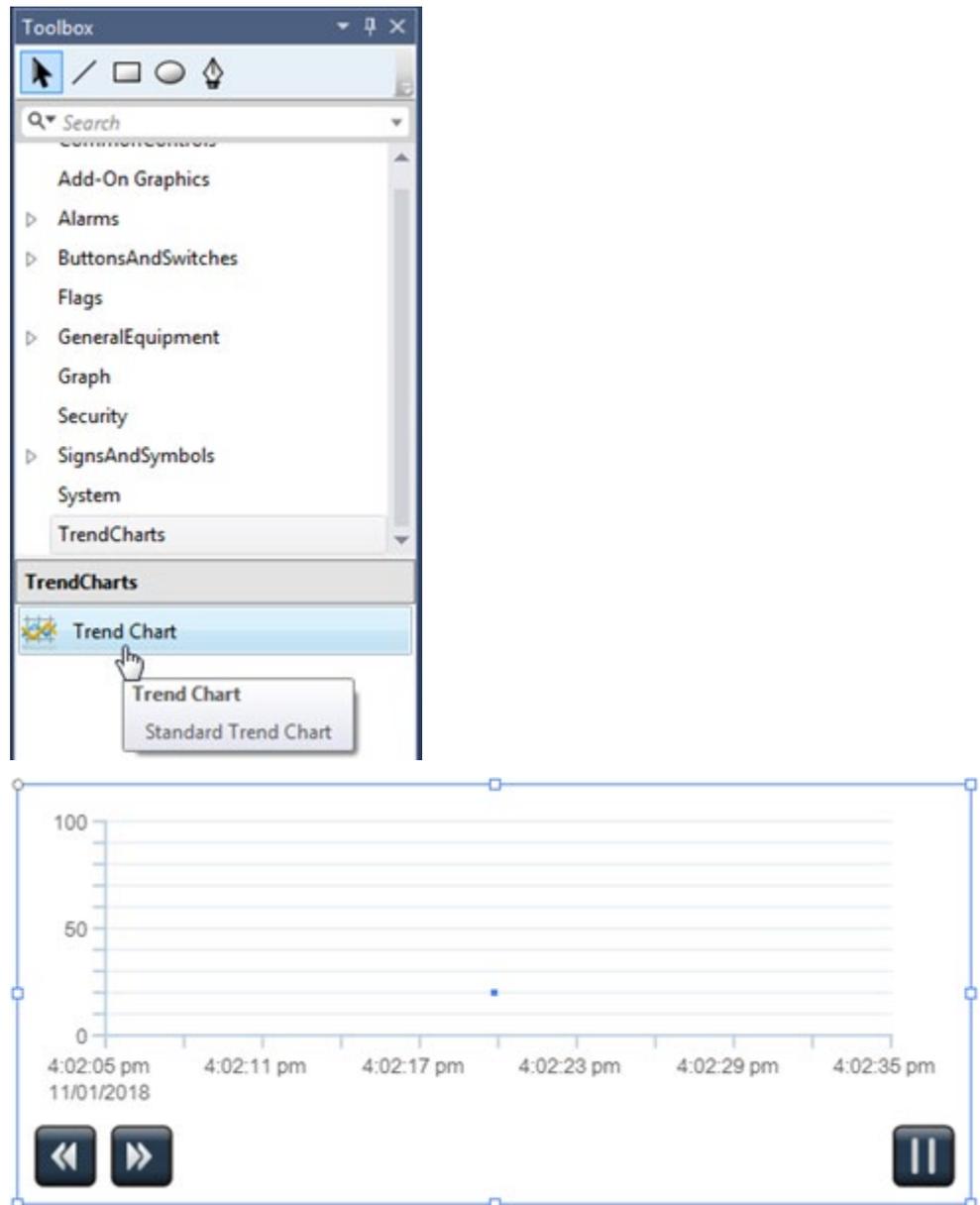
See also

[Display data on a trend chart](#) on [page 153](#)

[Log data](#) on [page 156](#)

Display data on a trend chart

View Designer includes a trend element to display how values change over time. To use the trend, in the **Toolbox**, add the **Trend Chart** element onto a screen or popup.

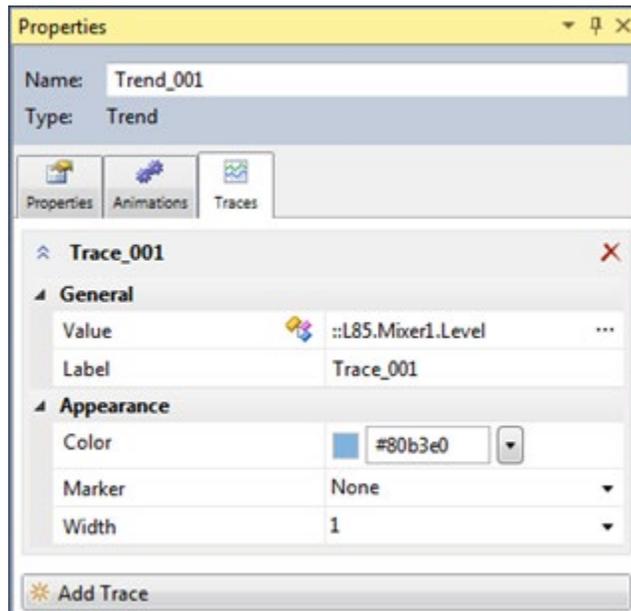


The **General** and **Appearance** properties that are unique to the Trend include:

- **MaxValue** and **MinValue**. Define the y-axis limits of the trend. View Designer supports binding the **MaxValue** and **MinValue** properties. For example, bind the **MinValue** and **MaxValue** properties to tags or expressions to dynamically change the view of the y-axis view at runtime.
- **SampleRate**. Defines how often (in milliseconds) the trend plots a new data point. The fastest rate supported is equal to the update rate chosen for the screen containing the trend. This is the rate at which the HMI device reads data from the controller.

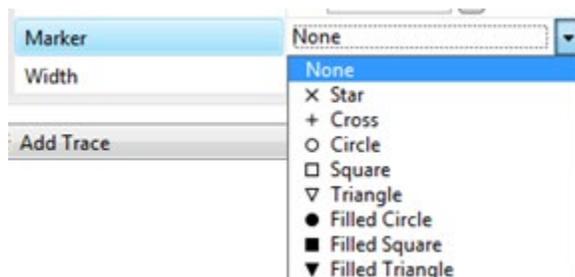
- **TimeSpan.** Defines the extent of the time axis in seconds. The maximum time span is 3600 seconds.
- **TimeSpanStart.** Defines the format of the time used by the trend. This first defines how time displays on the time axis of the trend. Also use this property to pan the trend to a specific point in time. If placing a text input on a screen and binding it to the **TimeSpanStart** property of the trend, enter a time in the format configured on the **TimeSpanStart** property. The trend pans back to that starting time.
- **ShowAxes, ShowButtons, ShowDate, ShowGridHorizontal, ShowGridVertical.** Turns visual elements of the trend on and off to gain a more complex or simpler visual representation. Turn off these properties to create a spark line style of a trend.

After configuring the **General** and **Appearance** properties, configure the traces displayed on the trend. A trace is the line or points on the trend to represent a changing tag value. Define up to eight traces for a trend. Configure traces in the **Traces** tab on the **Properties** pane.

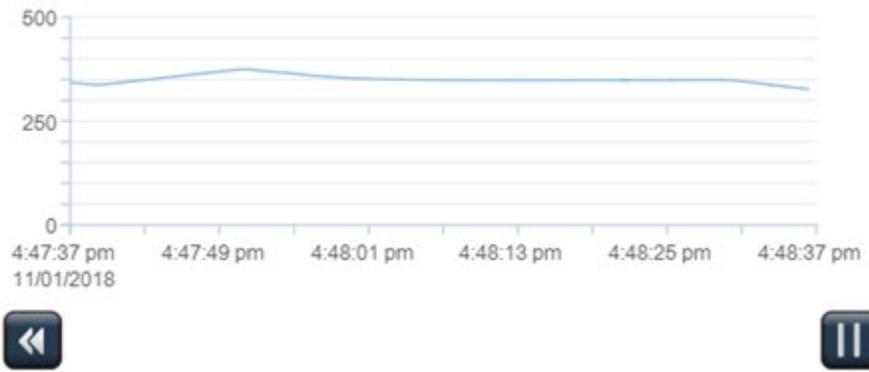


The **Trace** properties include:

- **Value.** Bind to a tag in the controller or a system tag.
- **Label.** Static text to document the meaning of the trace.
- **Marker.** Displays a marker for every data point plotted for the trace.
- **Width.** Defines the width of the trace line.



At runtime, the trend displays buttons to control viewing the trace data:



The pause button stops the trend drawing. The pan back button pauses the trend and pans back one half of the time span for each press of the button. After panning back, a pan forward button appears to pan forward.

After pausing the trend, the pause button changes to the play button. Pressing the play button returns the trend to the current time and returns to updating the trend in real time.

A trace with an unlogged Value tag shows real-time data and buffers up to 7500 samples to enable panning. Once the screen with the trend closes, the buffer erases. To view more historical data, use the data logging feature.



Tip: If real-time values are not available for a trace, the trend automatically checks the data logs to display historical data.

See also

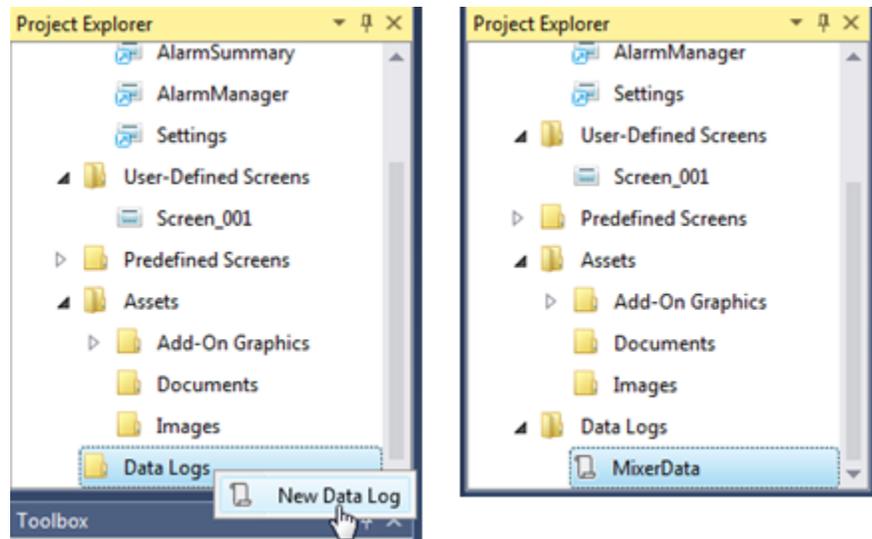
[Log data](#) on [page 156](#)

Log data

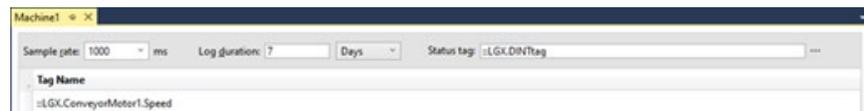
Data logging enables logging Logix tag values for longer term storage. To use data logging, insert at least a 4 gigabyte (GB) SD card into the PanelView 5000 HMI device. The PanelView 5000 HMI device supports SDHC SD cards that have up to 32 GB of storage. PanelView 5000 HMI devices log up to 250 different tags with a maximum of 500 milliseconds for up to 30 days (1.296 billion records). Fewer tags or a slower log rate increases the amount of time to log. Create up to three different data logs to log different sets of tags at different rates.

Format the SD card with FAT32 or EXT3 format. The EXT3 format is less susceptible to log file corruption from power loss on the PanelView 5000 HMI device. The EXT3 format needs a third-party utility for formatting on a PC.

To define a data log, right-click the **Data Logs** folder in the **Project Explorer** and select **New Data Log**. Create an optional descriptive name for the data log.



Select the data log to open the data log configuration.



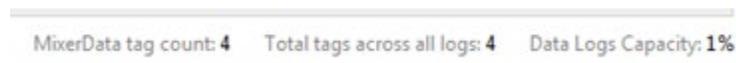
The Sample rate defines the logging rate for the tags in this data log. The Log duration stores samples for the specified time. Data logs are circular. After the Log duration, newer data starts overwriting the oldest data. Select the ellipsis button to open the tag browser to select a tag to log. Add additional rows to add more tags.

Bind the status tag to a Logix DINT tag to optionally indicate the status of the individual data log. The status code values are:

- Running: 10
- Stopped: 20
- Safely Stopped: 30 (Indicates that the SD card was shut down from the Data Export screen.)

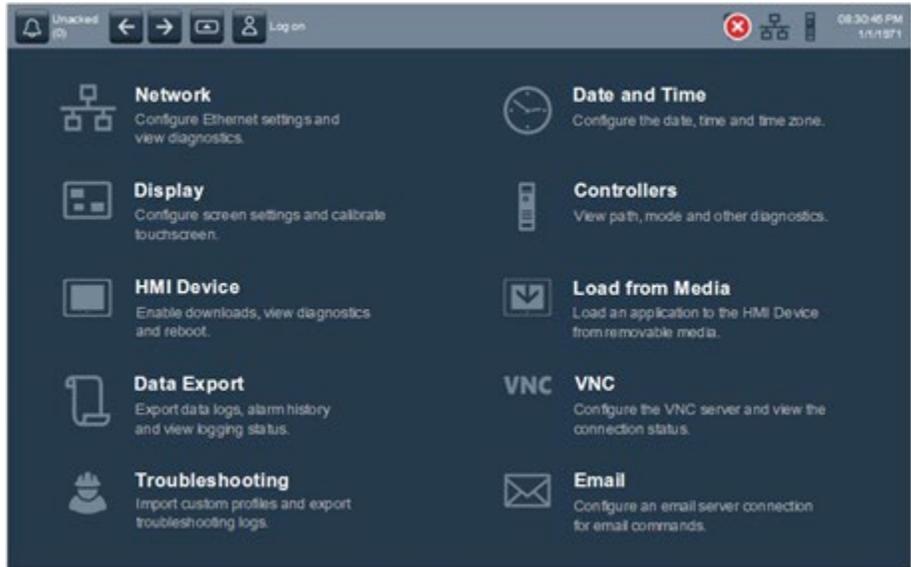
Data in the lower right of the data log configuration displays:

- The number of tags in the current log
- The number of tags across all the logs
- The percentage of the total number of allowable records used across all the logs.

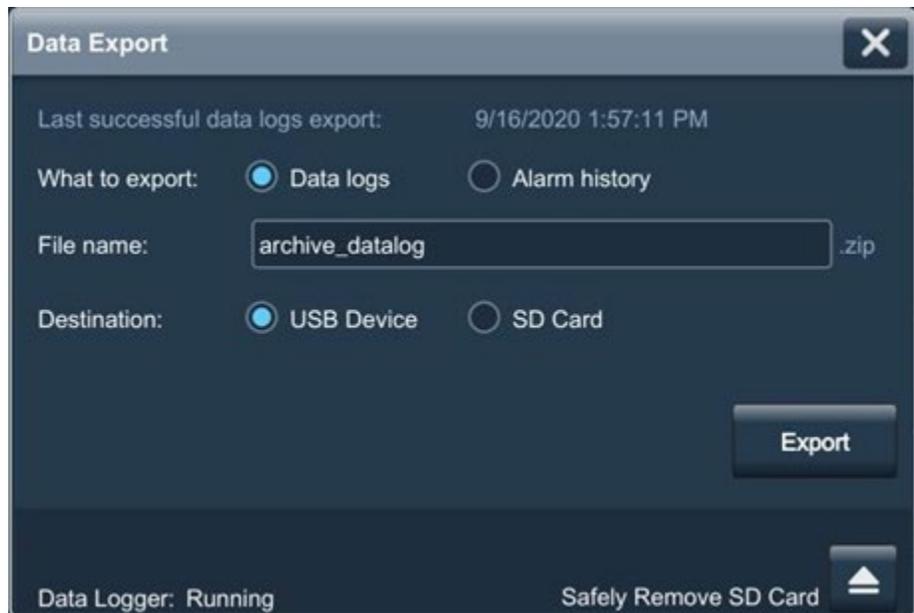


Inserting an SD card in an HMI device automatically logs data after downloading the project to the HMI device. Removing the SD card stops data logging. Data logging restarts automatically after inserting an SD card.

Export data logs to the SD card or a USB drive to perform external analysis of the logged data. The data logs export to a zip file. The zip file has a .csv file for each data log defined for the project. To export, from the **Settings** screen, select **Data Export**.



On the **Data Export** screen, enter the name of the exported zip file and specify SD card or USB drive. Large data logs display progress while exporting. After the zip file exports, the **Last successful export** update appears with the date and time of the latest export.



Remove the media and insert it into a computer to view the data log contents. Extract the file to view the .csv files for the individual data logs. The .csv files use the same name as the name of the data log.

Opening the .csv file in Excel displays the log values. Each tag logged appears in a different column:

	A	B	C	D	E
1	# Data Log Export				
2	# HMI Device: 2715P-T12WD				
3	# HMI IP Address: 10.88.32.204				
4	# Project Name: TestV5				
5	# Data Collection: MixerData				
6	# Export Date/Time: 2018-11-26 20:40:09.855 UTC				
7	Timestamp	2715P-T12WD.L85.Mixer1.Level	2715P-T12WD.L85.Mixer1.LevelSP	2715P-T12WD.L85.Mixer2.Level	2715P-T12WD.L85.Mixer2.LevelSP
8	50:29.0	250	250	403.2000122	403.2000122
9	50:29.5	250.0000153	250	403.1999817	403.2000122
10	50:30.0	249.9999695	250	403.2000427	403.2000122
11	50:30.5	249.9999847	250	403.2000122	403.2000122
12	50:31.0	249.9999195	250	403.1999817	403.2000122

To view the timestamp in a different format, select the timestamp cells in Column A and format the cells in Excel to define the format. Select "m/d/yyyy h:mm:ss.000" to see time in milliseconds or "m/d/yyyy h:mm:ss" to see the time in seconds.

	A	
1	# Data Log Export	
2	# HMI Device: 2715P-T12WD	
3	# HMI IP Address: 10.88.32.204	
4	# Project Name: TestV5	
5	# Data Collection: MixerData	
6	# Export Date/Time: 2018-11-26 20:40:09.855 UTC	
7	Timestamp	2715P-T12W
8	11/26/2018 19:50:28.966	
9	11/26/2018 19:50:29.480	
10	11/26/2018 19:50:30.055	

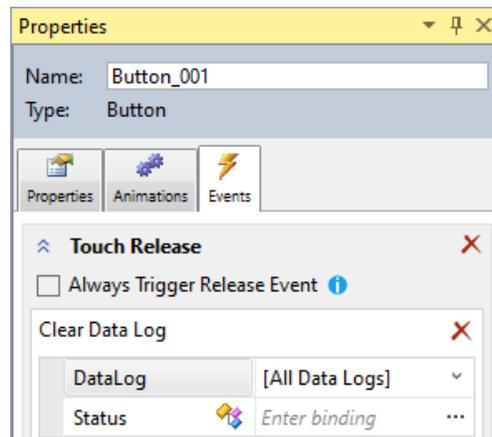
See also

[Trend chart and data log overview](#) on [page 153](#)

Clearing a data log

To make sure the data log only contains new values from a batch or machine run, clear the data log at the start of the batch or machine run. Export the data log at the end of the batch or machine run to have a specific record of logged values obtained during the batch or machine run.

To clear a data log, use the Clear Data Log command. Execute this command from any event such as a button touch event or a project event:



Use the DataLog property to select which log to clear. Select either All Data Logs to clear all data logs configured in the HMI device, or select a specific data log to clear.

Bind the Status property to a DINT Logix tag to optionally get a status code indicating the status of the Clear Data Log command. This can be used to indicate when the clear is complete since it may take some time for a large data log to clear. The status code values are:

- Data log clearing in progress: 10
- Data log successfully cleared: 20
- Unable to clear data log. Data log is not configured: 30
- Unable to clear data log. Data log is not running: 40
- Selected data log does not exist: 50
- Clear data log command is already in progress: 60
- Unable to clear data log. Export is in progress: 70
- Unable to clear data log. Insert SD card: 80
- Unable to restart the cleared data log. Remove and replace the SD card:
90
- Unable to clear the data log. Try again or restart terminal: 100

Load from Media

Use **Load from Media** to load a project from an SD card or USB drive to a PanelView 5000 HMI device without using View Designer to download. For example, send an updated project to a user of a machine that does not have View Designer available to download the project.

Load from Media also supports updating the firmware on the PanelView 5000 if the project requires a different firmware than the current version on the PanelView 5000.

See also

[Load a project from media](#) on [page 161](#)

Load a project from media

Load a project from an SD card or USB drive in order to load the project to a PanelView 5000 HMI device without using View Designer. First save the project as a .vpdr file.

To load a project from media

1. Select **File > Save Project as**. The **Save Project As** dialog box opens.
2. Under **Save as type**, select **View Designer Runtime Application (.vpdr)** and select **Save**. The **Save Runtime Application** dialog box opens.
 **Tip:** The .vpdr file contains all of the project content to load the file from an SD card or USB drive.
3. Select **Save**.
4. Copy the .vpdr file into the root directory of an SD card or USB drive and insert the media device into the PanelView 5000 HMI device.



Tips:

- To allow **Load from Media** to install a new revision of firmware on the PanelView 5000, copy the firmware .dmk file to the root directory of the SD card or USB drive. The firmware major revision must match the major revision of View Designer used to create the .vpdr project.

Firmware .dmk files are usually a part of the Studio 5000 install located at:

C:\Users\Public\Documents\Rockwell Automation\Firmware Kits

- Version 6 and later versions of firmware support loading the firmware from media. For earlier versions, load the firmware using ControlFLASH.
- Only the PanelView 5000 HMI devices manufactured after September 13, 2019 support updating firmware from the base version 1 firmware using the **Load from Media** feature. For older PanelView 5000 HMI devices, update the version 1 firmware using ControlFLASH.
- Update to a newer minor revision of firmware by copying the appropriate .dmk file to the SD card or USB drive and load the project. **Load from Media** automatically updates the PanelView 5000 and loads the project.
- Loading multiple firmware .dmk files on removable media if there are multiple .vpdr files using different versions of View Designer is supported.

5. On the **Settings** screen, select **Load from Media**. The **Load Application** screen opens.
6. Select the media type containing the .vpdr file, select the .vpdr file to load, and select **Next**.



Tip: Removable media can contain multiple .vpdr files in order to load projects for multiple PanelView 5000 HMI devices from the same SD card or USB drive.

7. Select the HMI to controller paths currently in use on the PanelView 5000 or use the HMI to controller path defined in the .vpdr project that is loading.



Tip: Use the default **Keep paths currently in use on HMI device** when the same project on multiple HMI devices communicate with different Logix controllers.

8. Select **Load** to start loading the project from the removable media. If a firmware update is required to use the .vpdr file, and the firmware .dmk file is on the removable media, the firmware update occurs and the project downloads. The PanelView 5000 HMI device then runs the new project.
9. Remove the media device containing the .vpdr file.

See also

[Load from Media](#) on [page 161](#)

VNC

Use the VNC server in the PanelView 5000 terminals to remotely monitor the terminal with a third-party VNC client. Use password configuration in View Designer and runtime access limiting on the PanelView 5000 HMI device to limit access rights for VNC clients. Access right settings include no access, view only access, or full access.

See also

[Configure VNC access in View Designer](#) on [page 163](#)

[VNC popup](#) on [page 163](#)

Configure VNC access in View Designer

Specify access rights by entering a password for view-only or full-control access. Users gain access rights when remotely connecting to an HMI device by entering the view-only or full-control password.

To configure VNC access

1. Select **Tools > Security Administration** and then select the **VNC** tab.
2. On the **Security Administration** dialog box, enter passwords for View-only and Full-control access.
 - Leaving an access password blank disables the type of access.
 - Leaving both passwords blank disables VNC access.
 - Passwords must be between six and eight characters in length.
3. Load the project to the PanelView 5000 HMI device.
4. Navigate to **Settings** and select **VNC** to enable the VNC server on the PanelView 5000 HMI device.
5. On the **VNC** popup, select the level of access or disable access.

See also

[VNC popup](#) on [page 163](#)

VNC popup

Access the **VNC** popup on the PanelView 5000 HMI device to select the level of user access or disable access. The access settings work with the passwords set in View Designer to enable restrictive user access. For example, if view-only access on the VNC popup is selected, and a user enters the full-control password on a VNC client, the user receives view-only access. This limits user

access when performing sensitive operations on the PanelView5000. After rebooting the PanelView 5000 HMI device, the VNC server configuration automatically resets to the **Disabled** state.

Use the `::Local:HMIDevice.VNCServer.ActiveConnections` tag for monitoring connection status. The **VNC** popup displays a Remote VNC active connection status message for a connected VNC client.

When launching the VNC client on a remote computer, connect to the IP address of the PanelView 5000 HMI device. An enter password message appears. Enter the view-only or full-control password to obtain that type of access, unless limited by the VNC setting on the PanelView 5000 HMI device.

See also

[VNC on page 163](#)

[Configure VNC access in View Designer on page 163](#)

Email notification

Send email notifications from the PanelView 5000 to remote operators. Do this when process or machine conditions change. For example, notify maintenance team members when a machine shuts down due to a fault. The PanelView 5000 sends emails using common email services such as Gmail or Yahoo Mail.

To use **Email Notification**:

- Configure the connection to the email server.
- Add a **Send Email** command to an event.

See also

[Load a project from media](#) on [page 161](#)

Configure the connection to the email server

Configure the connection to the email server at runtime using the **Settings** screen.

The configuration on the HMI device for the email server connection persists after rebooting the HMI device, cycling power to the HMI device, and downloading a runtime application file to the HMI device.

To configure the connection to the email server

1. Select **Navigation**  and select **Settings**.
2. Select **Email**. The **Email** popup opens.
3. Configure the settings to connect to the email server:
 - **Server Name**. Enter the name of the third-party email server. For example: smtp.gmail.com or smtp.mail.yahoo.com
 - **Authentication**. Select **Password** to require a password to connect to the email server. Selecting **None** requires no authentication to connect to the email server.
 - **User Name**. If authentication is required, enter the email address of user sending the email.
 - **Password**. If authentication is required, enter the password to connect to the email server.
 - **Sender**. Enter the name of the sender to appear in the emails.
 - **Use SSL**. Select to use secure communication to the email server.

- **Server Port.** Enter the port number between 1 - 65535 of the email server.



Tips:

- Not using SSL results in the email notification service supporting STARTTLS only.
- Ports 22, 53, 80, 2222, 27053, and 44818 are reserved for other services on the HMI device. Entering a Server Port that is out of range or reserved displays an error.
- To create a secure connection with a server (TLS/STARTTLS) either:
 - Select **SSL** and set a port with an available SSL connection. For Gmail and Yahoo Mail, in **Server Port**, enter **465**.
 - Clear **SSL** and set a port with the server that has a TLS/STARTTLS option. For Gmail and Yahoo Mail, in **Server Port**, enter **587** or **25**. If TLS/STARTTLS is not available, the connection is not secure.
- The HMI device connects to the email server and sends email notifications even when the SMTP client requires password authentication and the email server does not require password authentication.
- The PanelView 5000 supports only simple password authentication. Configure the email service for less secure access to allow the PanelView 5000 to sign in. For example:
 - Google Gmail. Log in to Google Gmail and select **Less secure app access**.
 - Yahoo Mail. Log in to Yahoo Mail and select **Allow apps that use less secure sign in**.

Send email command

After setting up the connection to the email server, send emails by adding a **Send Email** command to an event. Bind the properties of a Send Email command to tags or expressions to change the action that occurs at runtime.

Prerequisites

- Create an event to automatically trigger the command:
 - **Touch Press.** Occurs when touching the item on the screen that is configured with the event.
 - **Touch Release.** Occurs when releasing the item on the screen that is configured with the event.
 - **Key Press.** Occurs when pressing the specified key.
 - **Key Release.** Occurs when releasing the specified key.
 - **Project Event.** Occurs when the ExecuteWhen expression evaluates from **False** to **True** and the event is enabled.
 - **State Enter.** Occurs when a graphic element, screen, popup, or the System Banner transitions into the selected state through an animation.
 - **State Exit.** Occurs when a graphic element, screen, popup, or the System Banner transitions out of the selected state through an animation.

IMPORTANT To send an email when a State Enter or State Exit event triggers, configure the graphic element, screen, popup, or System Banner for **State Table** animation.

To add a Send Email command to an event

1. Select the graphic element configured with the touch or key event, or select a **Project Event**.
2. In the **Properties** pane on the **Events** tab, click **Add Command**.
3. Expand the **Notification** command category and select **Send Email**.
4. Configure the properties on the event card:
 - **Recipient.** Enter the email address of the recipient. Separate multiple recipients with commas. Enter a static value for this property or bind to an expression. Binding to an expression using string tags can change the list of recipients at runtime.
 - **Subject.** Enter the subject to be used by the email. Enter a static value for this property or bind to an expression.
 - **Message.** Enter the message the email is to send. Enter a static value for this property or bind to an expression. Binding to an expression can include process or machine values along with the message.
 - **Status** (optional). This property does not support binding to properties of other elements on a screen. Bind this property to an integer tag in the controller to display the status code for sending the email:
 - **Successfully sent: 10.** This code indicates the email is successfully sent to all recipients.
 - **Error returned from email server: 20.** This code indicates multiple errors such as a timeout while sending the email, an incorrect email address, or a communication error with the email server.
 - **Unable to reach email server: 30.** This code indicates multiple errors such as an invalid password, invalid user name, or invalid server name.
 - **Sending is in progress (email added to queue): 40.** This code indicates the email is being sent.
 - **Email request rejected: 50.** This code indicates the maximum buffer size of 120 email requests are in queue. Decrease the number of email commands that execute simultaneously.

Emails as texts

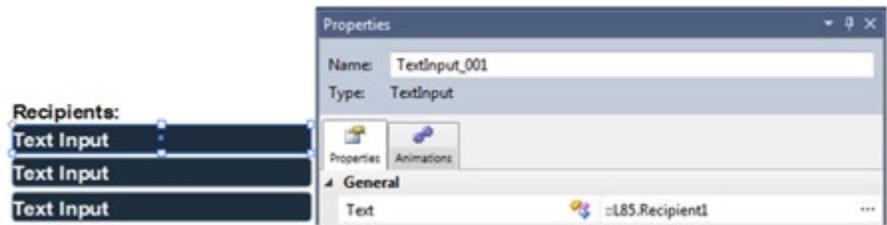
Many cell phone providers support sending emails as texts. This helps remote personnel receive notifications about machine or process conditions. For example, to send an email as a text to the Verizon cell phone number of 555-123-4567, configure 5551234567@vtext.com as the email recipient. Verify the correct address format with the cell phone provider.

Example: Configure a list of email recipients online and send them emails

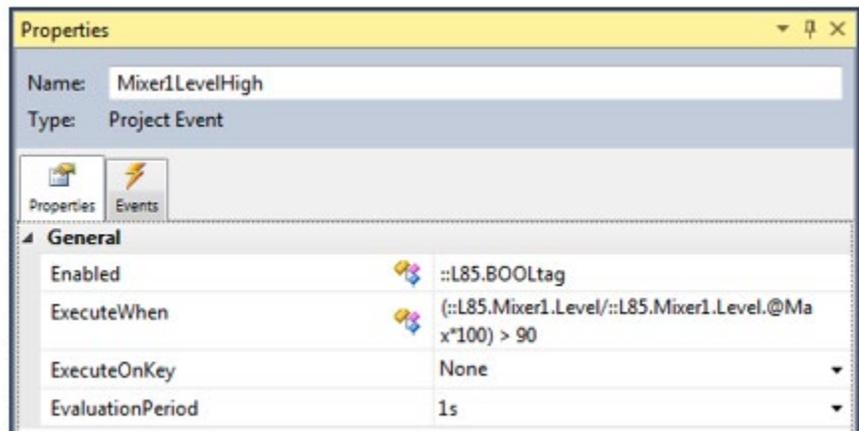
A machine OEM wants to configure a list of recipients at runtime to receive email notifications about the machine status instead of hard coding the recipients into the email commands.

To configure the recipients from the PanelView 5000, create an email

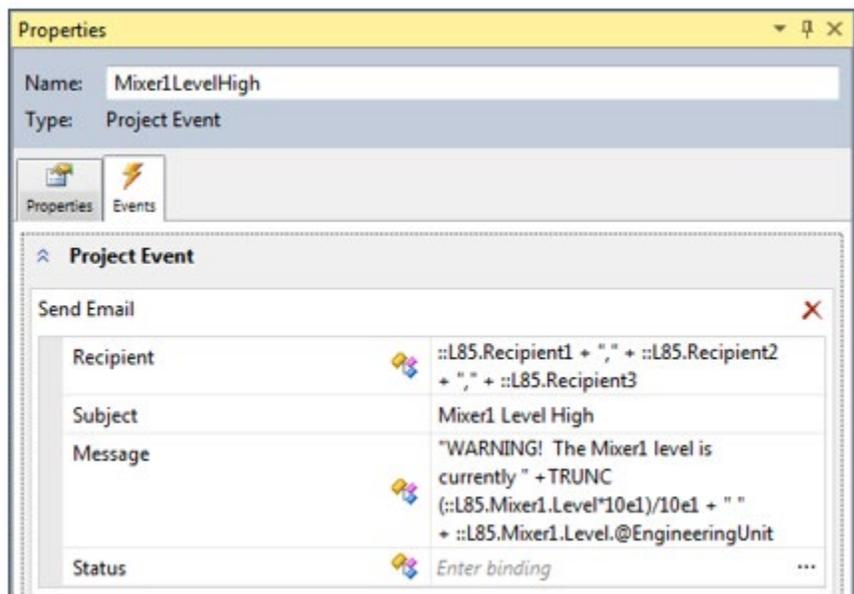
configuration screen where the user enters recipients. On the screen, use **Text Inputs** to write the recipients to string tags in the Logix controller. The string tags use **::L85.Recipient1**, **::L85.Recipient2**, and **::L85.Recipient3** to store the recipients.



This example uses a **Project Event** that sends an email when the **Mixer1** level is greater than 90%:



Add a **Send Email** command on the **Events** tab of the **Mixer1LevelHigh** project event:



At runtime, enter the email addresses of the recipients in the text inputs on the custom email configuration screen. The addresses in the **Recipient** property of the **Send Email** command create the list of recipients using the "+" operator and commas. The **Message** property of the **Send Email** command reads values from the Logix controller to supply additional information to email recipients.

Index

A

Add-On Graphics 89

- create 89
- custom properties 91
 - use 91
- instance 93
 - create 93
- Toolbox 93
- use 89

Alarms 97

- add 97
- Alarm Manager 98, 100
- Alarm Summary 98, 100
- Alarm Summary Medium 100
- AlarmManager screen 100
- AlarmSummary screen 100
 - properties 100
 - table 100
- configure 97
- Details pane 98
- guidelines 98
- Logix Designer 97
- overview 97
- PanelView 5000 HMI device 97
- Predefined Screens folder 100
- Project Explorer 100

Application tab 13

C

color table 41, 43

Controller reference 11

- add 11

D

data 150

- log 150

Data 147

- log 147

E

event types 47

- Button Behavior 47
- Key Press 47
- Key Release 47
- State Enter 47
- State Exit 47
- Touch Press 47
- Touch Release 47

events 47

- Use button behaviors 47

Expression 30

- examples 31

Expression Editor 30

G

graphic element 23

- bind 23, 30

I

images 113

- add 113
- Button Behavior 114
- change 115
- configure 114
- momentary button 114
- use 113

L

Language Switching

- configure 119
- Export Languages 119
- Import Languages 119
- Language 119, 121
- Switch Language 121
- use 119

Load from Media 155

Location field 13

N

Navigation menu 68

- properties 69
- setup 68
- shortcuts 68

P**path 13**

- add 11

PDF Viewer 125

- commands 132

- create 132

- example 132

- overview 125

- PDF document 129

 - example 129

 - guidelines 139

 - work 139

- PDF graphic element 125

 - bookmark properties 137

 - bind 137

 - properties 137

- PDF Viewer Add-On Graphic 125

 - add 127

 - example 127

 - PDF_Viewer_Landscape 127

 - PDF_Viewer_Portrait 125

 - Select PDF Viewer dialog box 129

Project Explorer 113

- Images folder 113

Project Properties 121**Projects 11**

- add images 113

- download 13

- upload 14

- Use 11

property 23

- bind 23, 28

R**runtime 19, 67****S****Screens 73**

- create 73

- reuse 73

Settings screen 19

- elements 19

- Network 19

state table 41**System Banner 18**

- contents 18

- modify 18

T**Trend 147**

- display data 147

- properties 147

 - Appearance 147

 - General 147

- trace 147

 - properties 147

- use 147

V**VNC 157**

- access 157

- dialog box 157

- use 157

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.

Allen-Bradley, expanding human possibility, Logix, Rockwell Automation, and Rockwell Software are trademarks of Rockwell Automation, Inc.

EtherNet/IP is a trademark of ODVA, Inc.

Trademarks not belonging to Rockwell Automation are property of their respective companies.

Rockwell Otomasyon Ticaret A.Ş. Kar Plaza İş Merkezi E Blok Kat:6 34752, İçerenköy, İstanbul, Tel: +90 (216) 5698400 EEE Yönetmeliğine Uygundur

Connect with us.    

rockwellautomation.com ————— expanding **human possibility**™

AMERICAS: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444

EUROPE/MIDDLE EAST/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

ASIA PACIFIC: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846