Allen-Bradley

Studio 5000 View Designer Getting Results Guide

Allen-Bradley • Rockwell Software

Rockwell Automation

Important user information

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

Activities including installation, adjustments, putting into service, use, assembly, disassembly, and maintenance are required to be carried out by suitably trained personnel in accordance with applicable code of practice. If this equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

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).

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
System Banner contents on page 20	Added General Diagnostics and Network Diagnostics.
Use the settings screen at runtime on page 21	Added all predefined screens and popups support security settings for assigning user roles that can view and change screens and interact with screens at runtime.
Example 4: Use key press and key release events on page 58	New topic
Project Events on page 61	New topic
Create a project event on page 61	New topic
Example: Write the current user name or screen name to a controller using a project event on page 65	New topic
Alarm Summary and Alarm Manager on page 100	New topic
Log data on page 137	New topic
Load from Media on page 141	New topic
Load a project from media on page 141	New topic
Email notification on page 145	New topic

Chapter 6

Project Events

Chapter 7

Navigation menu

Chapter 8

Reuse screens

Chapter 9

Add-On Graphics

Chapter 10

Alarms

Chapter 11

Use images

Chapter 12

Language switching

Chapter 13

PDF Viewer

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.

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.
<PV First> 5300 Terminals User Manual publication 2713P-UM001	Describes how to install, configure, operate, and troubleshoot the PanelView 5300 terminals.
PanelView 5500 Terminals User Manual publication 2715-UM001	Describes how to install, configure, operate, and troubleshoot the PanelView 5500 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.

Legal Notices

Copyright notice

This document and any accompanying Rockwell Software products are copyrighted by Rockwell Automation Technologies, Inc. Any reproduction and/or distribution without prior written consent from Rockwell Automation Technologies, Inc. is strictly prohibited. Please refer to the license agreement for details.

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.

Other Licenses

The software included in this product contains copyrighted software that is licensed under one or more open source licenses. Copies of those licenses are included with the software. Corresponding Source code for open source packages included in this product can be located at their respective web site(s).

You may alternately obtain complete Corresponding Source code by contacting Rockwell Automation via our 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.

A full list of all open source software used in this product and their corresponding licenses can be found in the OPENSOURCE folder included with these Release Notes. The default installed location of these licenses is C:\Program Files (x86)\Common Files\Rockwell\Help\Studio 5000 View Designer Help\V6\ReleaseNotes\OPENSOURCE\index.htm.

Trademark Notices

Allen-Bradley, Rockwell Automation, and Rockwell Software are trademarks of Rockwell Automation, Inc.

Any Rockwell Automation software or hardware not mentioned here is also a trademark, registered or otherwise, of Rockwell Automation, Inc.

Other Trademarks

All other trademarks are the property of their respective holders and are hereby acknowledged.

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

Warranty

This product is warranted in accordance with the product license. The product's performance may be affected by system configuration, the application being performed, operator control, maintenance, and other related factors. Rockwell Automation is not responsible for these intervening factors. The instructions in this document do not cover all the details or variations in the equipment, procedure, or process described, nor do they provide directions for meeting every possible contingency during installation, operation, or maintenance. This product's implementation may vary among users.

This document is current as of the time of release of the product; however, the accompanying software may have changed since the release. Rockwell Automation, Inc. reserves the right to change any information contained in this document or the software at any time without prior notice. It is your responsibility to obtain the most current information available from Rockwell when installing or using this product.

Contact Rockwell Automation

Customer Support Telephone — 1.888.382.1583

Online Support — http://www.rockwellautomation.com/support/

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 13

Configure a View Designer path to PanelView 5000 on page 16

Download a project on page 16

Upload a project on page 17

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.

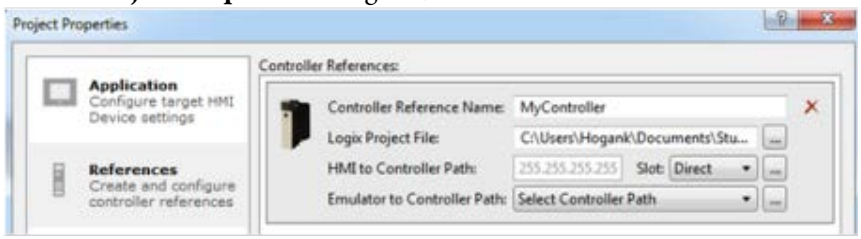
Adding a controller reference adds the controller reference to the next available reference card on the **References** tab. Each controller reference has a number in the reference card. This controller number appears when browsing for HMI device controller tags in the **Tag Browser**.

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 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 RSLinx Classic 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. This enables testing projects on a different test controller and avoids 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

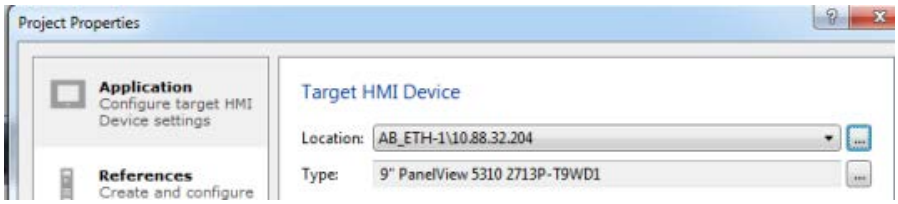
Configure a View Designer path to PanelView 5000 on page 16

Download a project on page 16

Upload a project on page 17

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 16

Upload a project on page 17

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 17

Upload a project

Upload a project from an HMI device. This is useful when the .vpd file for the project is not 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.

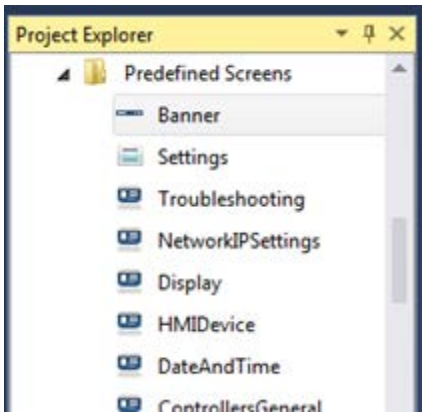
See also

Project Administration on page 13

Predefined Screens and the system banner

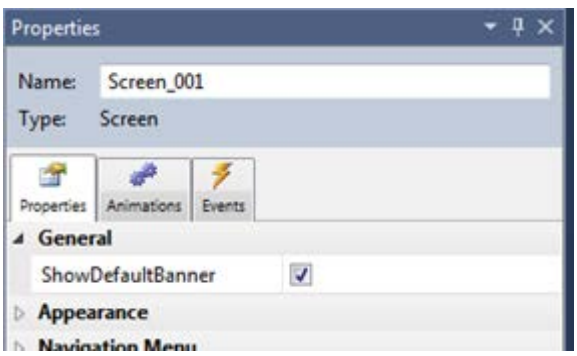
View Designer includes a set of predefined screens and a banner in the **Predefined Screens** folder in **Project Explorer**. Use predefined screens and the system banner to configure and view the status of an application. Also change content in the **Predefined Screens** folder.

Tips:
- View Designer does not allow renaming or deleting content or adding screens or popups to the **Project Explorer** 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.



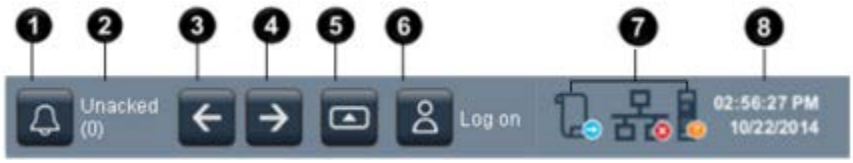
See also

System Banner contents on page 20

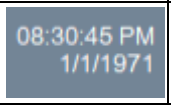
Use the system banner on page 21

System Banner contents

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



Item	Description	
1		**Alarm Status Indicator** opens the **Alarm Summary** screen on the HMI device. The indicator turns red and flashes when active alarms are unacknowledged. The indicator turns solid red when all active alarms are acknowledged. The indicator returns to the default state with a gray icon when all alarms return to normal, even if alarms are unacknowledged.
2	The **Unacked** number displays the number of unacknowledged alarms in the project, regardless of the inhibit state of the alarms.	
3		**Previous** opens the previous screen in the navigation history on the HMI device.
4		**Next** opens the next screen in the navigation history on the HMI device. To navigate with **Next** requires displaying a previous screen first.
5		**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.
6		**Log On** logs on or off the project. The name of the user appears in the **System Banner**. Screens display according to the corresponding user role.
7		**General Diagnostics Status.** These icons appear depending on the status of data logging and project events. •  **Data Log export.** The data log is exporting. •  **Data log error.** The data log is in error while collecting data or exporting. •  **Data log 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 log warning**. Tags in the data log are in error. Select the icon to view the error details. •  **Project Event Status Error**. A project event is in error. Select the icon to view the error details. •  **Error**. Project events and data logs are in error. Select the icon to view the error details.

		**Network Status.** These icons appears only if there is an issue with network communication: •  **Network caution.** The network is functioning but has an issue that requires attention. For example, the Device Level Ring may have a break. Select the icon to open **HMI Device Configuration Network** for details about the issue. •  **Network error.** The network is not functioning. An Ethernet link is disconnected, or there is a duplicate IP address.
•		**Controller Status.** These icons appears only if there is an issue with any controller in the project: •  **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.
8		The current time and date of the HMI device.

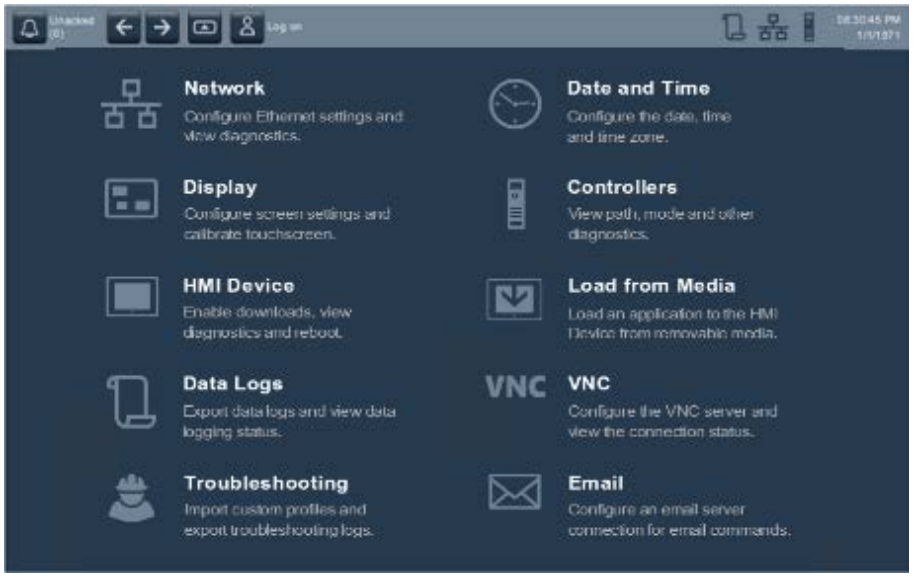
To open and change the System Banner, in the **Predefined Screens** folder, double-click **System Banner**. Changing the System Banner includes moving or deleting existing elements or adding user-defined content. For example, add a company logo to the System Banner to have the logo appear at the top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

See also

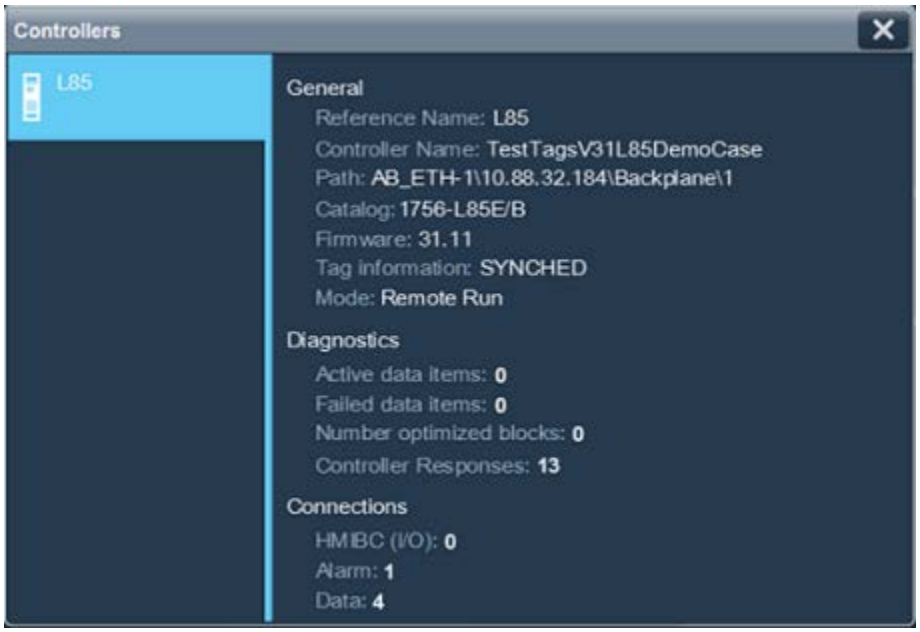
Predefined Screens and the system banner on page 19

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 and the system banner on page 19

System Banner contents on page 20

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 properties on page 23

Example: Bind a property on page 28

Example: Bind graphic elements on a screen on page 31

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.

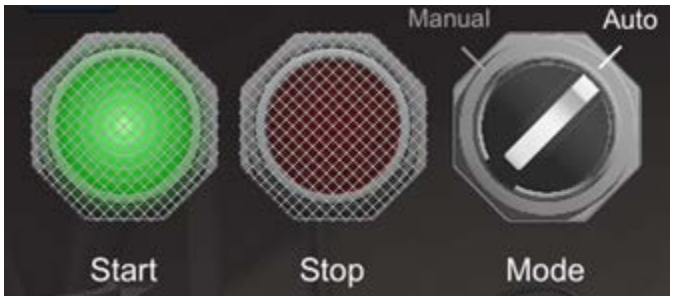
Important:	• Bind color properties only to string tags. Binding to an integer tag, a floating-point tag, or an expression are not available and do not produce a change in the color. • View Designer requires selecting a default color. Selecting NoFill does not apply the bound tag value at runtime.

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 needs to 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: Bind a property on page 28

Example: Bind graphic elements on a screen on page 31

Property binding on page 23

Supported color keywords and RGB values on page 26

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
aliceblue	F0F8FF
antiquewhite	FAEBD7
aqua	00FFFF
aquamarine	7FFFD4
azure	F0FFFF
beige	F5F5DC
bisque	FFE4C4
black	000000
blanchedalmond	FFEBCD
blue	0000FF
blueviolet	8A2BE2
brown	A52A2A
burlywood	DEB887
cadetblue	5F9EA0
chartreuse	7FFF00
chocolate	D2691E
coral	FF7F50
cornflowerblue	6495ED
cornsilk	FFF8DC
crimson	DC143C
cyan	00FFFF
darkblue	00008B
darkcyan	008B8B
darkgoldenrod	B8860B
darkgray	A9A9A9
darkgreen	006400
darkgrey	A9A9A9
darkkhaki	BDB76B
darkmagenta	8B008B
darkolivegreen	556B2F
darkorange	FF8C00
darkorchid	9932CC
darkred	8B0000
darksalmon	E9967A
darkseagreen	8FBC8F
darkslateblue	483D8B
darkslategray	2F4F4F
darkslategrey	2F4F4F
darkturquoise	00CED1
darkviolet	9400D3
deeppink	FF1493
deepskyblue	00BFFF
dimgray	696969
dimgrey	696969
dodgerblue	1E90FF
firebrick	B22222
floralwhite	FFFAF0
forestgreen	228B22
fuchsia	FF00FF

Color Name	#RRGGBB
gainsboro	DCDCDC
ghostwhite	F8F8FF
gold	FFD700
goldenrod	DAA520
gray	808080
grey	808080
green	008000
greenyellow	ADFF2F
honeydew	F0FFF0
hotpink	FF69B4
indianred	CD5C5C
indigo	4B0082
ivory	FFFFF0
khaki	F0E68C
lavender	E6E6FA
lavenderblush	FFF0F5
lawngreen	7CFC00
lemonchiffon	FFFACD
lightblue	ADD8E6
lightcoral	F08080
lightcyan	E0FFFF
lightgoldenrodyellow	FAFAD2
lightgray	D3D3D3
lightgreen	90EE90
lightgrey	D3D3D3
lightpink	FFB6C1
lightsalmon	FFA07A
lightseagreen	20B2AA
lightskyblue	87CEFA
lightslategray	778899
lightslategrey	778899
lightsteelblue	B0C4DE
lightyellow	FFFFE0
lime	00FF00
limegreen	32CD32
linen	FAF0E6
magenta	FF00FF
maroon	800000
mediumaquamarine	66CDAA
mediumblue	0000CD
mediumorchid	BA55D3
mediumpurple	9370DB
mediumseagreen	3CB371
mediumslateblue	7B68EE
mediumspringgreen	00FA9A
mediumturquoise	48D1CC
mediumvioletred	C71585
midnightblue	191970
mintcream	F5FFFA

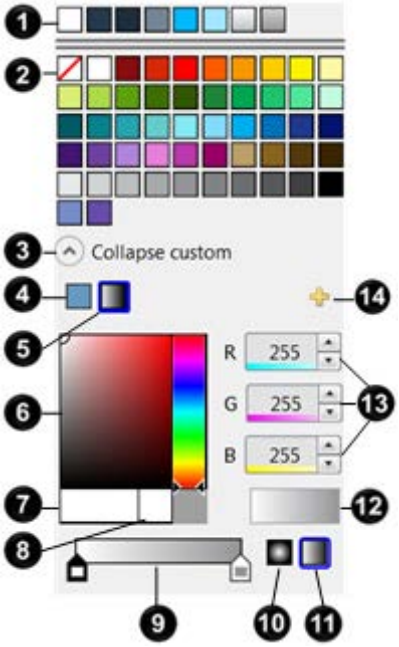
Color Name	#RRGGBB
mistyrose	FFE4E1
moccasin	FFE4B5
navajowhite	FFDEAD
navy	000080
oldlace	FDF5E6
olive	808000
olivedrab	6B8E23
orange	FFA500
orangered	FF4500
orchid	DA70D6
palegoldenrod	EEE8AA
palegreen	98FB98
paleturquoise	AFEEEE
palevioletred	DB7093
papayawhip	FFEFD5
peachpuff	FFDAB9
peru	CD853F
pink	FFC0CB
plum	DDA0DD
powderblue	B0E0E6
purple	800080
red	FF0000
rosybrown	BC8F8F
royalblue	4169E1
saddlebrown	8B4513
salmon	FA8072
sandybrown	F4A460
seagreen	2E8B57
seashell	FFF5EE
sienna	A0522D
silver	C0C0C0
skyblue	87CEEB
slateblue	6A5ACD
slategray	708090
slategrey	708090
snow	FFFAFA
springgreen	00FF7F
steelblue	4682B4
tan	D2B48C
teal	008080
thistle	D8BFD8
tomato	FF6347
turquoise	40E0D0
violet	EE82EE
wheat	F5DEB3
white	FFFFFF
whitesmoke	F5F5F5
yellow	FFFF00
yellowgreen	9ACD32

See also

Bind a property on page 23

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

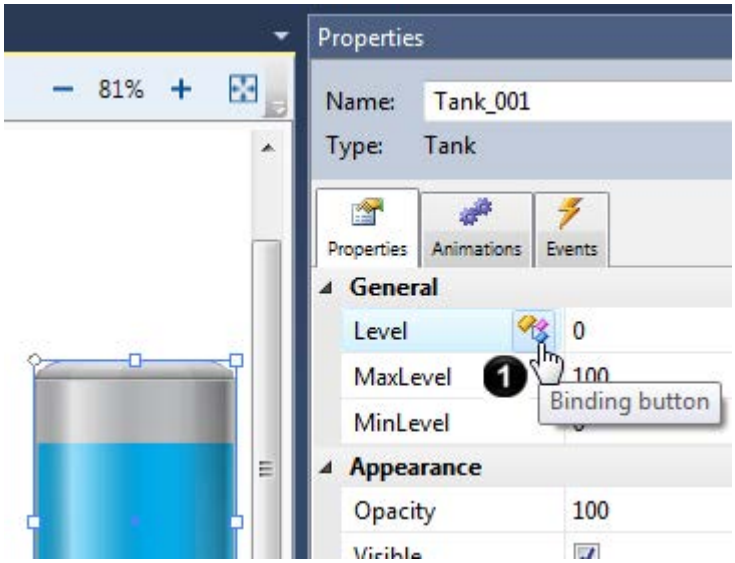
Example: Bind a property on page 28

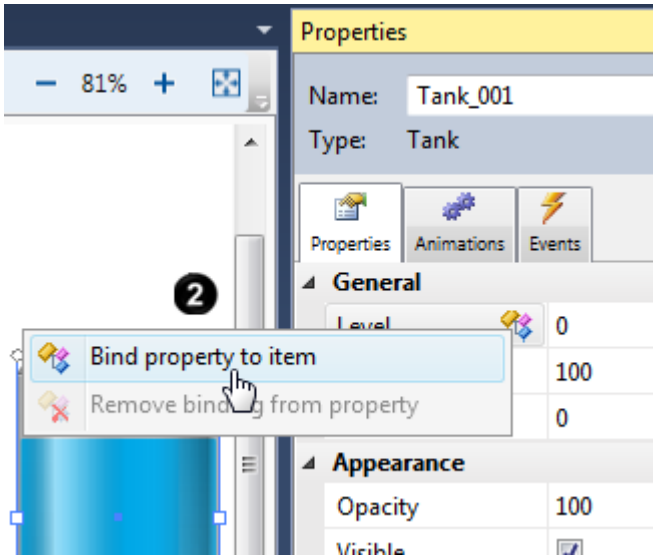
Example: Bind graphic elements on a screen on page 31

Expression Examples on page 32

Example 1: Bind a property

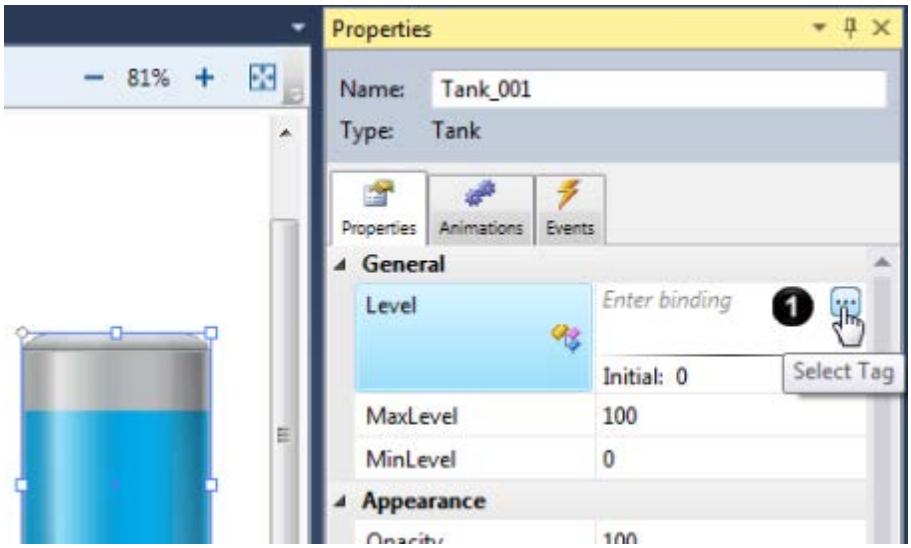
Many graphic elements in the View Designer **Toolbox** have built-in animations. Use animations by binding to their properties. For example, bind a tag or an expression to the **Level** property of the Tank element to animate the level shown on the screen. To create a binding, hover on the property in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

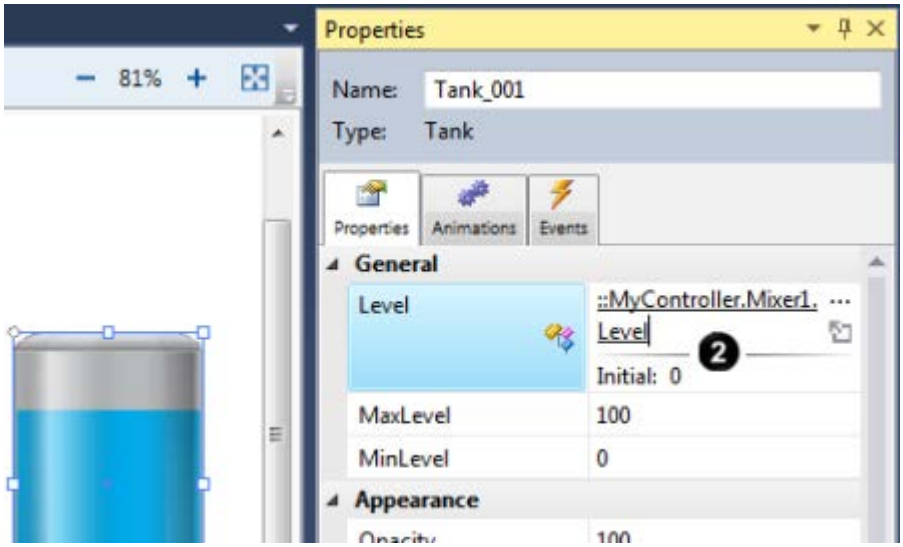




Item	Description
1	Binding button
2	Bind property to item

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

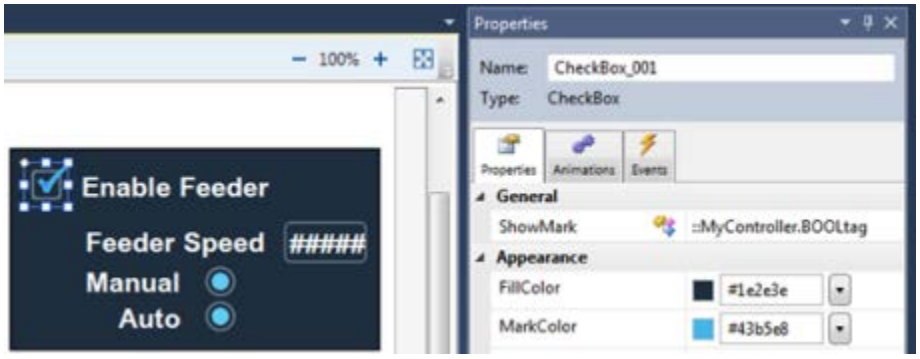
Bind properties on page 23

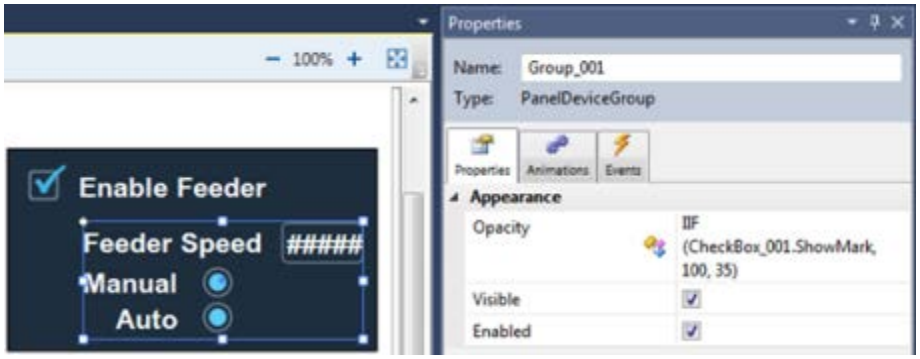
Property Binding on page 23

Example 2: Bind graphic element properties to other properties

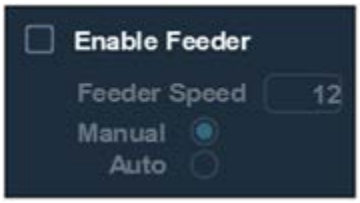
Creating bindings between graphical elements on a screen to have the property of one element affect other graphic elements.

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 properties on page 23

Property Binding on page 23

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 32

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	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. Keep the following in mind when using string tags with the relational operators: • 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	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 `::ControllerName.TagName.` To find a tag, type the tag name or open the Tag Browser and select a tag.	`::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.tag1==5)` returns 1 (both statements are true)

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 non-zero 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 non-zero (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: • 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.	`tag1 == tag2` 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.	`tag1 != tag2` 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.	`tag1 < tag2` 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.	`tag1 > tag2` 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.	`tag1 <= tag2` 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.	`tag1 >= tag2` 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 (non-zero).	`(tag1 < tag2) && (tag1 == 5)r` returns a 1 because both statements are non-zero (true) `tag1 && tag2 r` returns a 1 because both tag1 and tag2 are non-zero (true)
|| (Logical OR)	Returns a value of 1 if either the statement to the left or to the right of the operator is true (non-zero).	`(tag1 > tag2) || (tag1 == 5)r` returns a value of 1 because tag1 == 5 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 (a>b) evaluates to true, then !(a>b) evaluates to false.	`!(tag1 < tag2)` The expression tag1 < tag2 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 tag1 = 5 (binary 0000 0000 0000 0101) and tag2 = 3 (binary 0000 0000 0000 0011), 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.	`tag1 & tag2` returns 1 (binary 0000 0000 0000 0001)
| (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.	`tag1 | tag2` returns 7 (binary 0000 0000 0000 0111)
^ (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.	`tag1 ^ tag2` returns 6 (binary 0000 0000 0000 0110)
>> (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.	`tag1 >> 1` returns 2 (binary 0000 0000 0000 0010)
<< (Left Shift)	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. If the left bit is a 1, an overflow occurs, and an error message appears. To prevent this, use the bitwise AND (&&) operator in an expression. For example, (dev << 1) && 65535, where 65535 is 1111 1111 1111 1111 in binary form. Where dev is a tag name whose value is shifting left.	`tag1 << 1` returns 10 (binary 0000 0000 0000 1010)
~ (Complement)	Gives the ones complement of a number. For example, use this operator to reverse every bit within the number so that every 1 bit becomes a 0 and vice versa.	`~ tag1 r`eturns -6 (binary 1111 1111 1111 1010)

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	Property names refer to components of the current application. This differs from tags, which refer to data items in a controller. The graphic elements must all be on the same screen, popup, or **System Banner**. Expressions do not support alias properties. Use the syntax: ElementName.PropertyName An expression can have: • Properties that support bindings. • Multiple graphic element properties. • Graphic element property bindings AND tags AND user-defined properties or any combination.	MyElement.X + MyElement.Y / MyElement.Opacity MyElement.X + ::MyController.MyTag + MyCustomProperty

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 32

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 fixed rate of 500 milliseconds. This update is asynchronous to any scan in a controller.

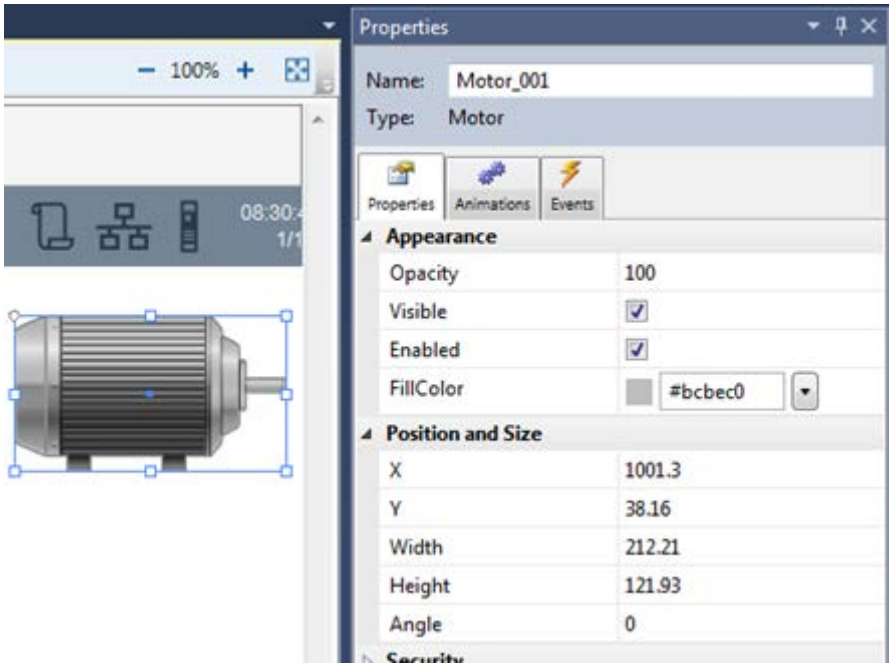
See also

Example: Use a state table to make a motor a multi-state indicator on page 42

Example: Use a color table to indicate level in a mixer on page 44

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

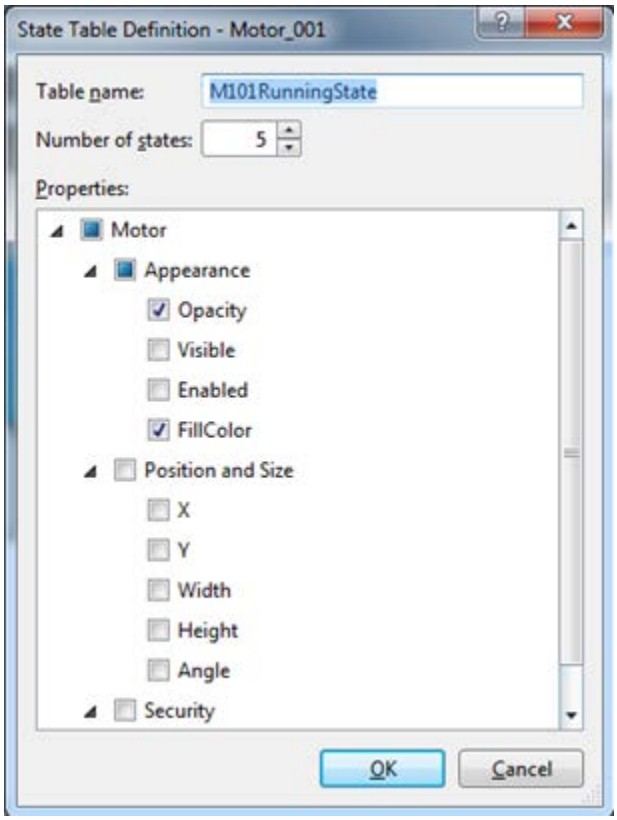
Place a motor graphic element on the screen. The motor has properties, such as **FillColor** and **Opacity**.



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

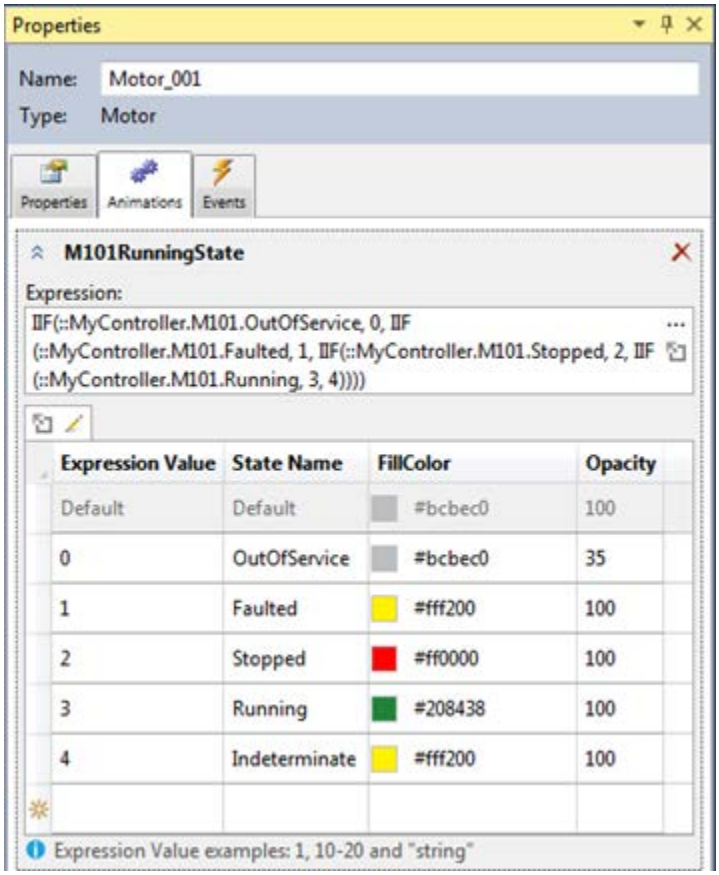
Use the **State Table Definition** dialog box to define a name for the state table, the number of states, and the properties to manipulate. Change these values at any

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.



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

See also

Example: Use a color table to indicate level in a mixer on page 44

Color and State tables on page 41

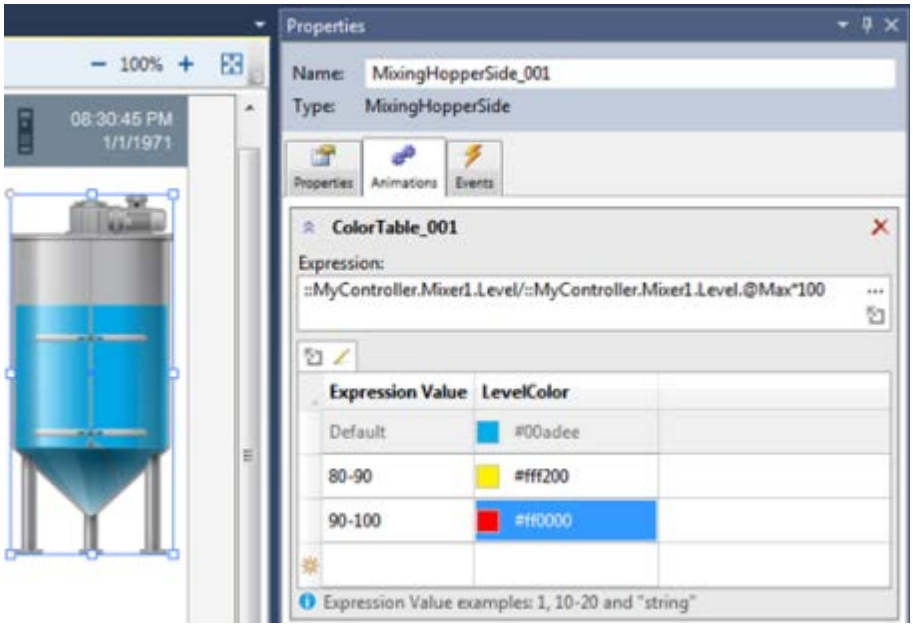
Events on page 47

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 47

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.
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.

Event Type	Triggers	Common uses
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 49

Example 2: Configure multiple events on a graphic element on page 50

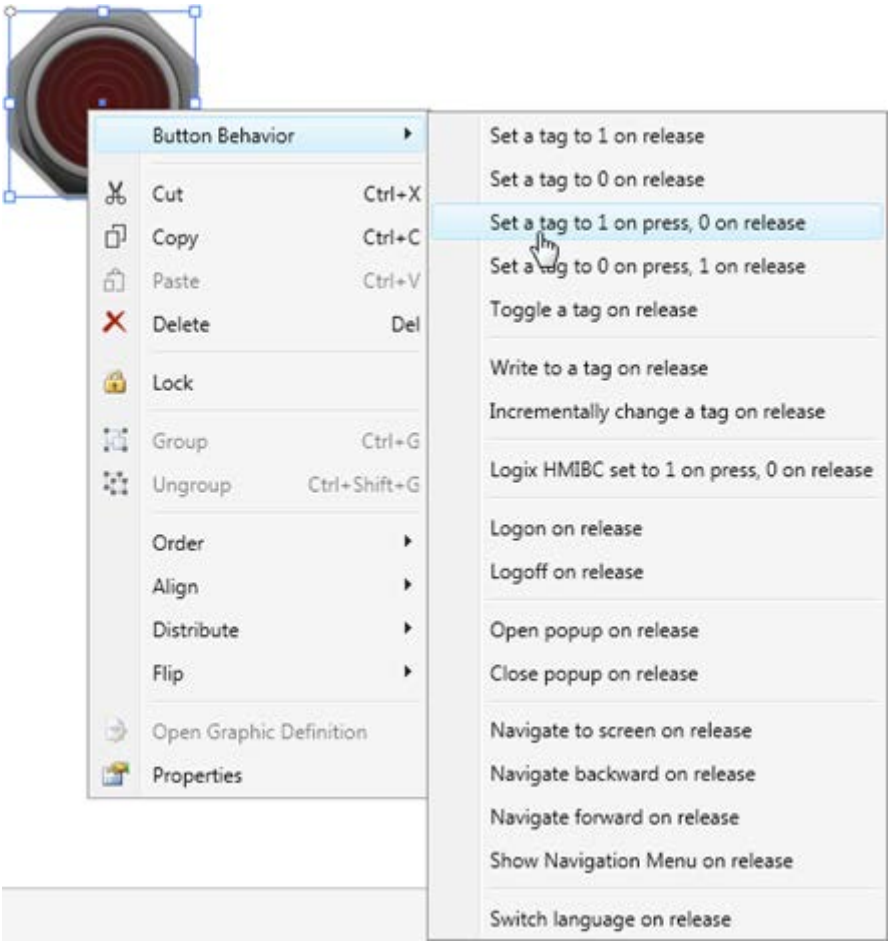
Example 3: Use State Enter and State Exit Events on page 56

Example 4: Use key press and key release events on page 58

Example 5: Use key press and key release events on page 58

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 the M101.Start bit resets when the operator stops pressing the button, or when a **State Enter** or **State Exit** event causes the screen to navigate to a different screen.

See also

Example 2: Configure multiple events on a graphic element on page 50

Example 3: Use State Enter and State Exit Events on page 56

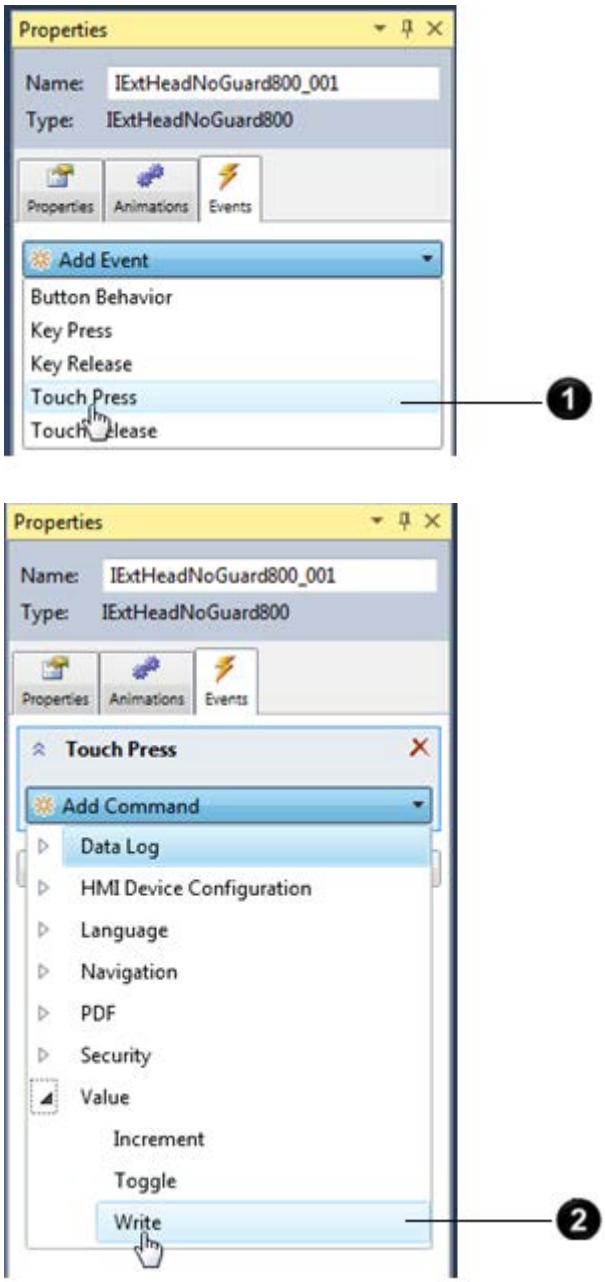
Example 4: Use key press and key release events on page 58

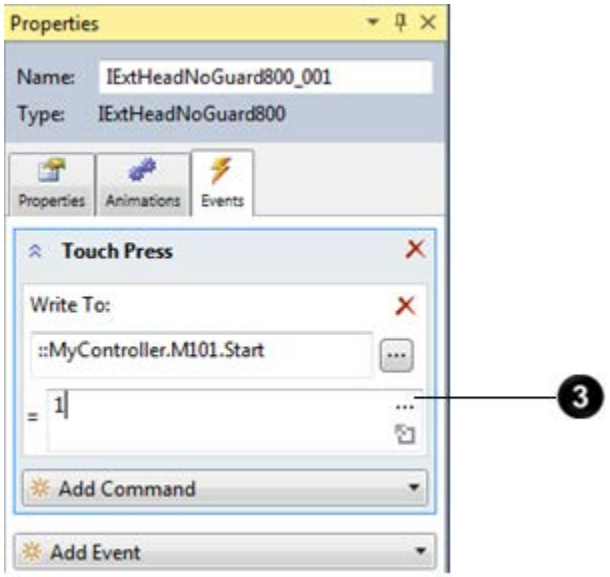
Example 5: Use key press and key release events on page 58

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.

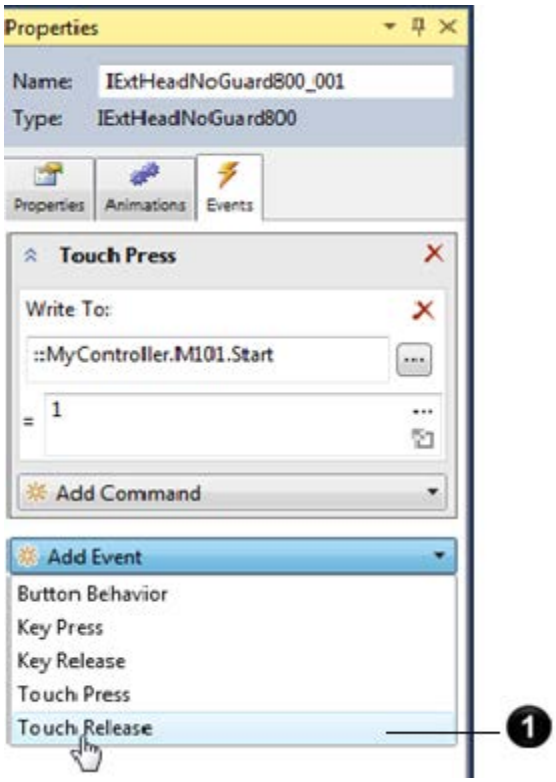
Configure a momentary button with press and release events. Define a **Touch Press** event and configure a **Write** command to set the Motor101.Start tag to 1.

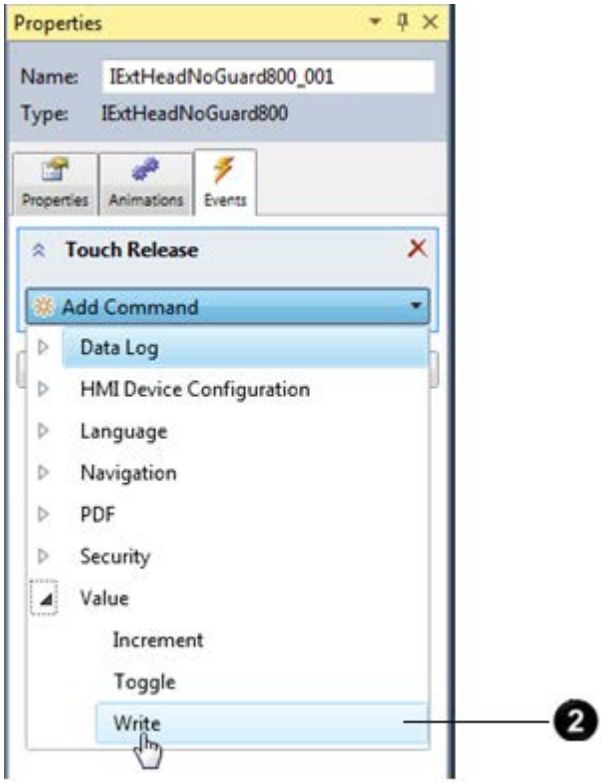


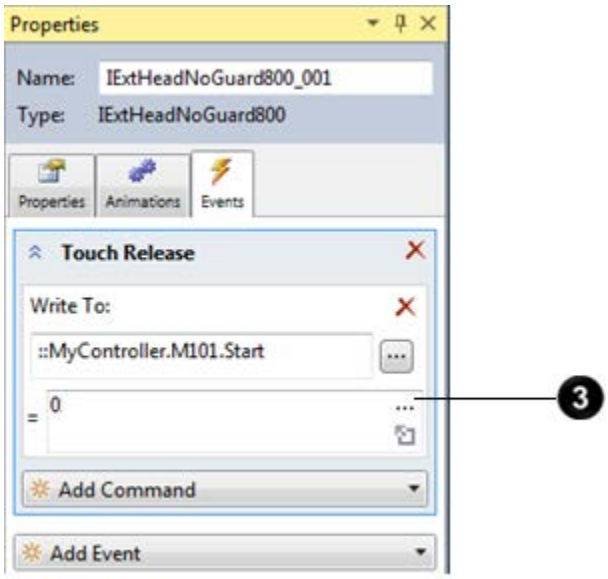


Item	Description
1	Select Touch Press event
2	Add Write command
3	Configure a command for a Touch Press event.

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.



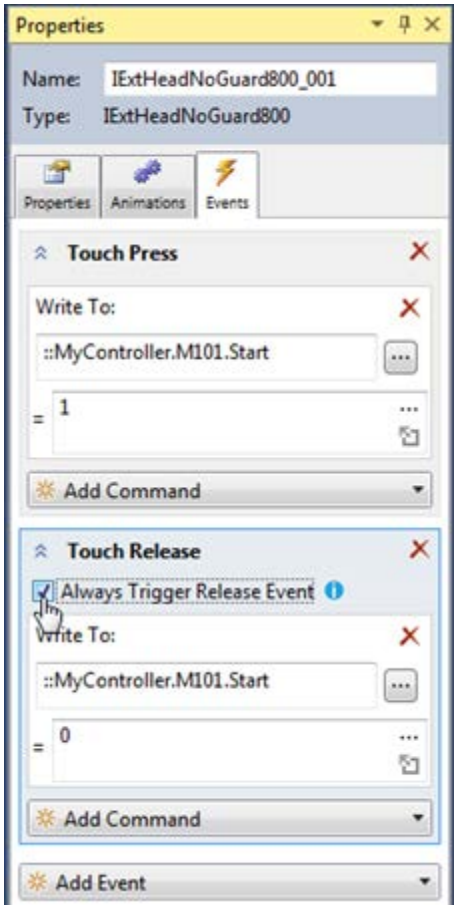




Item	Description
1	Select Touch Release event
2	Add Write command
3	Configure a command for Touch Release event.

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 75

Events on page 47

Example 3: Use State Enter and State Exit Events on page 56

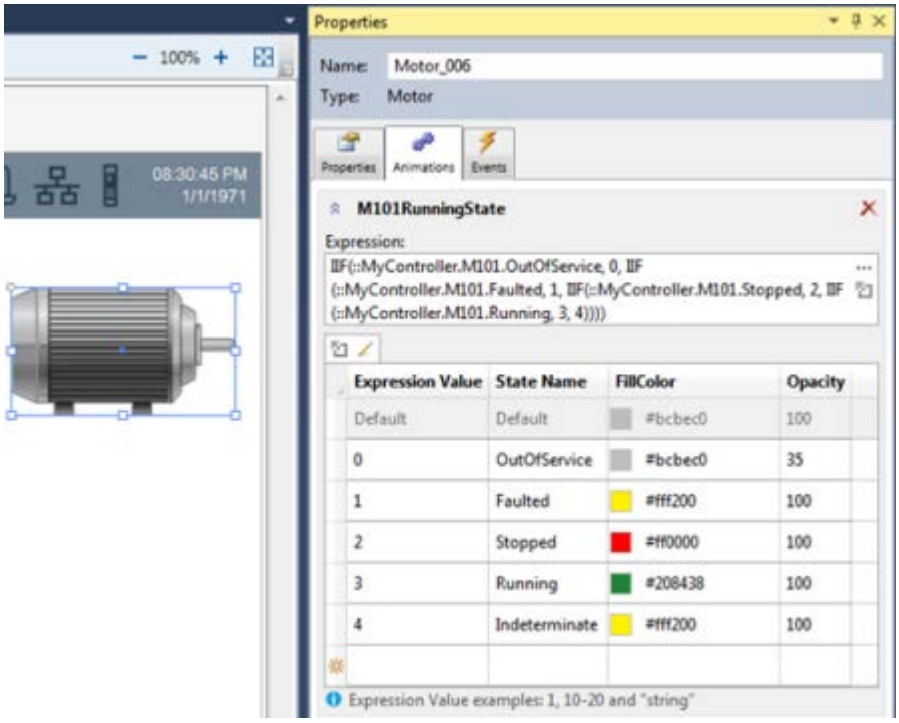
Example 4: Use key press and key release events on page 58

Example 5: Use key press and key release events on page 58

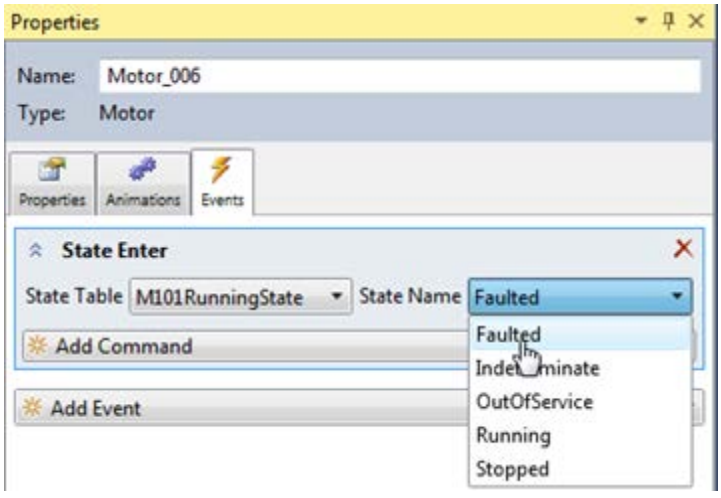
Example 3: Use State Enter and State Exit Events

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.

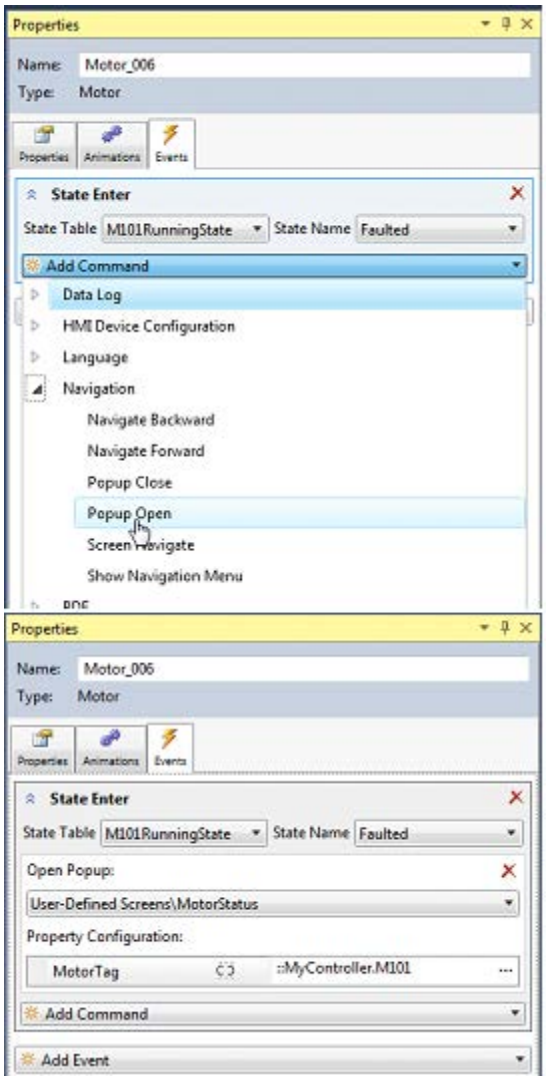
Connect State Enter and State Exit events to state tables for the events to operate. For example, assume a motor graphic element has a M101RunningState state table where multiple states animate the motor running state.



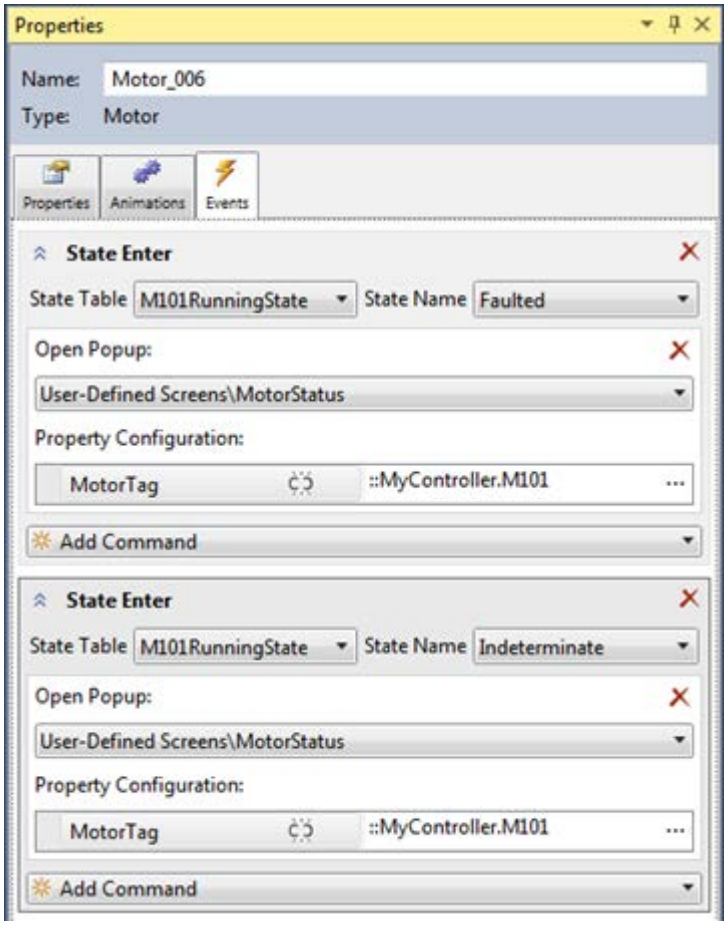
Define state events to trigger a Popup Open command to display a motor status detail popup screen when the motor enters a Fault or Indeterminate state. Configure the State Enter event for the motor entering the Faulted state. In the **Events** tab for the motor, create a **State Enter** event. Select 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 75

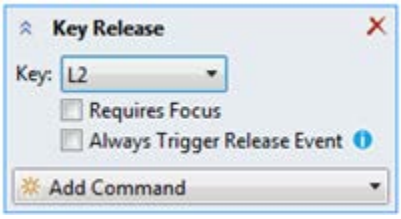
Events on page 47

Example 4: Use key press and key release events on page 58

Example 5: Use key press and key release events on page 58

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 47

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 61

Example: Write the current user name or screen name to a controller using a project event on page 65

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 1,000 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.

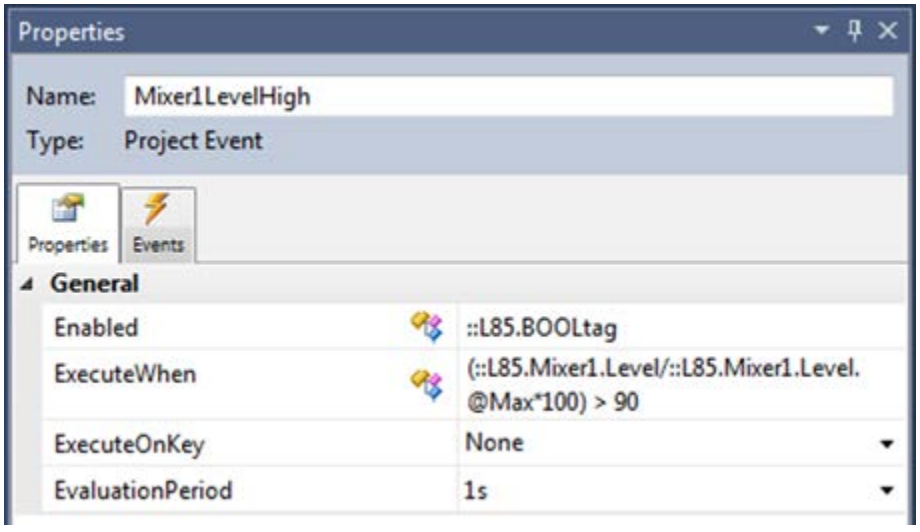
  Tip: Multiple screen navigation commands can exist for an event. All other types of commands for the event are executed before the navigation command. Only one navigation command is executed for an event. This is generally the first navigation command. The remaining navigation commands are ignored on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

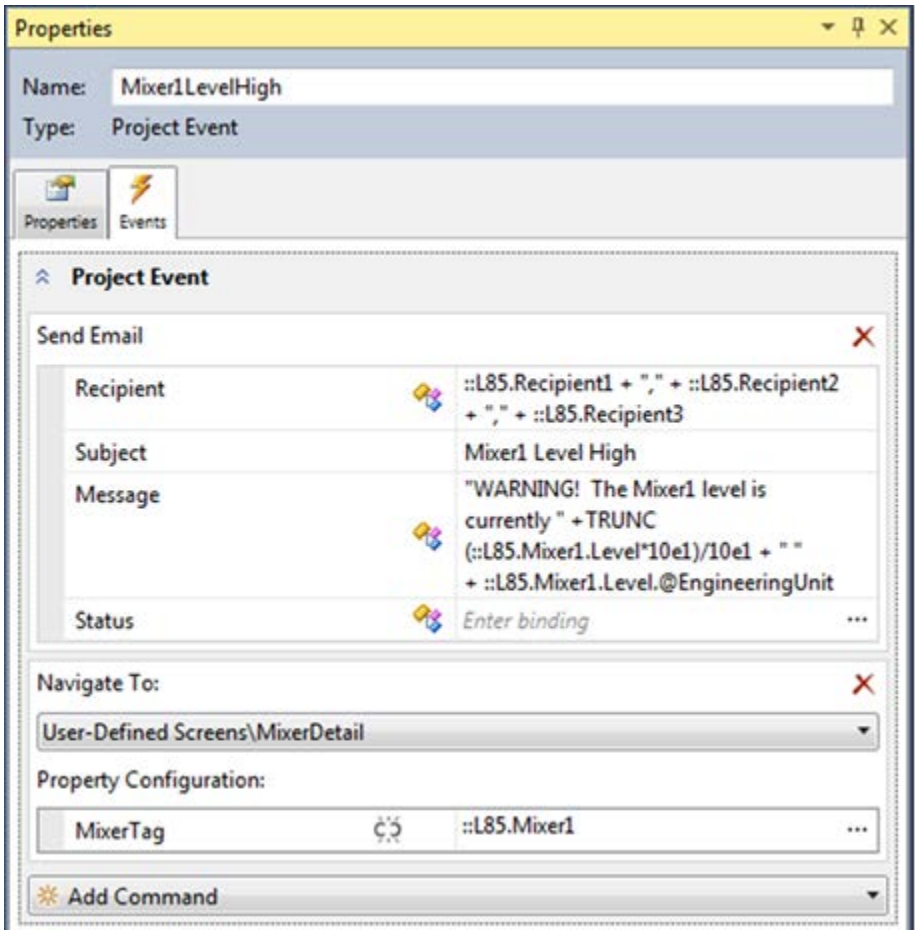
- **Notification**. Commands that send an email notification.
- **Security**. Commands that log on or log off the HMI device.
- **Value**. Commands that increment, toggle, or write a value to the selected tag.

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 1,000 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 very quickly can overload the project queue. If this happens, View Designer may skip project evaluation periods.
- If you define multiple navigation commands in a Project Event, View Designer executes only one navigation command for the event. For consistent results, use one navigation command for a Project Event.
- View Designer 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%.





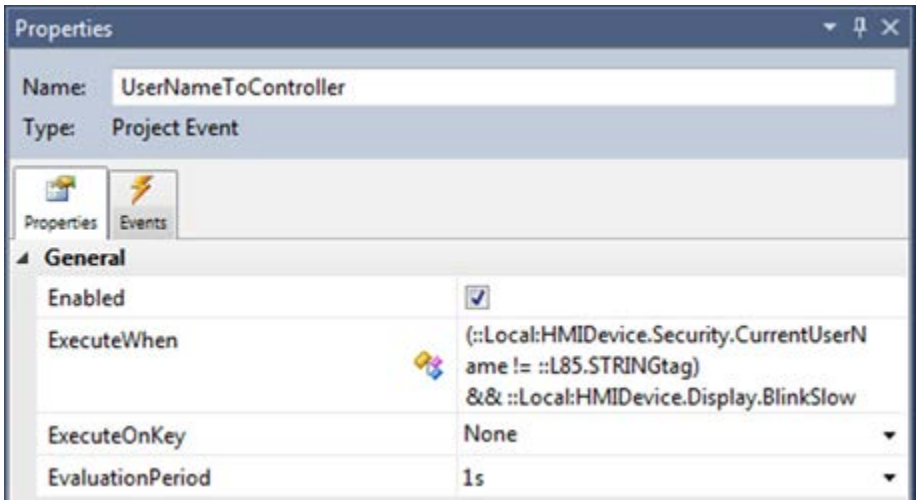
See also

Project Events on page 61

Example: Write the current user name or screen name to a controller using a project event on page 65

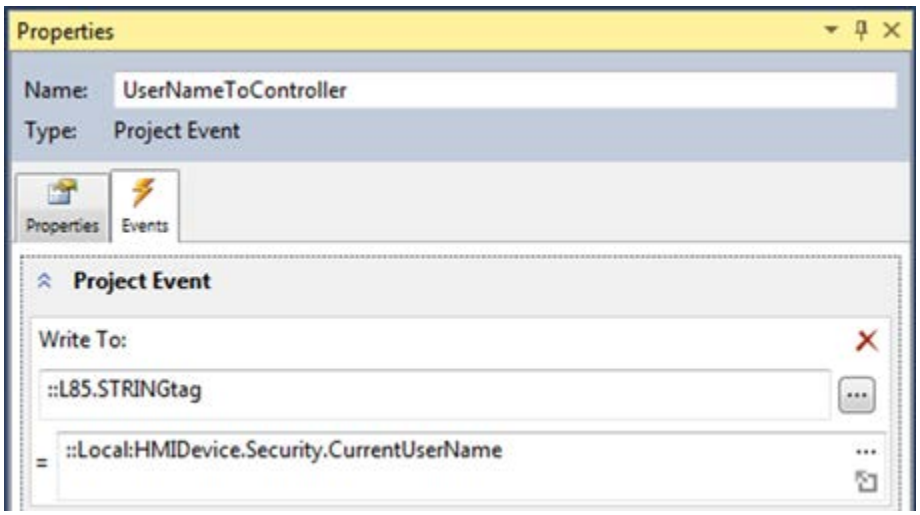
Example: Write the current user name or screen name to a controller using a project event

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 configure the **ExecuteWhen** property of the project event:

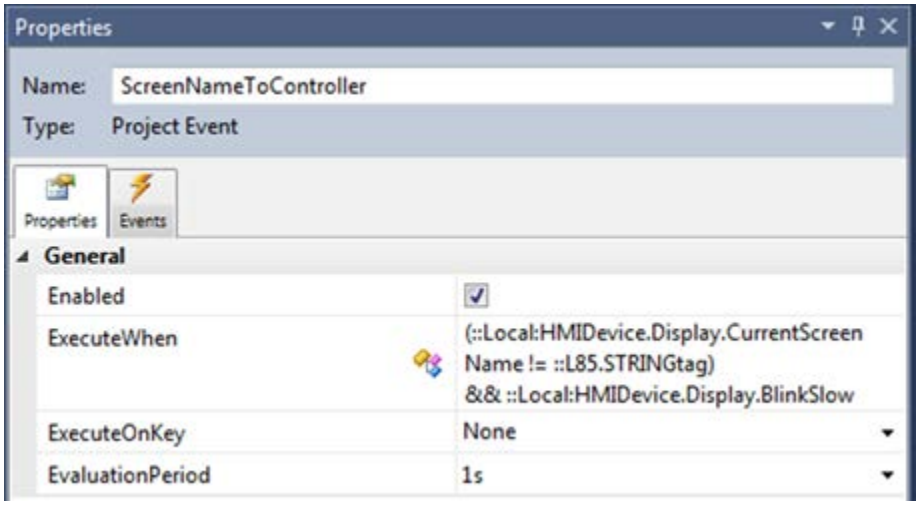


When the **CurrentUserName** local HMI tag is not equal to the value of a string tag in the controller, **ExecuteWhen** becomes true. The AND operator with the **BlinkSlow** local HMI tag forces a false to true transition for **ExecuteWhen** when starting the PanelView 5000.

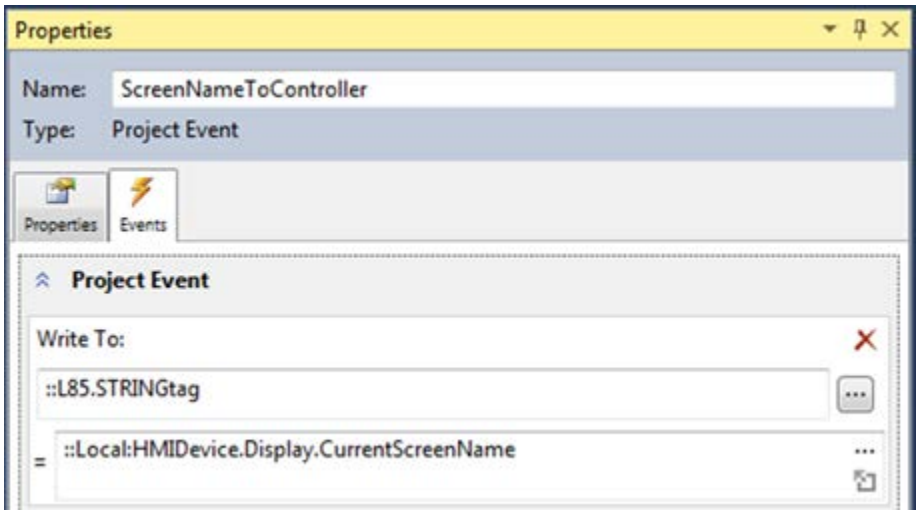
On the **Events** tab, configure a **Write To** command. Write the value of the **CurrentUserName** to the string tag when the UserNameToController project event triggers.



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 61

Navigation menu

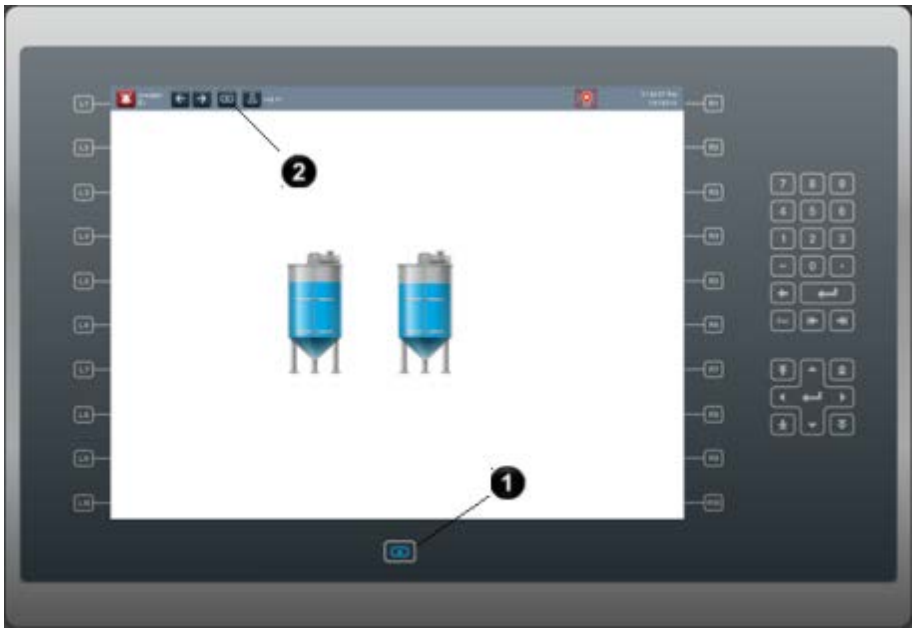
Use the navigation menu on the PanelView 5000 HMI device to set up navigation for operators. The navigation menu:

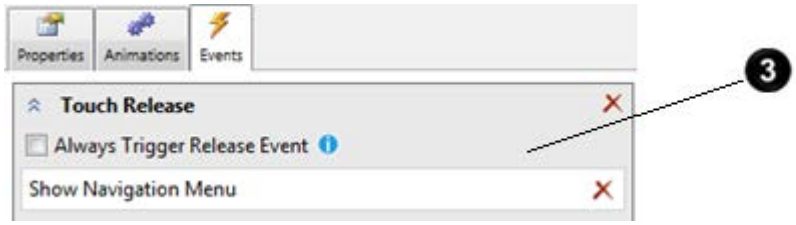
- Enables configuring navigation in the **Navigation** folder of **Project Explorer** to eliminate configuring navigation buttons on every screen.
- Saves screen space by sliding up when pressing the Navigation button and sliding down when selecting a screen.
- Automatically hides navigation shortcuts 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
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

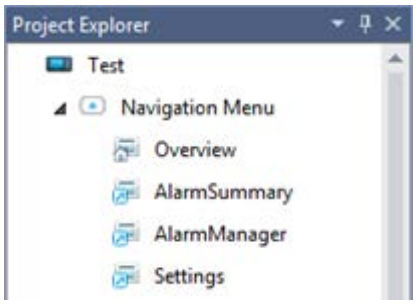
Define shortcuts to setup navigation menu on page 69

Example: Configure Navigation Menu properties for runtime on page 70

Events on page 47

Define shortcuts to setup Navigation Menu

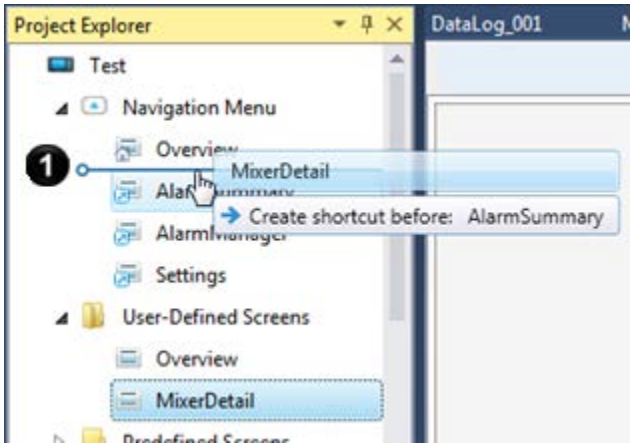
To set up the navigation menu, define shortcuts in the **Navigation Menu** folder in the **Project Explorer**.



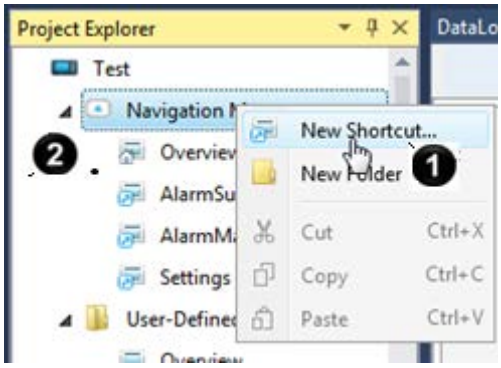
View Designer refers to items in the **Navigation Menu** folder as Shortcuts. Shortcuts are navigation links to the user-defined or pre-defined screens in a project.

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



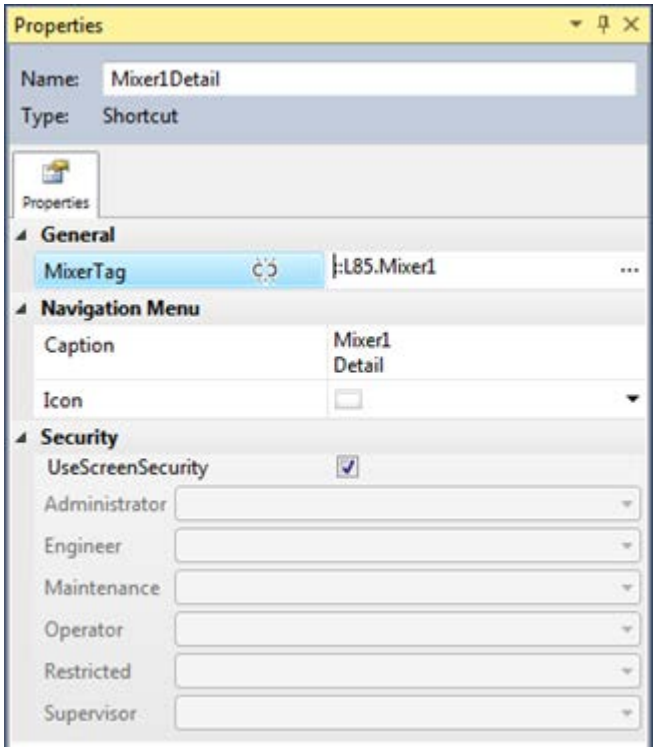
Item	Description
1	**New Shortcut**
2	**Navigation Menu** folder

See also

Example: Configure Navigation Menu properties for runtime on page 70

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:

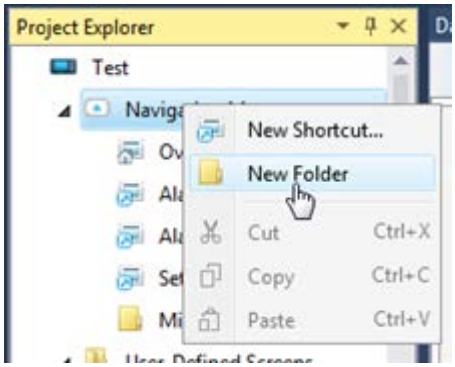
- Displays on the navigation menu icon for the shortcut
- Text defaults to the shortcut name
- Enables entering spaces
- Uses the Enter key to make a two-line caption
- Enables entering Unicode characters as necessary
- Supports language switching to enable operators to see the **Navigation Menu** items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

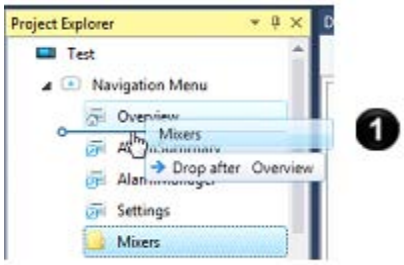
Use the **Security** property to define different security access for the shortcut. Typically, select the **UseScreenSecurity** checkbox to enable the shortcut to use the same security configuration as the screen it references.

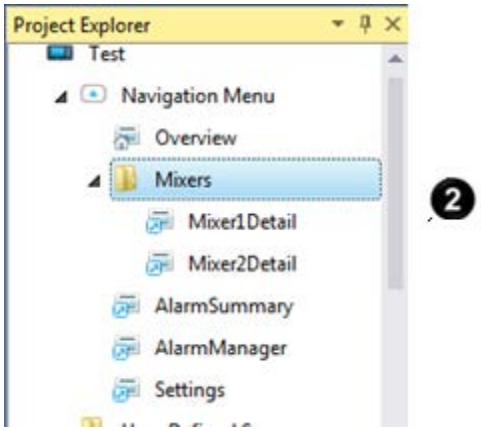
Tip: Specify security access for a user role when a reusable screen needs user roles to access different instances of a shortcut.

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**.



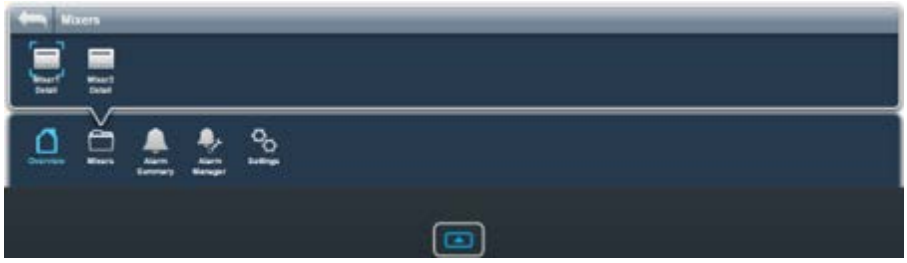
View Designer supports renaming the folder and dragging it to a desired place in the **Navigation Menu**. You can also drag other shortcuts into the folder or create new shortcuts in the folder.





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:

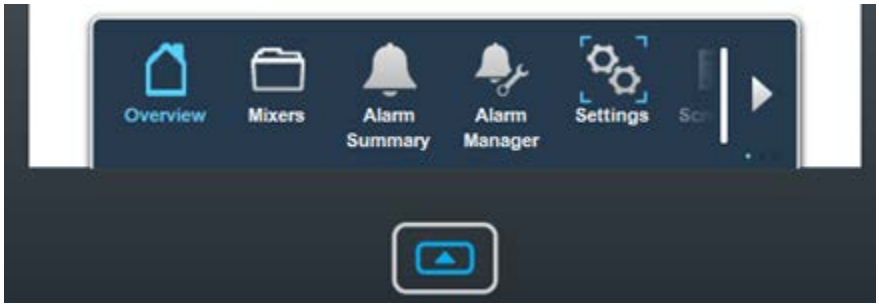


Complex navigation menu configurations support:

- 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 67

Define shortcuts to setup navigation menu on page 69

Reuse screens on page 75

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.

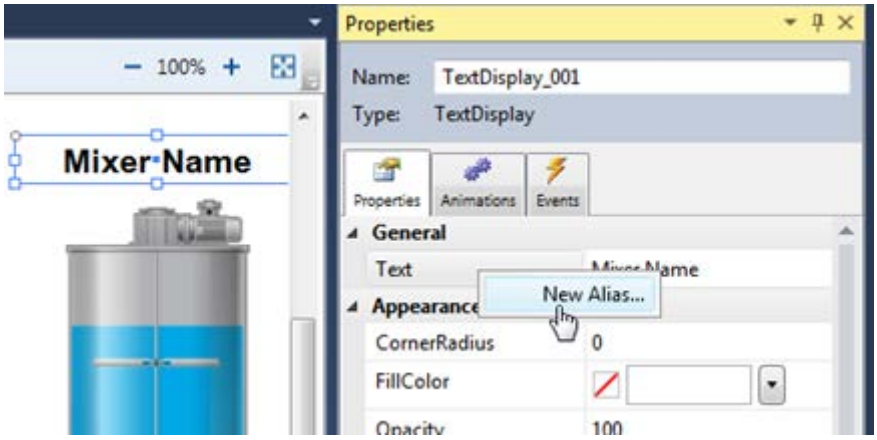
See also

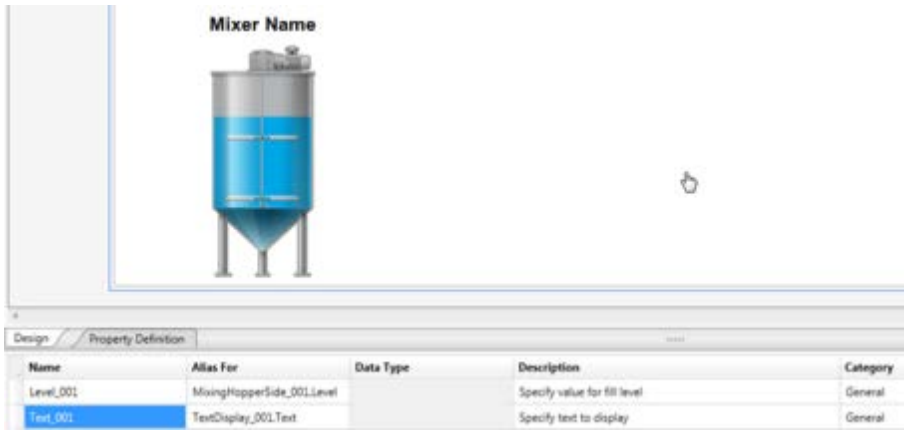
Example 1: Use alias properties on page 76

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.



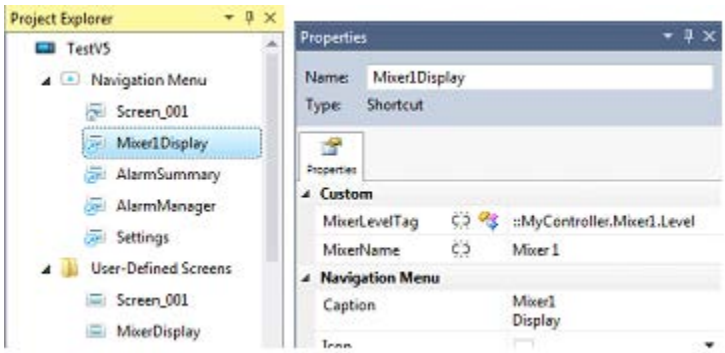


Use the **Property Definition** pane to:

- 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.

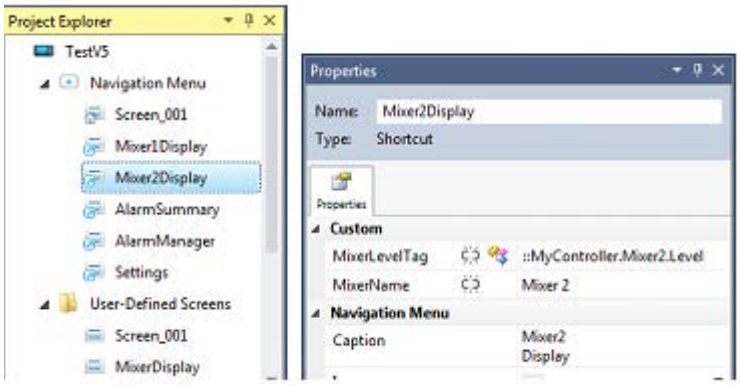
Set up a navigation to the **MixerDisplay** screen by dragging the **MixerDisplay** screen into the **Navigation Menu** folder. This creates a shortcut to the **MixerDisplay** screen on the runtime **Navigation Menu**. Rename the shortcut *Mixer1Display*. The properties of the shortcut display the configurable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

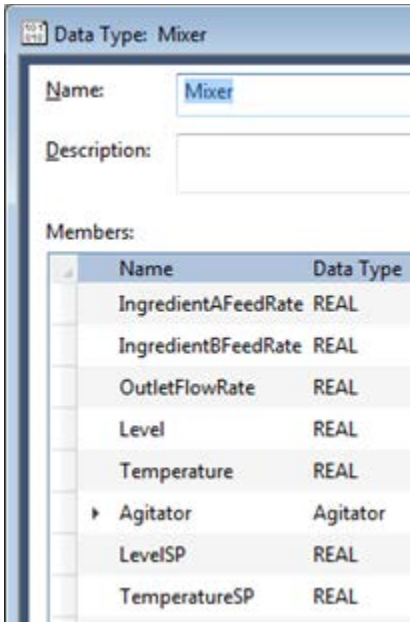
See also

Example 2: Use data type properties to create a reusable screen on page 80

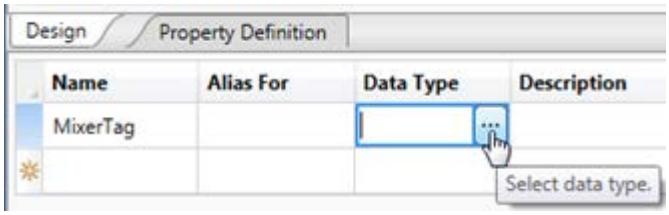
Reuse screens on page 75

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

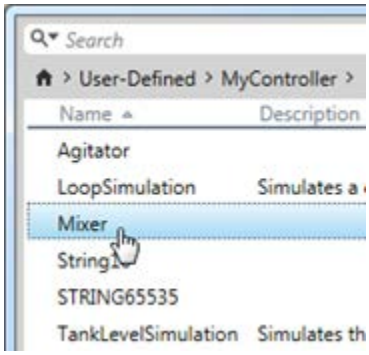
If Logix 5000™ represents equipment data as a data type, use instances of that data type to animate graphic elements on a screen. Use a custom screen property tied to the Logix data type. Create a *MixerDetail* screen that shows the status of a mixer represented by a user-defined *Mixer* data type in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



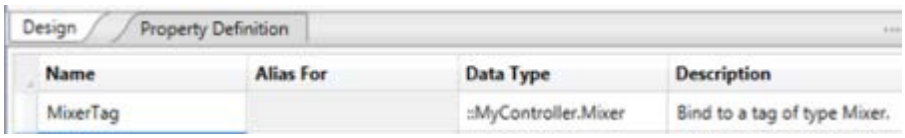
Create a custom property named MixerTag for the **MixerDetail** screen.



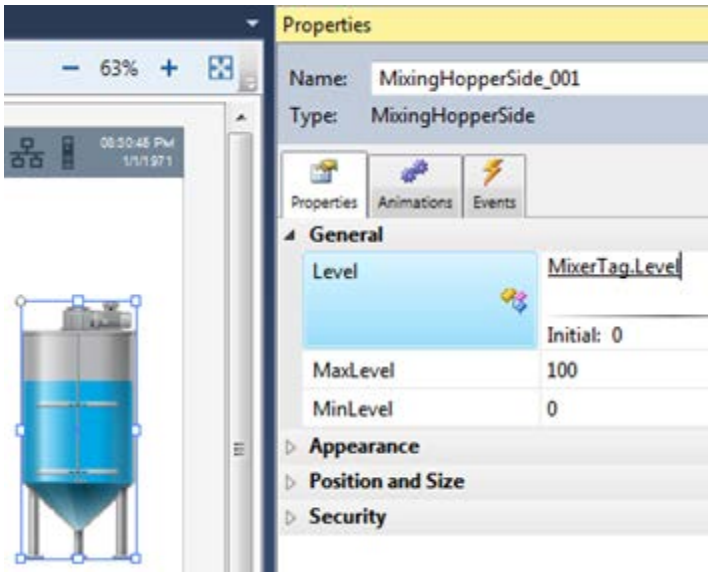
Browse the **Data Type** field to display all the data types in the Logix 5000 project and select the user defined Mixer data type.



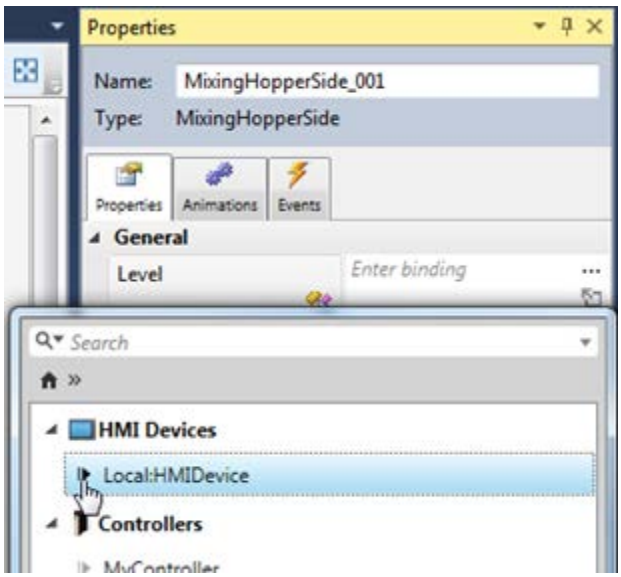
Enter an optional description for the custom MixerTag property.



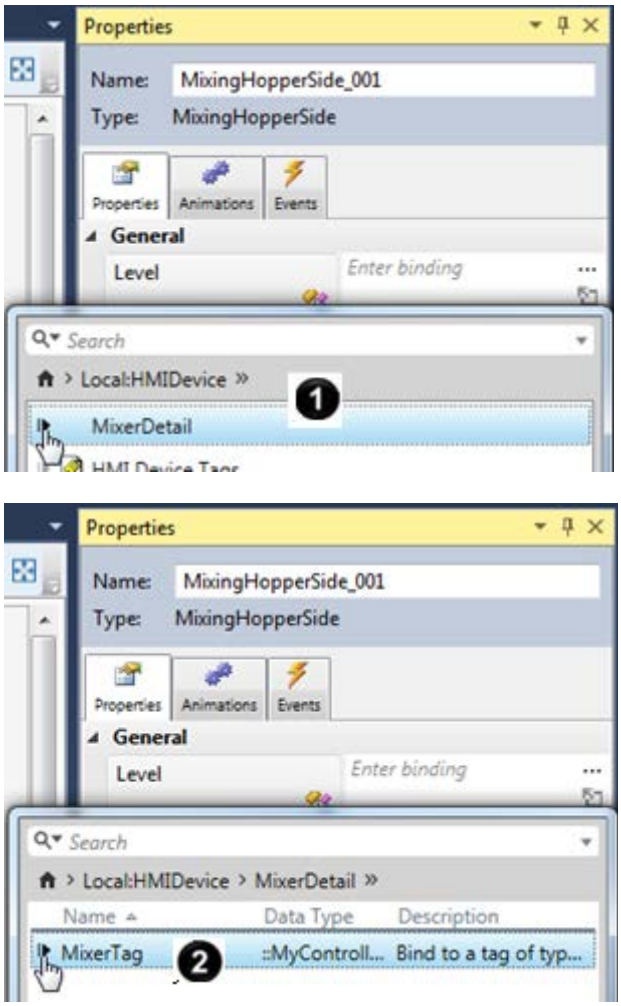
The custom MixerTag property appears as a property of type Mixer. View Designer supports using the definition of the data type to create animation bindings for graphic elements on the **MixerDetail** screen. For example, add a Mixer graphic element to the screen and bind the **Level** property to the **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:



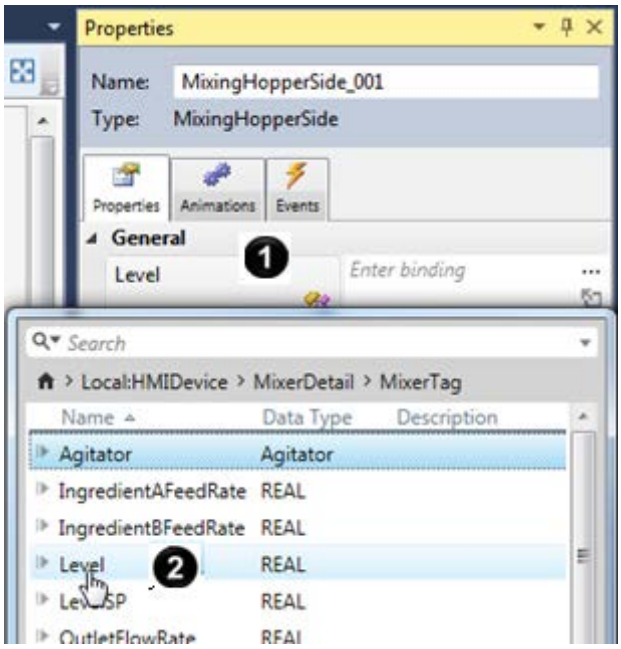
In this example, select the drill-in arrow next to **MixerDetail** to access the **MixerDetail** screen to view the custom MixerTag property.



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

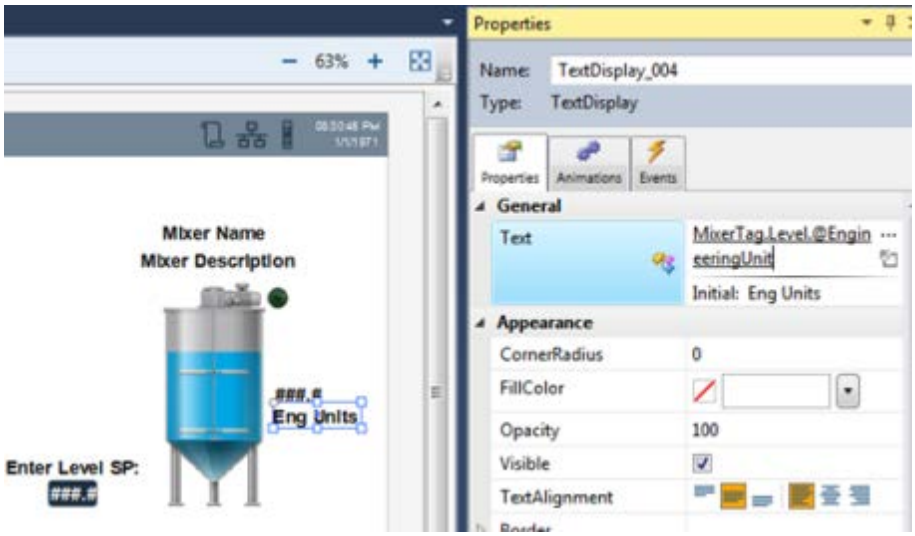
type definition. Then select the **Level** member of the **MixerTag** to bind to the **Level** property of the Mixing tank.



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

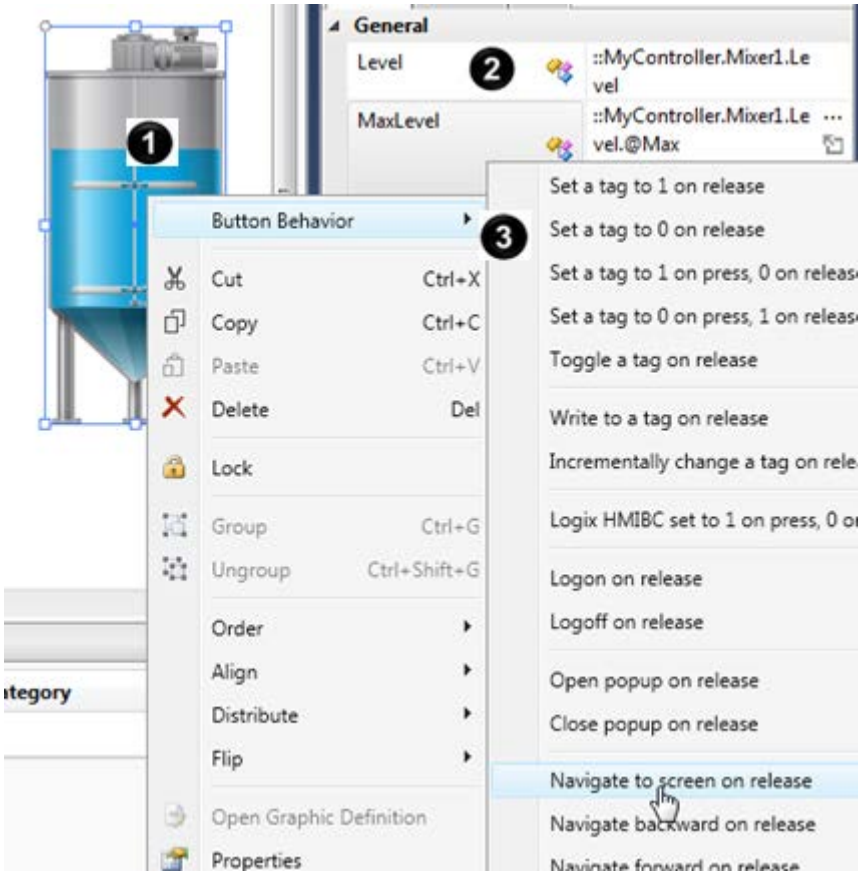
Browse the data type definition to configure all the bindings on the **MixerDetail** screen to reduce time and ensure accuracy instead of manually typing data type member names.

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

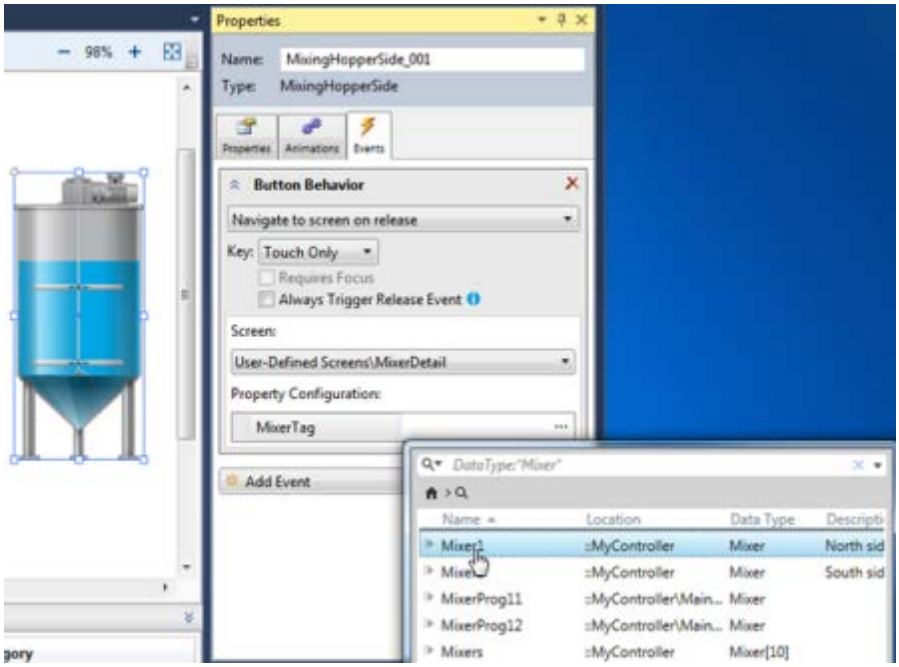
desired Mixer tags. Then create a **Button Behavior** to configure the element to act as a button.



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

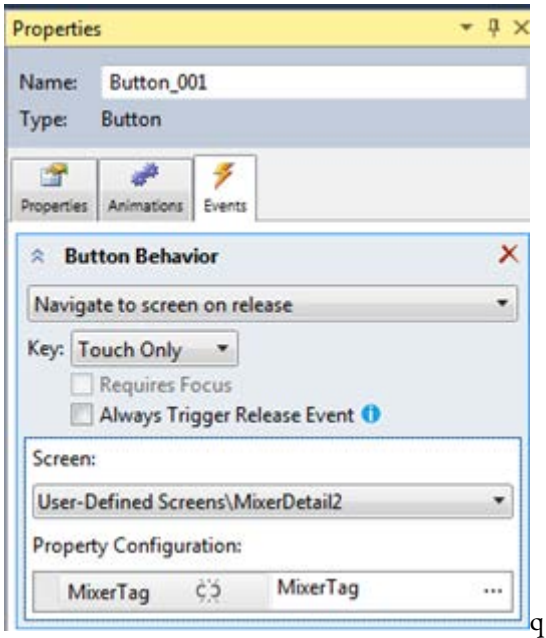
When selecting **Navigate to screen on release**, the **Events** tab in the **Properties** pane opens. Select the **MixerDetail** screen. Open the tag browser to select a Mixer tag instance for the MixerTag property. The tag browser filters

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.



q

This takes the MixerTag instance used by MixerDetail and passes it to the **MixerTag** property for **MixerDetail2**.

See also

Reuse screens on page 75

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

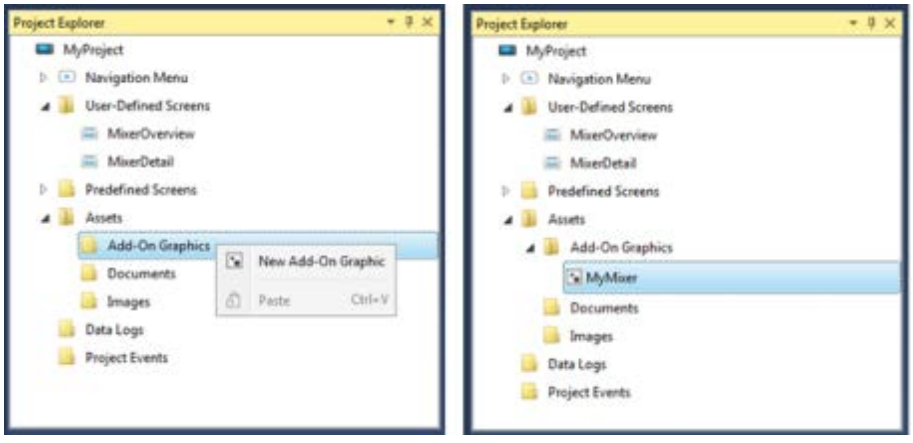
Create Add-On Graphic on page 89

Example: Use custom properties for Add-On Graphic on page 91

Create an instance for the Add-On Graphic definition on page 93

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 Add-On Graphic on page 91

Create an instance for the Add-On Graphic definition on page 93

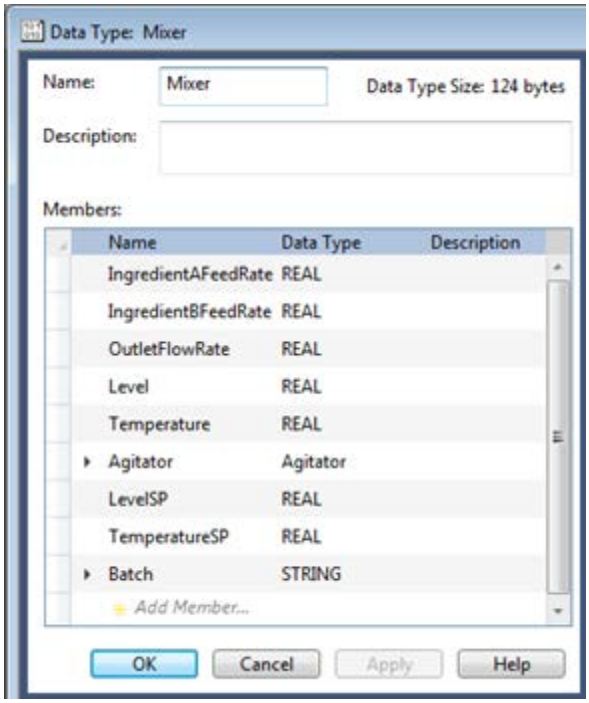
Example: Use custom properties for an Add-On Graphic

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.

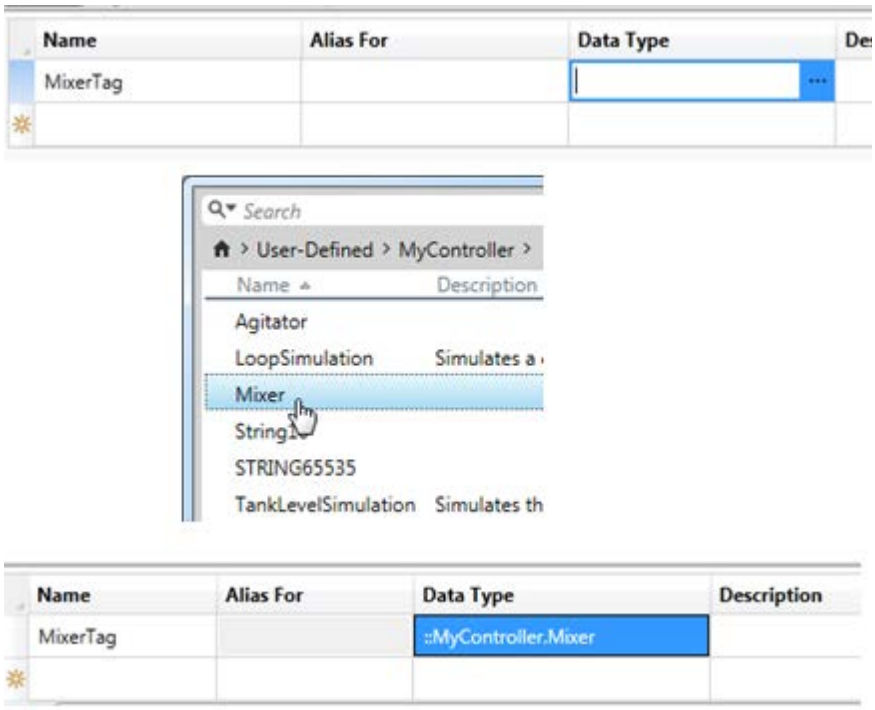
In this example, tie the **MyMixer** add-on graphic definition and its contents to a user-defined data type using the name Mixer in the Logix controller project. The Mixer data type in Logix Designer has the following members.



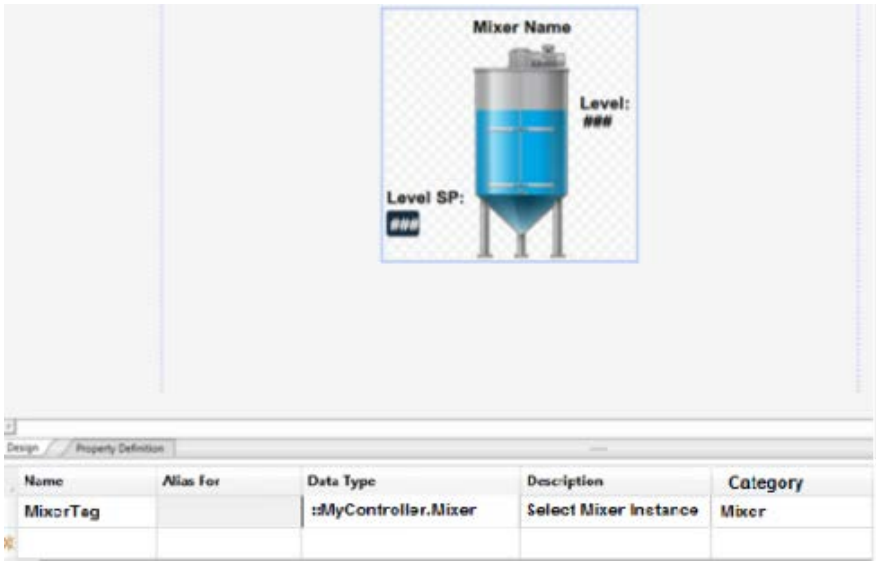
Pass in data from the Mixer data type instances in the controller to animate instances of the **MyMixer** Add-On Graphic used in the HMI. Create a custom property in the Add-On Graphic definition tied to the Mixer data type in the controller project.

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.



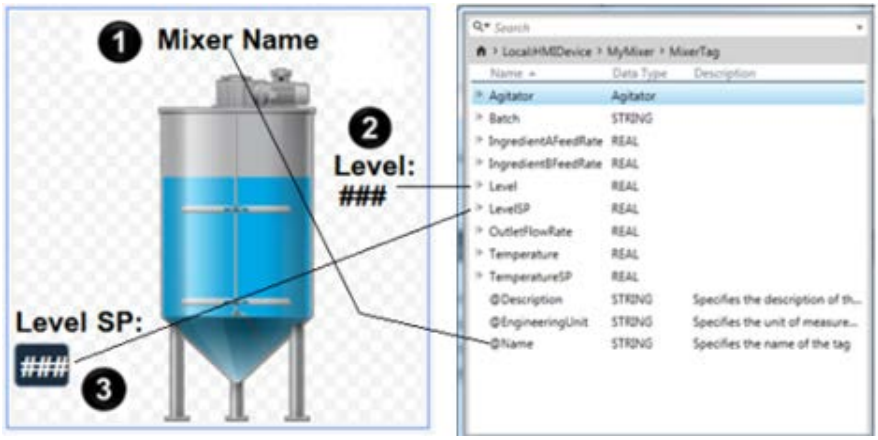
Enter a description and category to specify where the property appears in the properties pane.



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

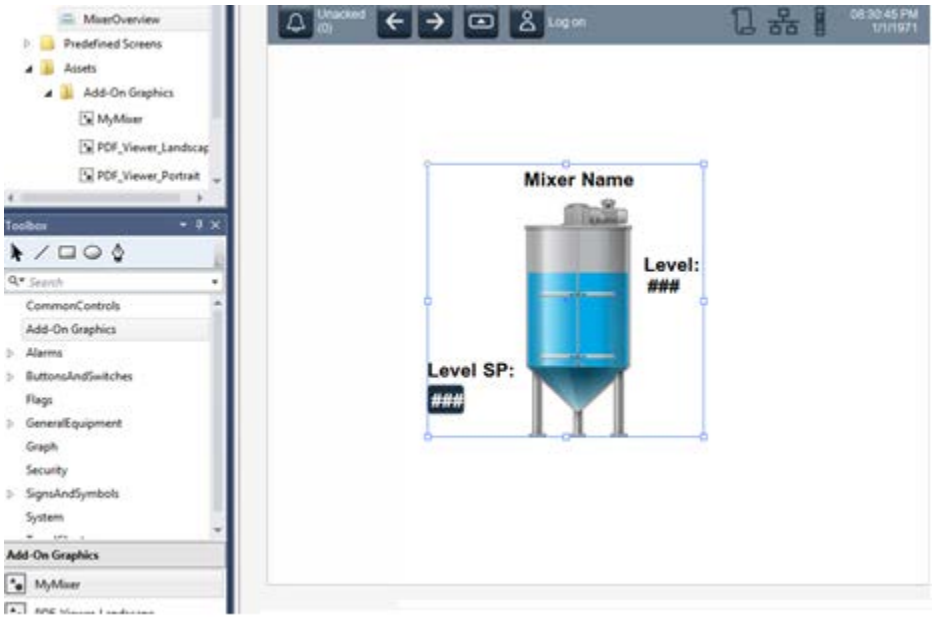
See also

Create an instance for the Add-On Graphic definition on page 93

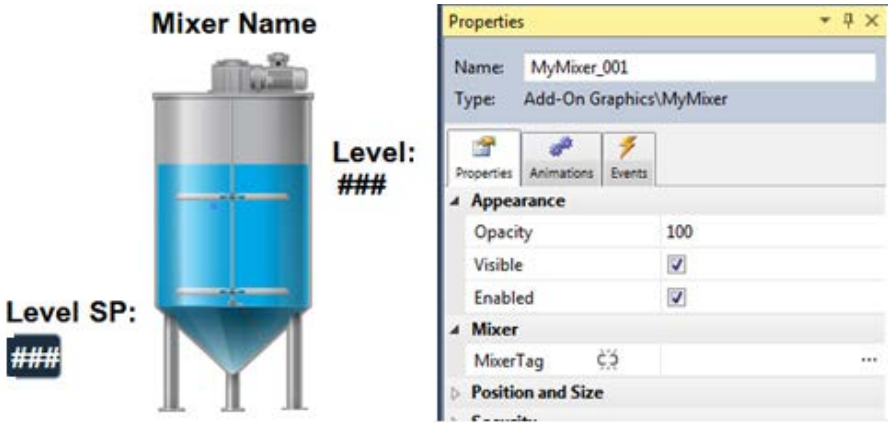
Create an instance for the Add-On Graphic definition

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.

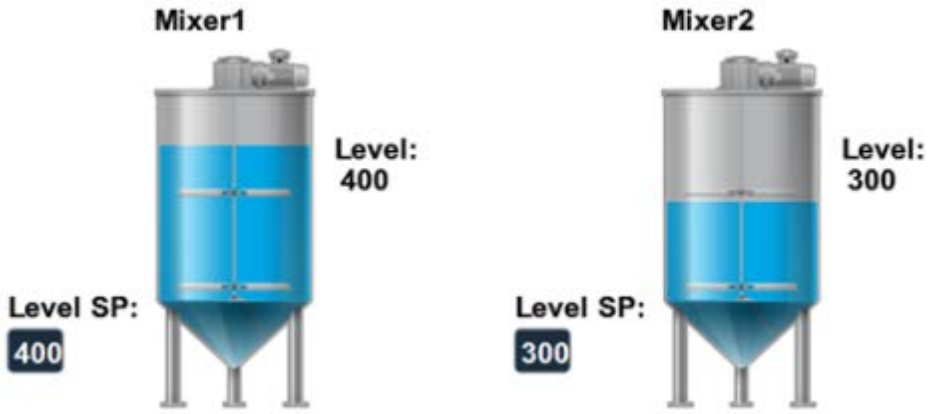


Click the ellipsis button to launch the tag browser and then bind the **MixerTag** property. The tag browser automatically filters by the **Mixer** data type and displays the tags in the controller of the type. This enables finding the Mixer1 instance of the data type by selecting **Mixer1**.

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 89

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 97

View Designer alarm guidelines on page 98

Configure the AlarmSummary table on page 100

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 98

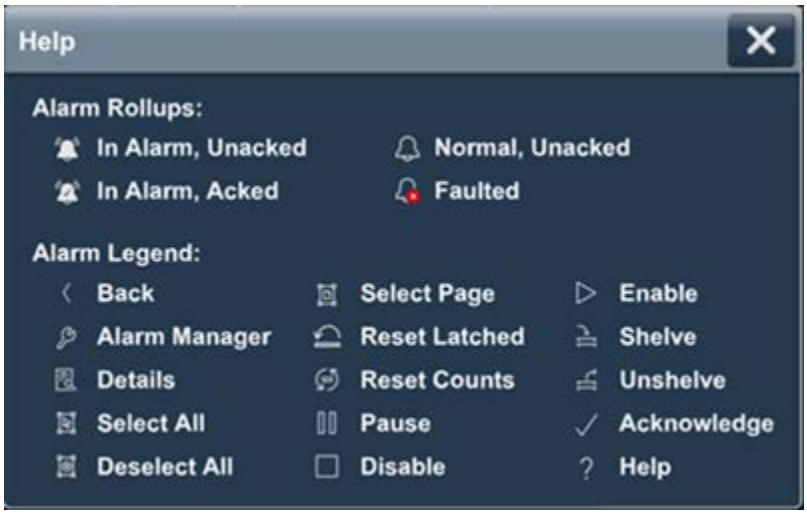
Configure the AlarmSummary table on page 100

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.

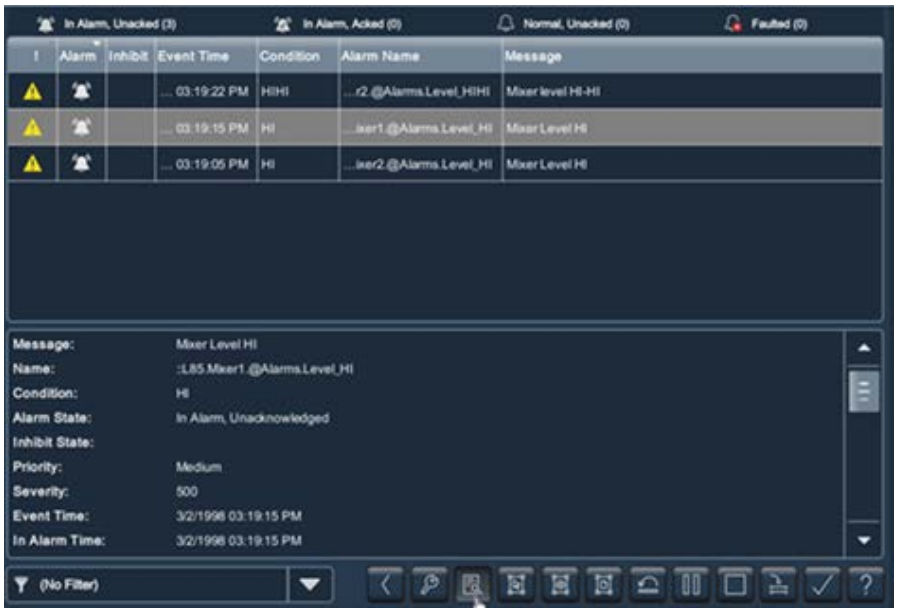
Pressing the alarm button navigates to the predefined Alarm Summary screen that displays all the active or unacknowledged alarms.

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.

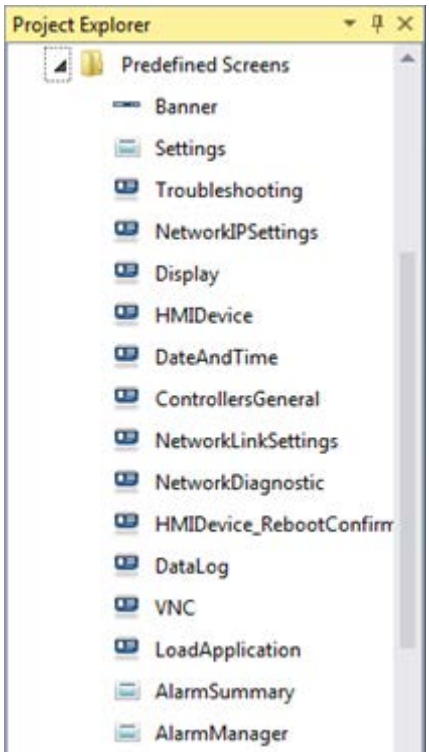


See also

Alarm Summary and Alarm Manager on page 100

Alarm Summary and Alarm Manager

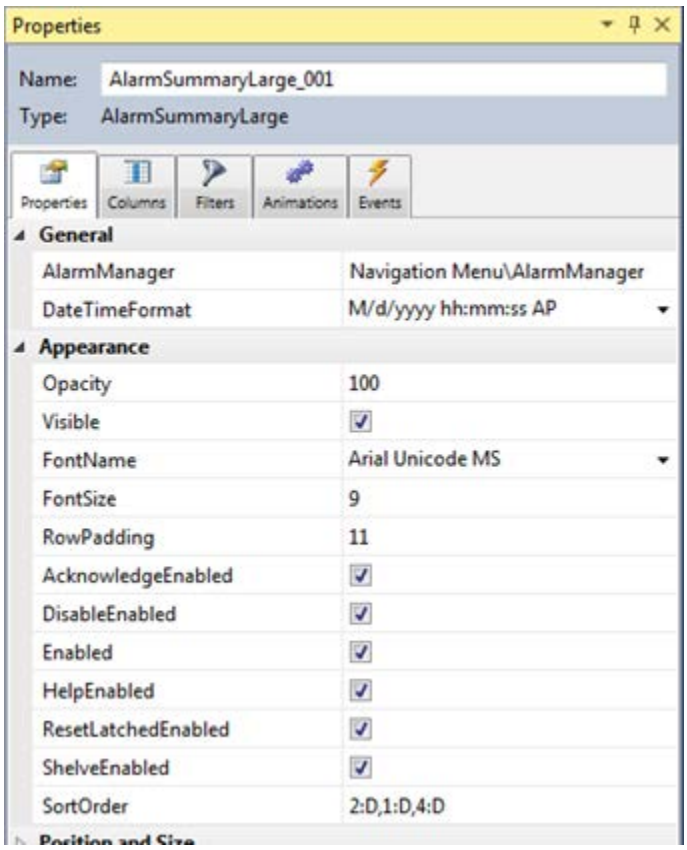
View Designer enables customizing the predefined Alarm Summary and Alarm Manager or creating new Alarm Summary and Alarm Manager screens. To customize the predefined screens, open the **AlarmSummary** or **AlarmManager** screens in the **Predefined Screens** folder in **Project Explorer**.



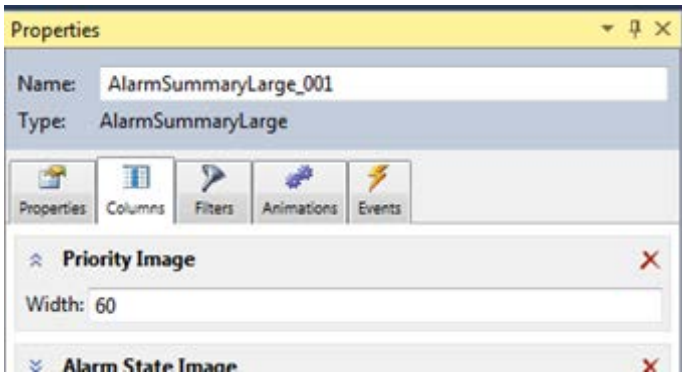
If opening the **AlarmSummary** screen, select the Alarm Summary table on the screen to configure the properties:

- **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.

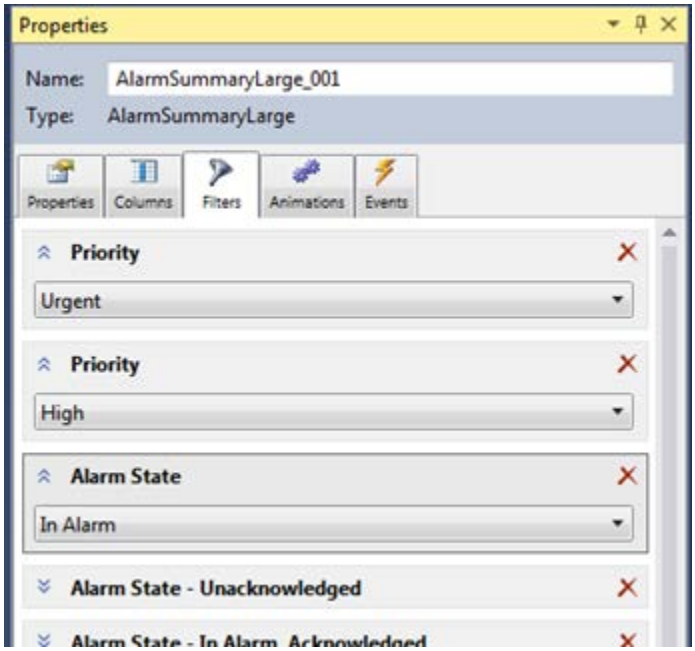
- **SortOrder**. Defines the sort order used when the alarm summary appears. Define the order by entering a valid column number, colon, and a D for descending or A for ascending, followed by a comma if adding a secondary or tertiary sort.



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.

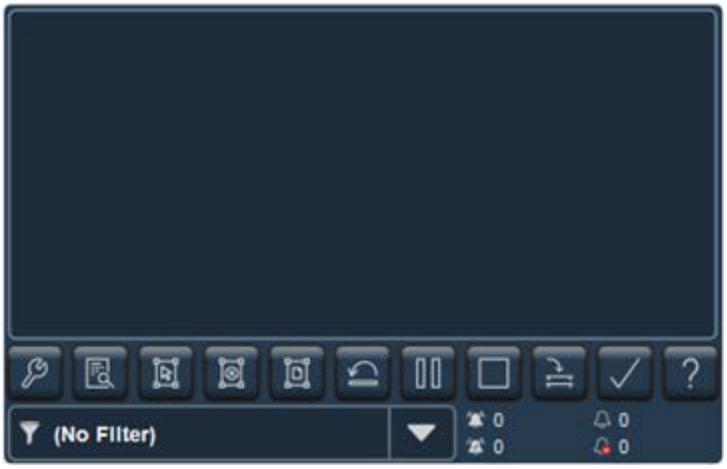
The **FilterIndex** property defines the default filter applied to the alarm table when the screen with the table opens. Bind to the **FilterIndex** property or create a custom **Alias** property aliased to the **FilterIndex** property of the Alarm Summary.

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 97

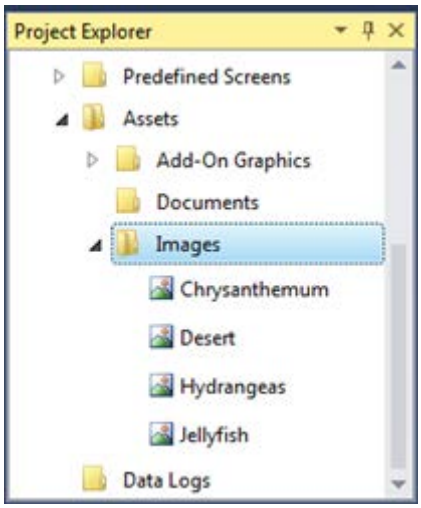
Use images

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

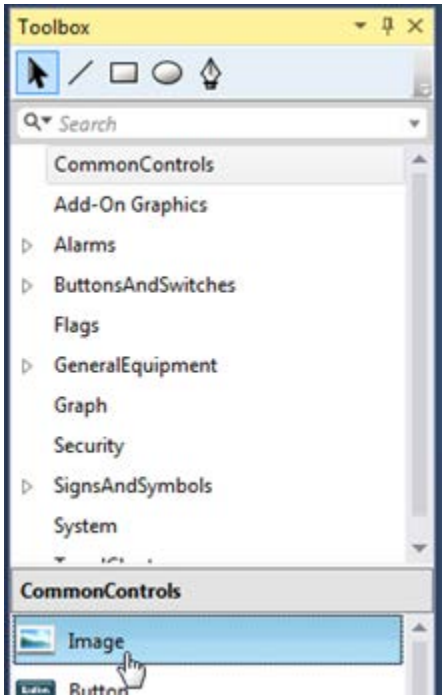
Tips:

- 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.
- 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.

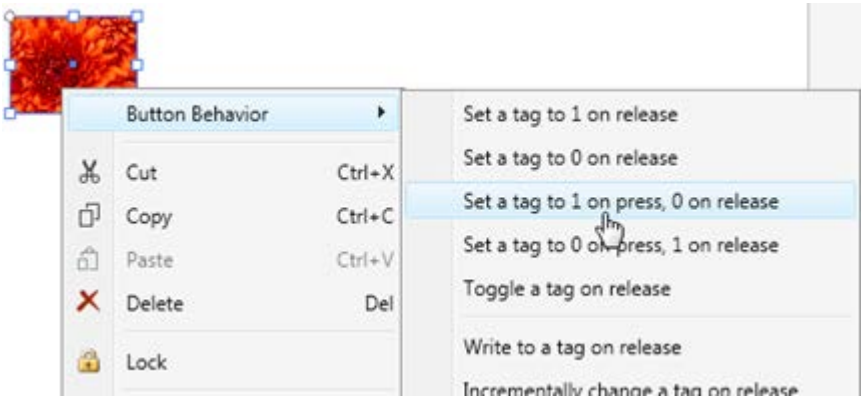
See also

Example 1: Configure the image to work as a momentary button on page 107

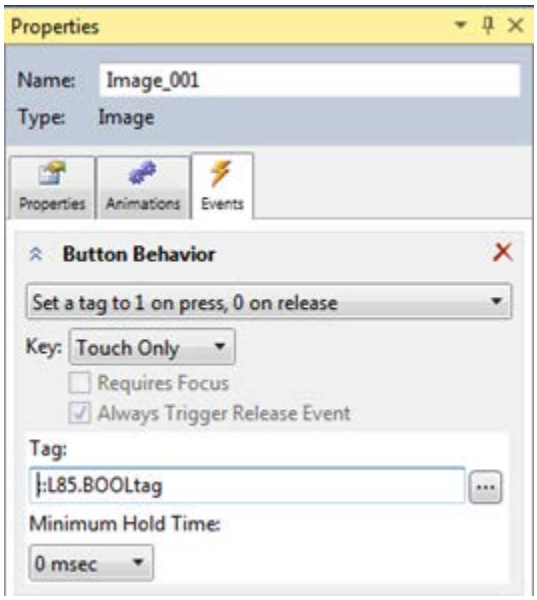
Example 2: Change images displayed in an image element at runtime on page 108

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 108

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

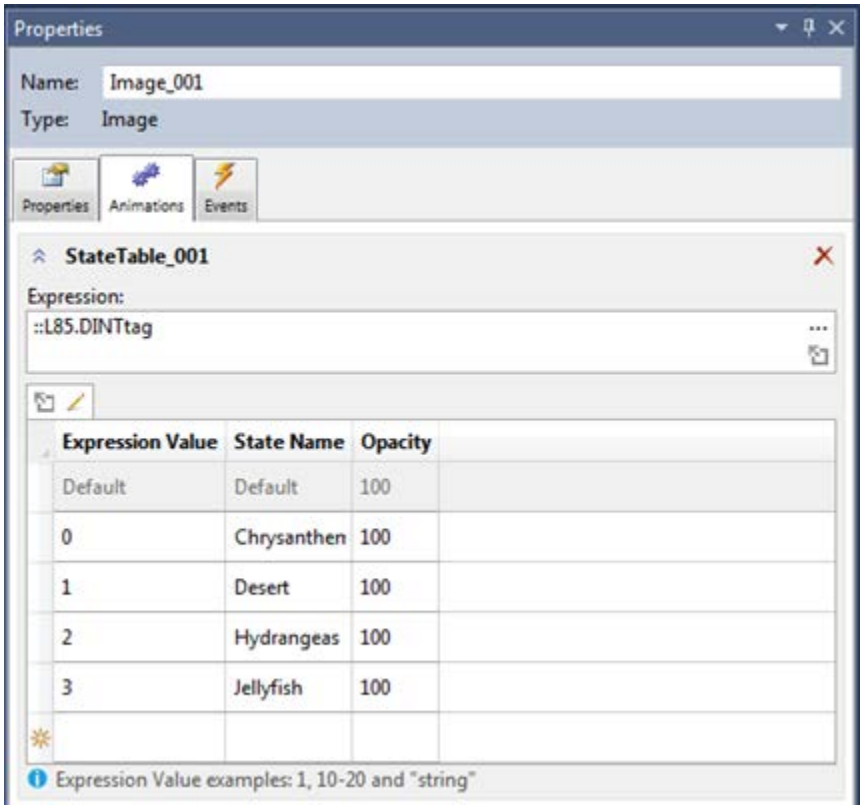
View Designer enables changing the images displayed in an Image element at runtime by either:

- 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.

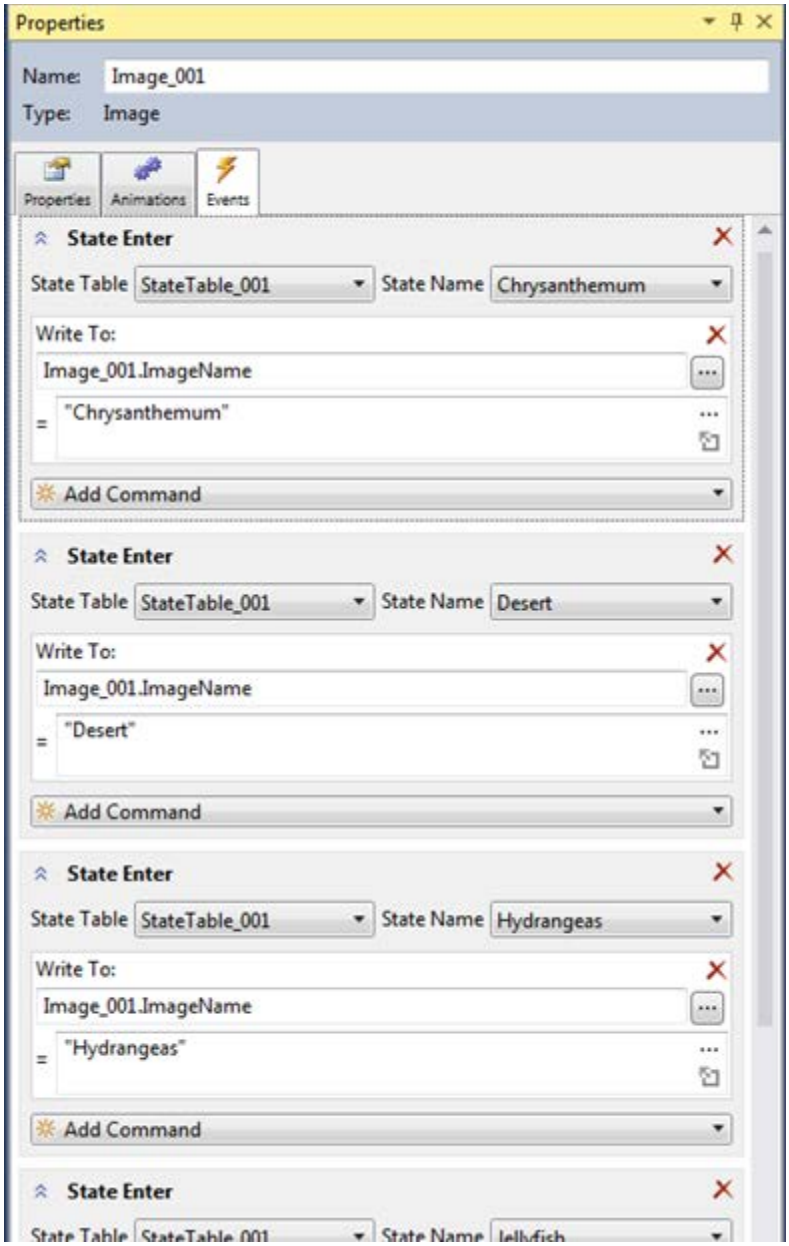
Create a state for the image using an expression of DINTtag and use four states to represent the different pictures to use.



The state table uses the **Opacity** property as the placeholder in this example since the value is the same in all states. The name of the states is the same as the image

files added to the Images folder in the Project Explorer. This helps keep track of the states.

The next step is to use State Enter commands to switch the images used by writing the image file name into the **ImageName** property of the Image element. Each state has a StateEnter event triggering a write command into the Image_001.ImageName property.



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

See also

Use images on page 105

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 111

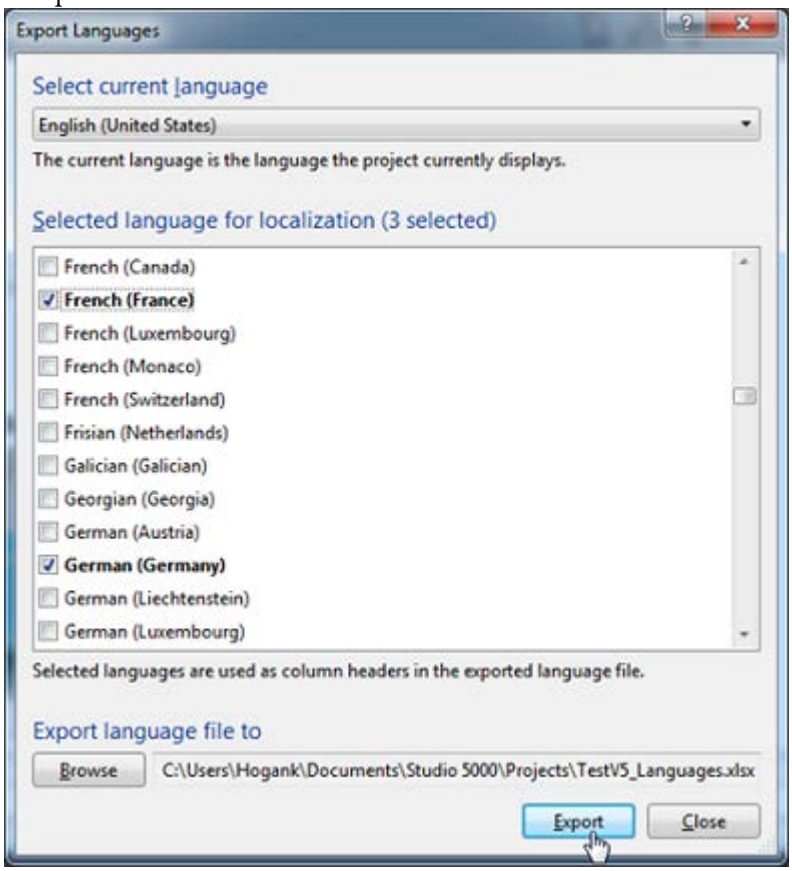
Configure language switching

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

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.

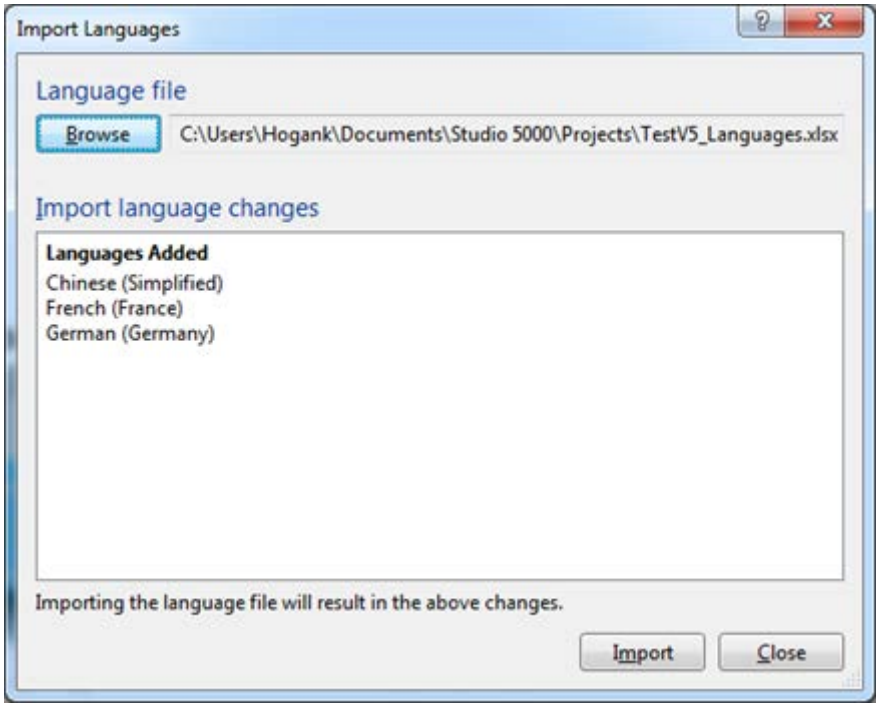
4. Determine the location to export the language spreadsheet to and select **Export**. The default languages spreadsheet xlsx file format is compatible with Microsoft Excel.

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.

7. 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.

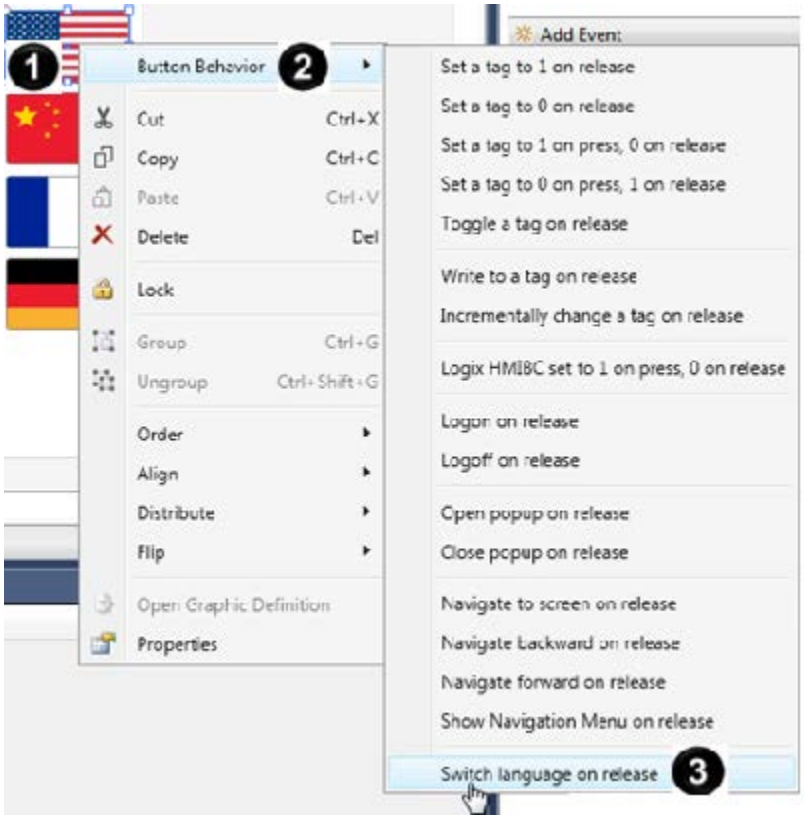


See also

Example: Configure language commands to switch languages on page 114

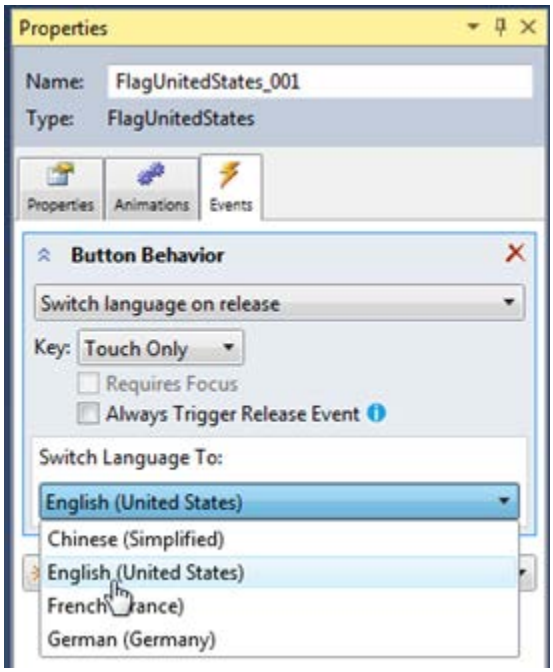
Example: Configure language commands to switch languages

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**.

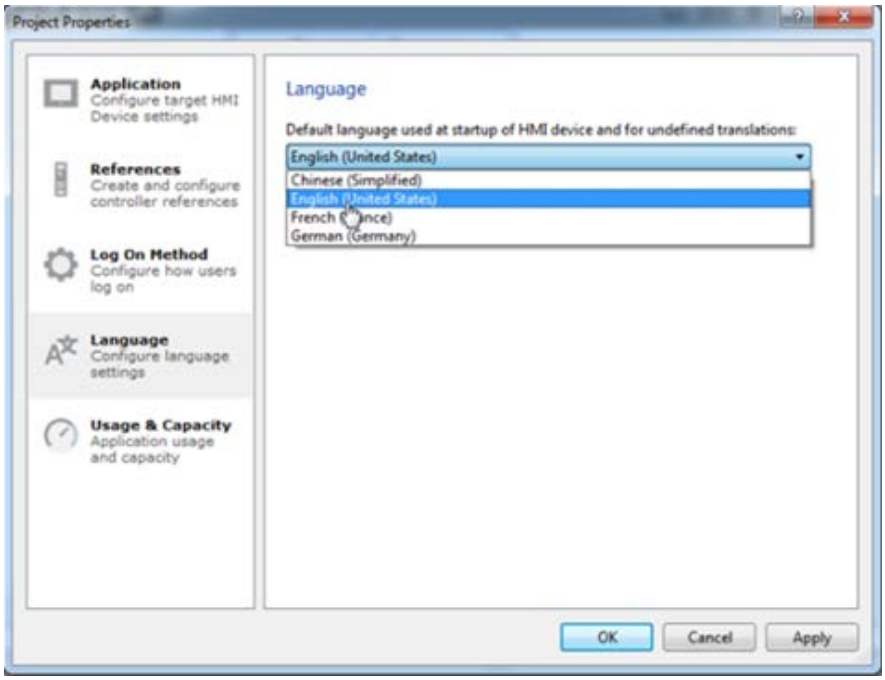


Item	Description
1	Flags
2	Button Behavior
3	Switch language on release

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.



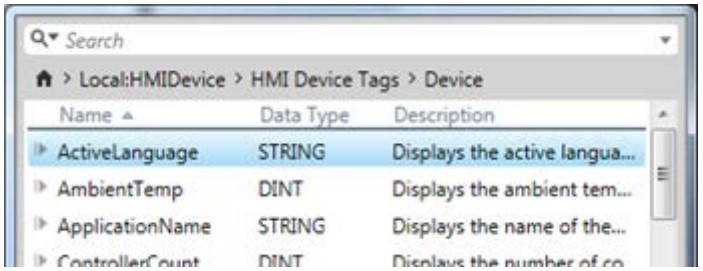
Define the default language for starting the PanelView 5000 HMI device and for the scenario of missing a language translation. On the **Project Properties** dialog box, select **Language**.



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.



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.

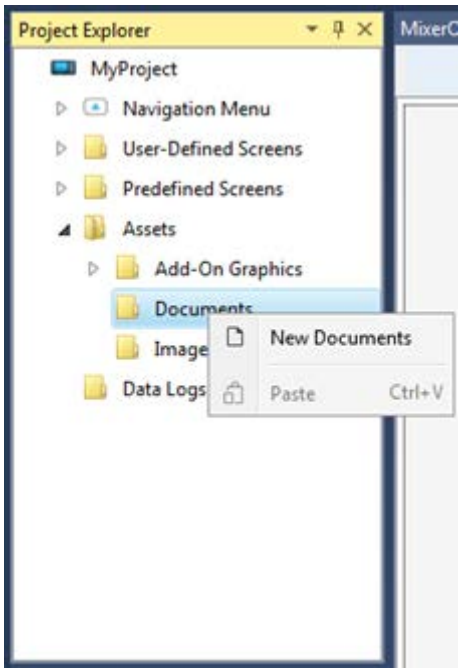
The Cyrillic character Д appears on the PanelView 5000 Text Display bound to STRINGtag.

See also

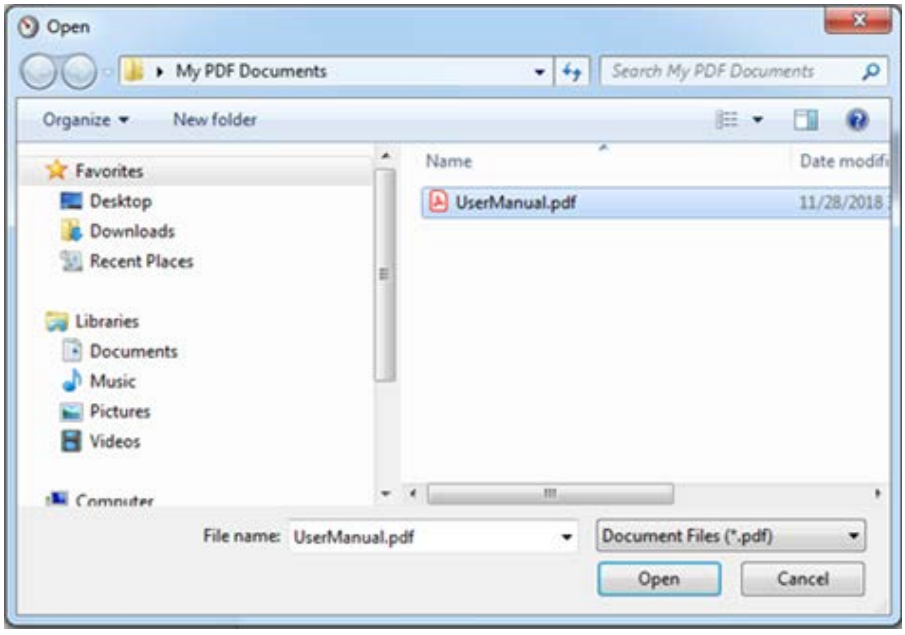
Language switching on page 111

PDF Viewer

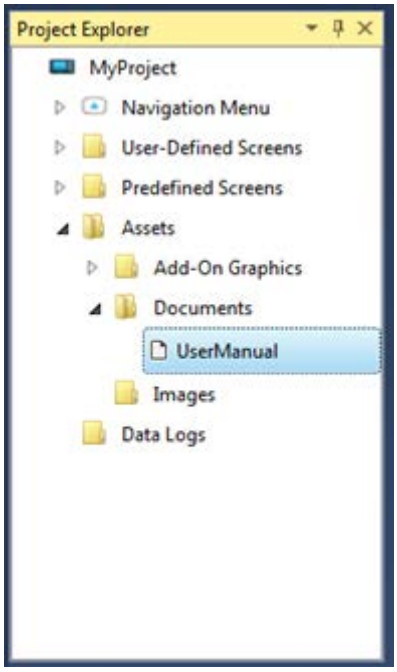
The PDF Viewer displays a PDF document on a PanelView 5000 HMI Device to view documentation, operating procedures, and machine setup instructions. First, add a PDF document to view on an HMI device in the **Assets > Documents** folder of the **Project Explorer**. In the **Project Explorer**, expand the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

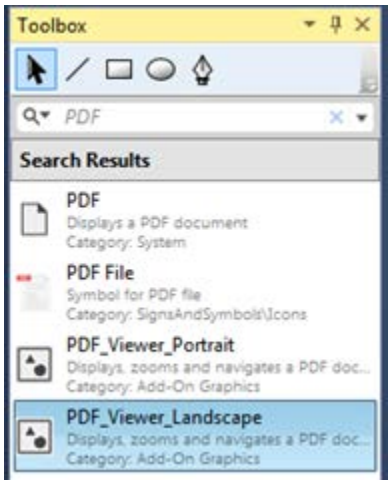
Example: Add a PDF Viewer Add-On Graphic to a screen on page 119

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

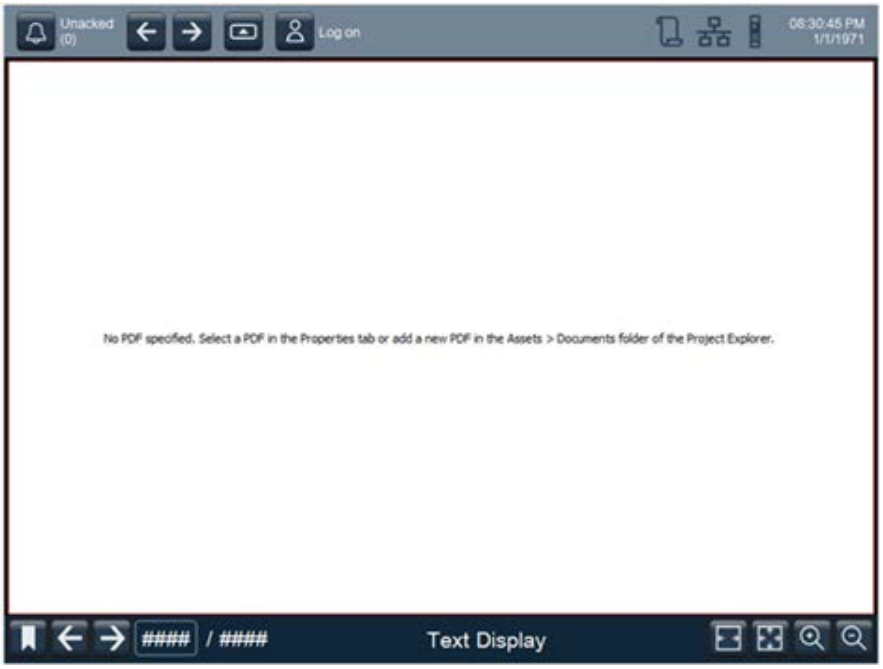
Add a PDF Viewer Add-On Graphic to a screen by searching for *PDF* in the **Toolbox**. Select a PDF Viewer Add-on Graphic that corresponds to the target HMI device screen size.

- 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**.

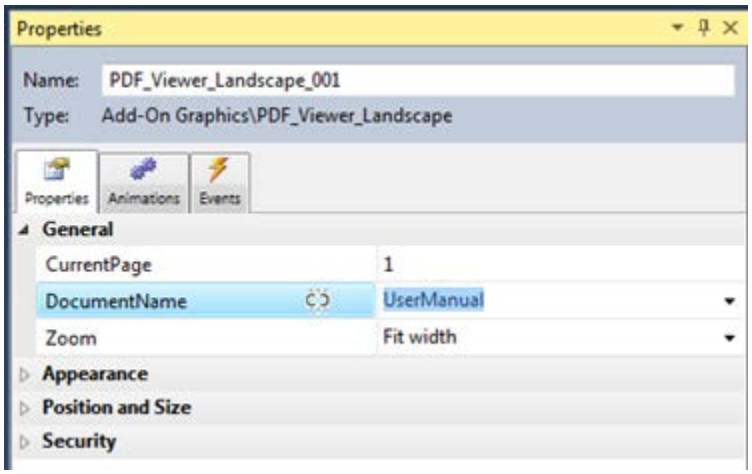


Tip: Expanding the **Assets > Add-On Graphics** folders and then dragging the PDF_Viewer Add-On Graphic or right-clicking the PDF_Viewer Add-On Graphic and selecting Add Add-on Graphic to (screen name) also adds a PDF Viewer Add-On Graphic to a screen or popup from Project Explorer.

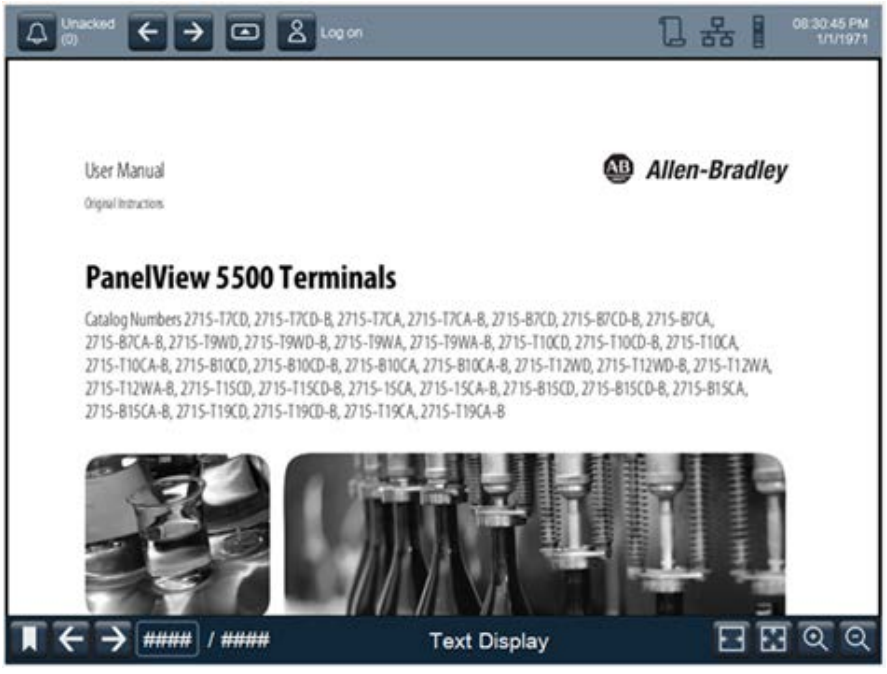
In the **Properties** window, expand the **General** category and configure the required properties:

- **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

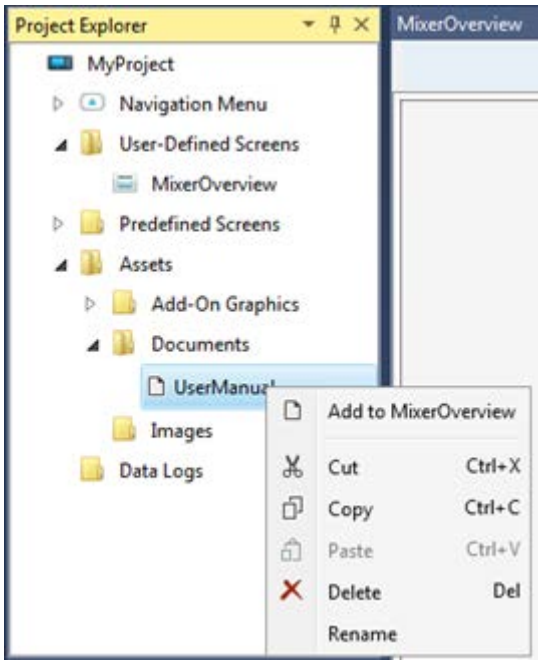
Example: Add a PDF Viewer graphic element to a screen on page 122

Example: Create custom PDF Viewer on page 124

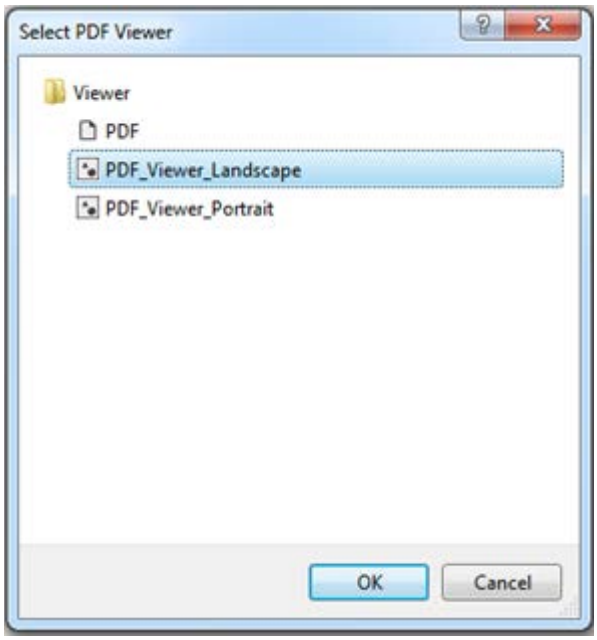
Example: Add buttons with PDF commands to screen on page 130

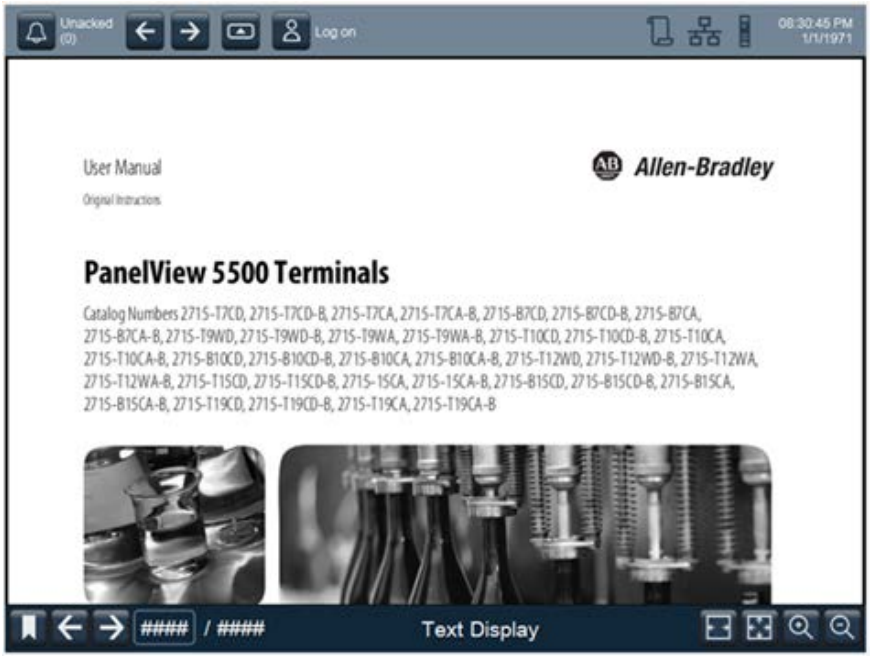
Example 2: Drag a PDF document onto the screen

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).

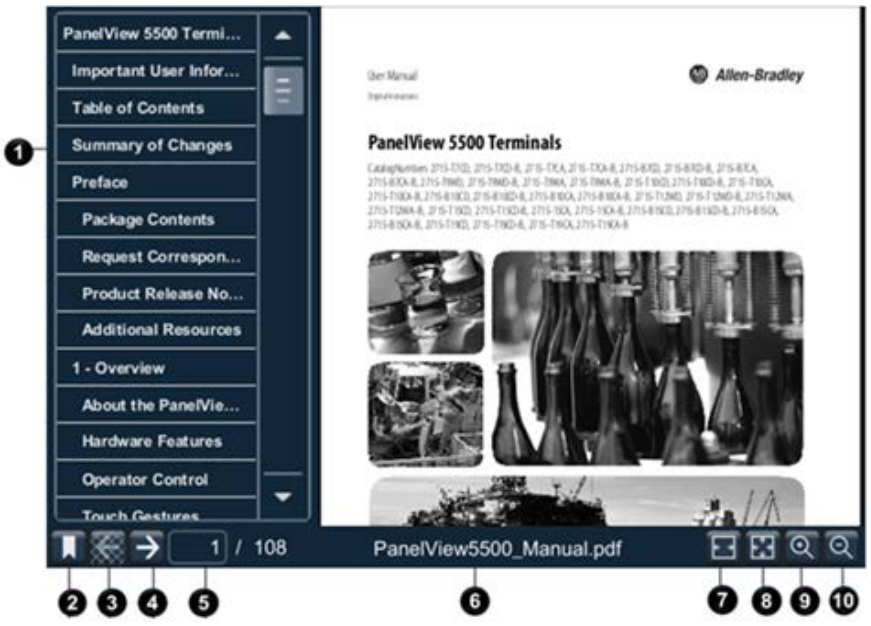


When the **Select PDF Viewer** dialog box appears, select the PDF Viewer element type. This example will use the PDF_Viewer_Landscape Add-On Graphic.





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.

Item	Description
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 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.

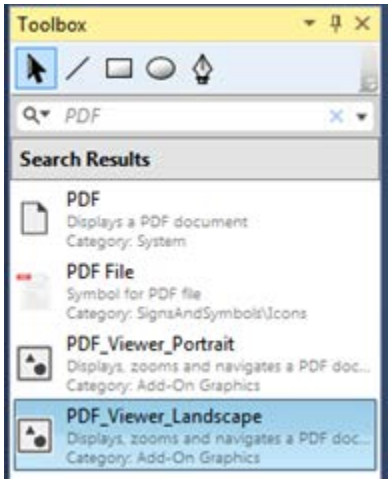
See also

Example: Create custom PDF Viewer on page 124

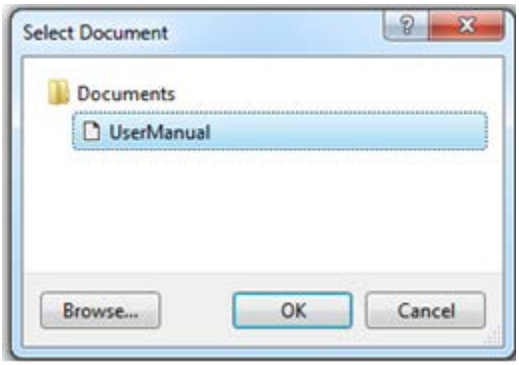
Example: Add buttons with PDF commands to screen on page 130

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.

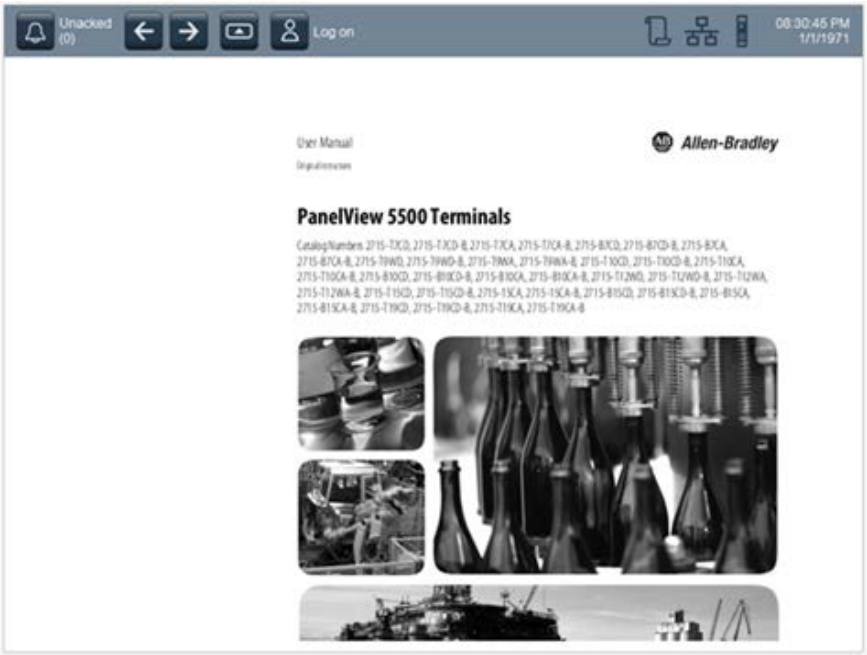


In the **Select Document** dialog box, select the PDF document to add to the screen, or select **Browse** to select a new PDF document.



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

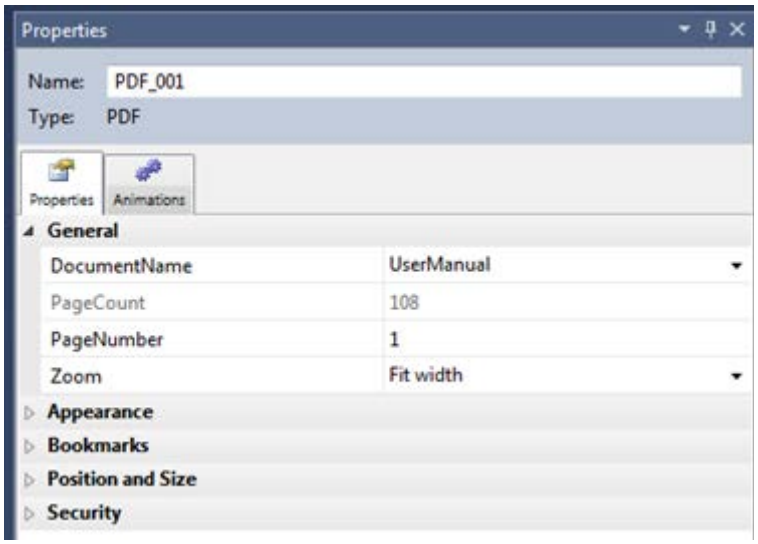
Next, size and position the PDF element on the screen as needed.



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.

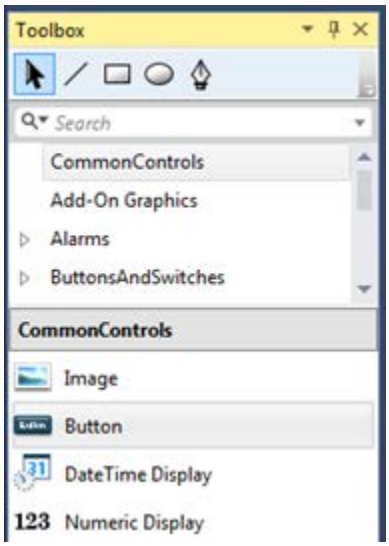
The PDF graphic element has touch support for navigating and panning pages within the referenced PDF document. To further customize a PDF graphic element, add buttons with event commands to navigate and zoom the PDF document on the HMI device. The following commands are available to use with the PDF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

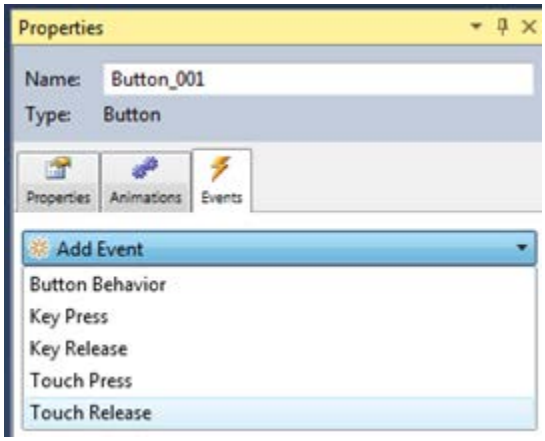
- Zoom Out. Decreases the document magnification by a percentage of the preset level in a PDF document. In the **ByPercent** box, enter a value to decrease magnification.
- Zoom To. Adjusts the magnification of the PDF to fit an entire page, fit the width of the page, or show the page at 100% in the 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.

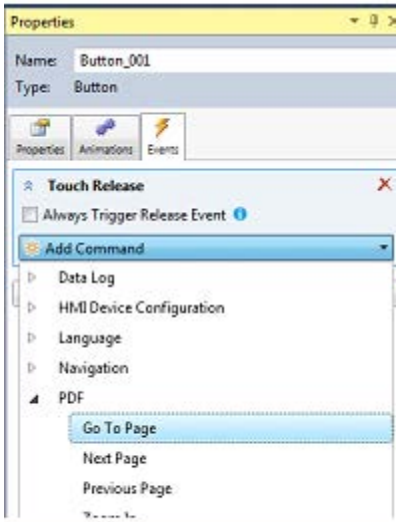
Add a button to the screen from the toolbox.

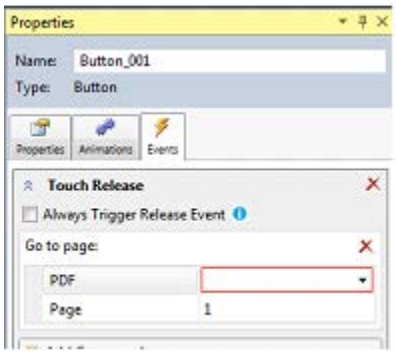


Next, access the **Events** tab in the Properties pane and add a **Touch Release** event to interact with the PDF Viewer graphic element.

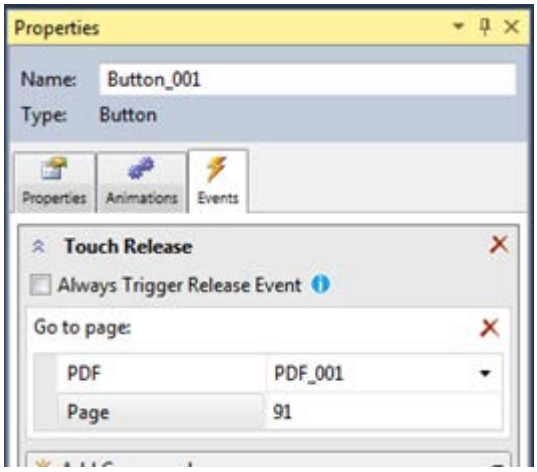


Next, add a **Go To Page** command to the event.

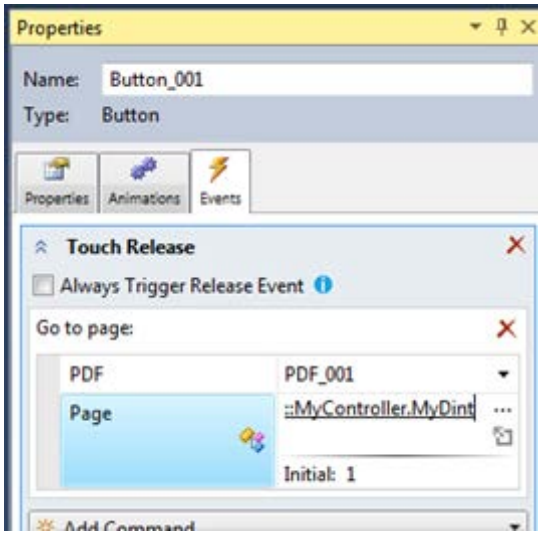




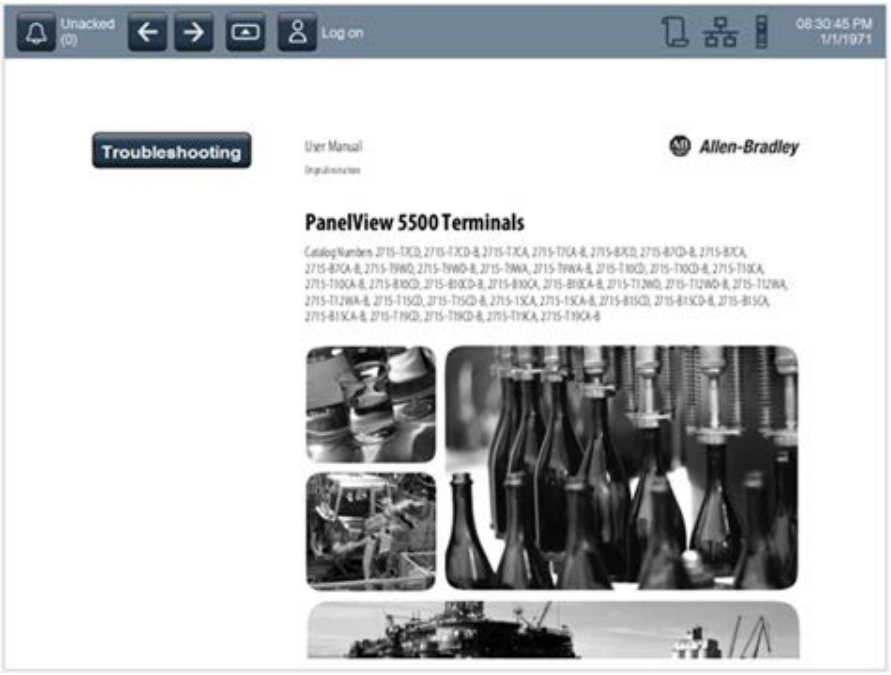
Select the PDF element to work with the command. Then enter the page number that opens when the command executes.



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.



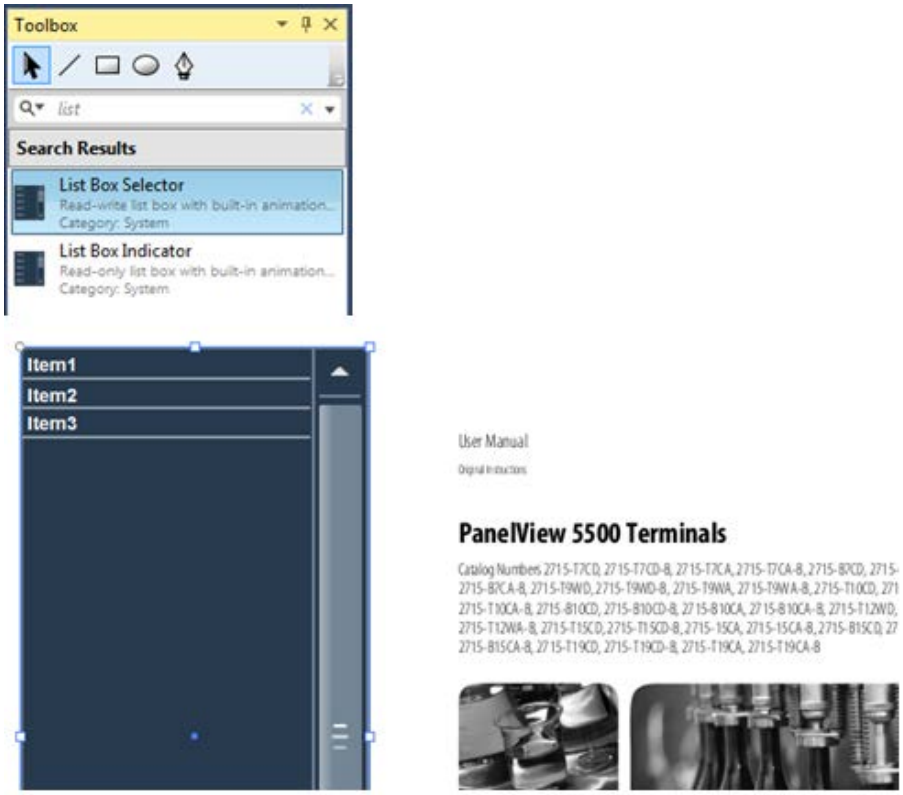
See also

Example: Add buttons with PDF commands to screen on page 130

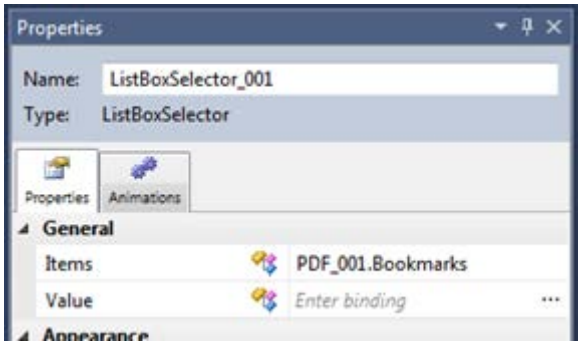
Bind bookmark properties of a PDF graphic element

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.

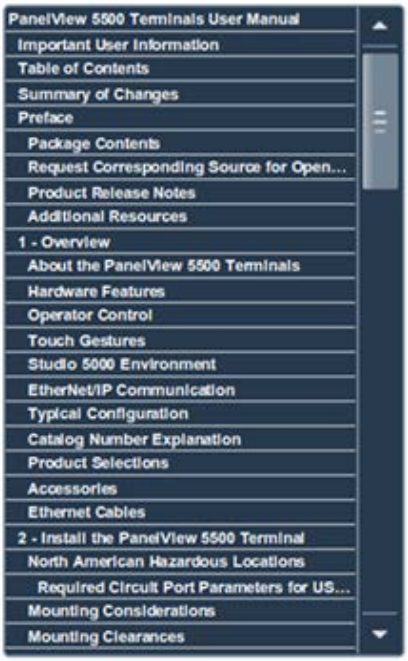


Next, use property-to-property binding to bind the **Items** property of the **List Box Selector** to the Bookmarks property of the PDF graphic element.



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 117

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 117

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 an 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 4GB 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

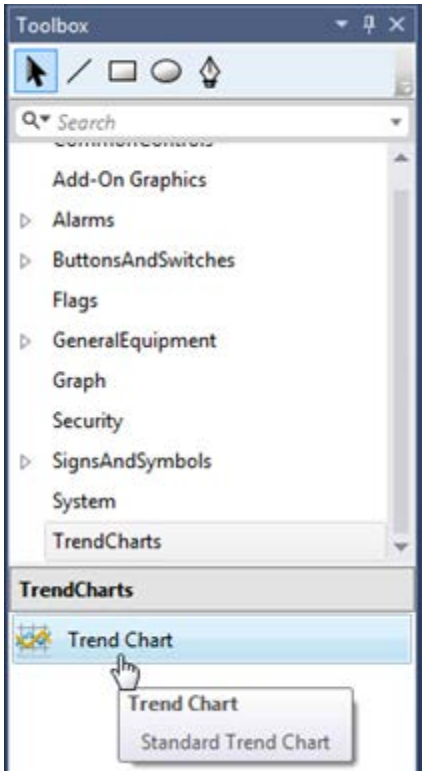
See also

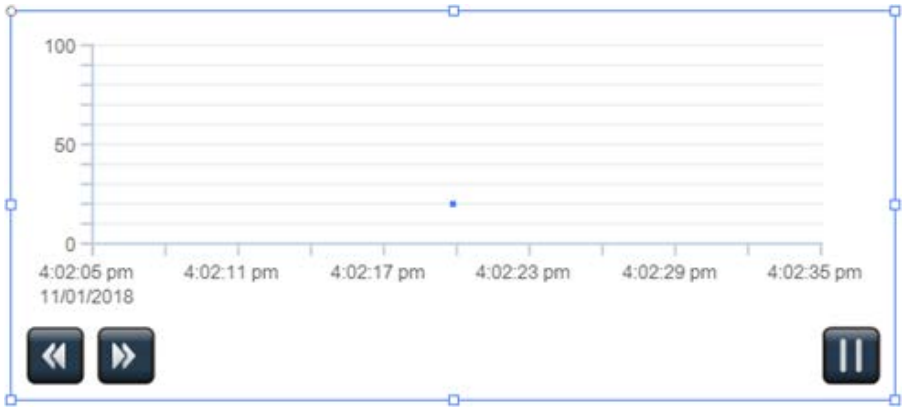
Display data on a trend chart on page 134

Log data on page 137

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.



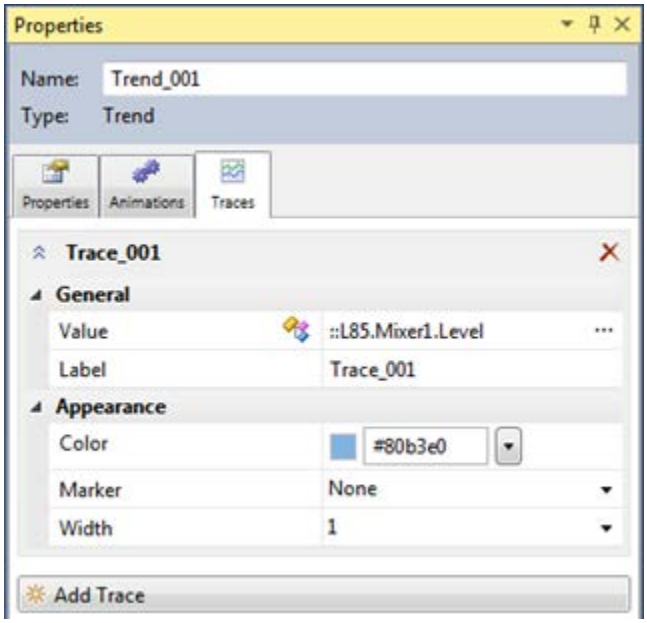


The **General** and **Appearance** properties that are unique to the Trend include:

- **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 500 milliseconds. This is the rate View Designer reads 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.

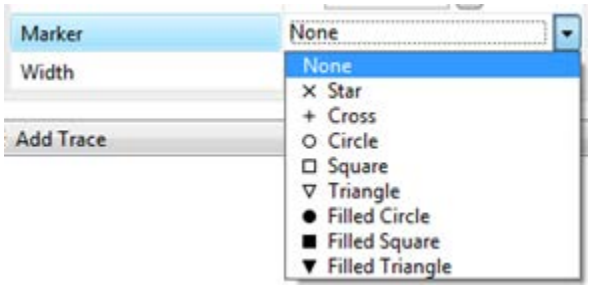
After configuring the **General** and **Appearance** properties, configure the traces displayed on the trend. A trace is the line or points on the trend to represent a changing tag value. Define up to 8 traces for a trend. Configure traces in the **Traces** tab on the **Properties** pane.



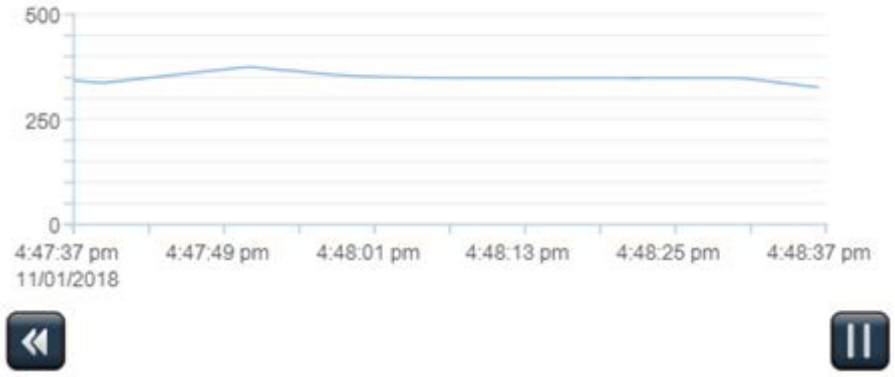
The **Trace** properties include:

- **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

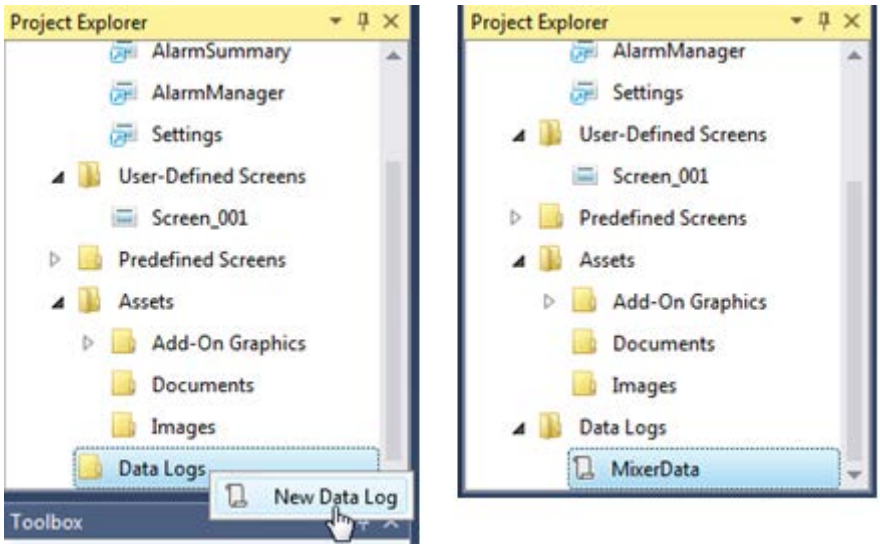
Log data on page 137

Log data

Data logging enables logging Logix tag values for longer term storage. To use data logging, insert at least a 4 gigabyte (GB) of RAM SD card into the PanelView 5000 HMI device. The PanelView 5000 HMI device supports SDHC SD cards that have up to 32 GB of RAM. 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.

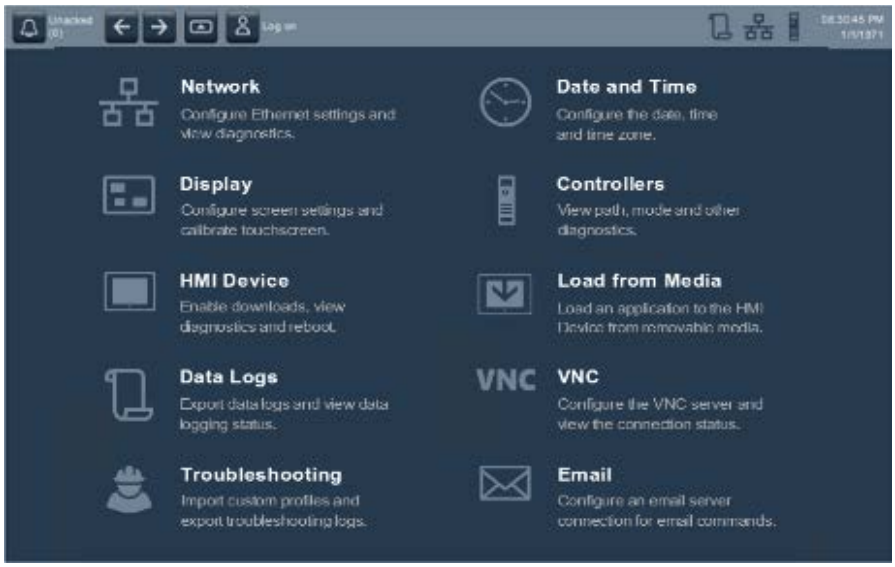
The Sample rate defines the logging rate for the tags in this data log. The Log duration stores samples for the specified time. Data logs are circular. After the Log duration, newer data starts overwriting the oldest data. Select the ellipsis button to open the tag browser to select a tag to log. Add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

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

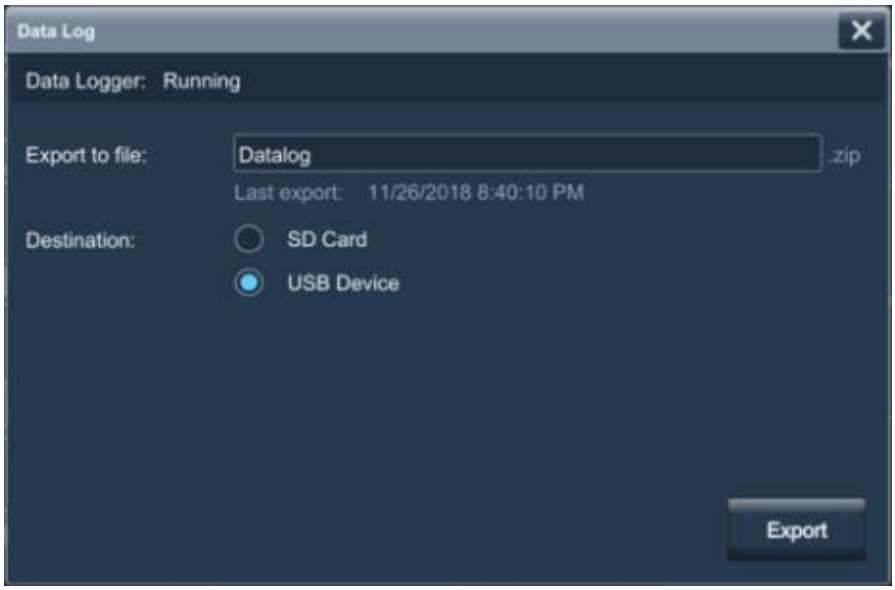
- The number of tags in the current log
- The number of tags across all the logs
- The percentage of the total number of allowable records used across all the logs.

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 Logs**.



On the **Data Logs** 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 export** update appears with the date and time of the latest export.

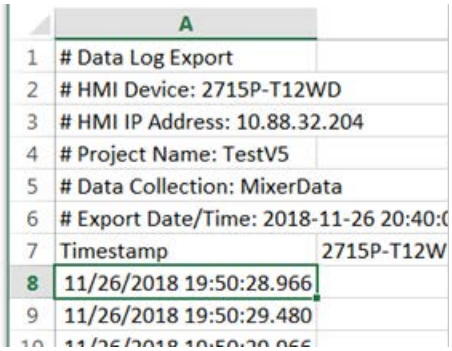


Remove the media and insert it into a computer to view the data log contents. Unzip the file to view the .csv files for the individual data logs. The .csv files use the same name as the name of the data log.

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

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 20:40:0	
7	Timestamp	2715P-T12W
8	11/26/2018 19:50:28.966	
9	11/26/2018 19:50:29.480	

See also

Log data and trend charts on page 133

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 141

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 141

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 on page 143

VNC popup on page 144

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 144

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 143

Configure VNC access on page 143

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 141

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.

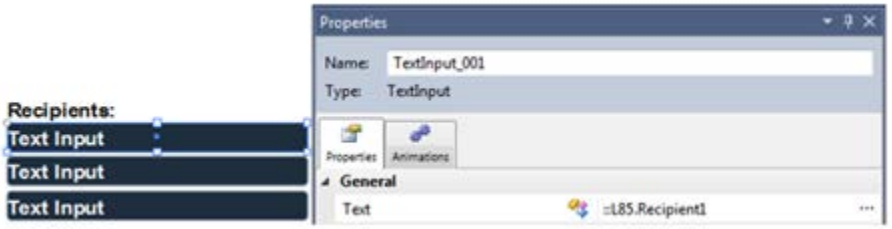
Emails as texts

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 555-123-4567@vtext.com as the email recipient. Verify the correct address format with the cell phone provider.

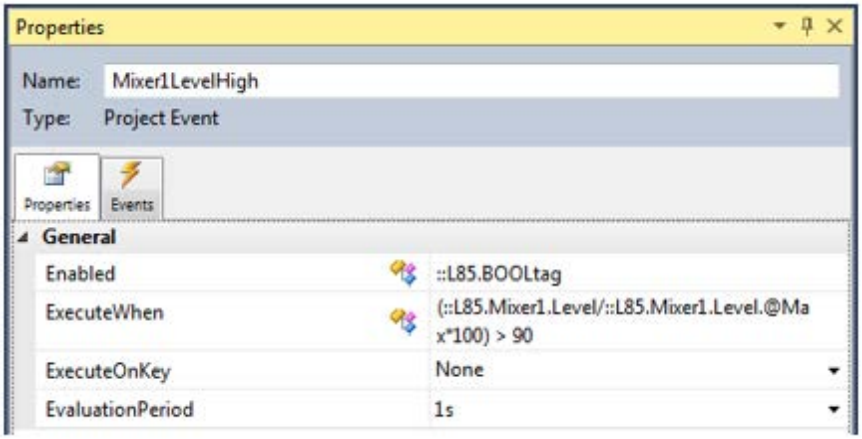
Example: Configure a list of email recipients online and send them emails

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.

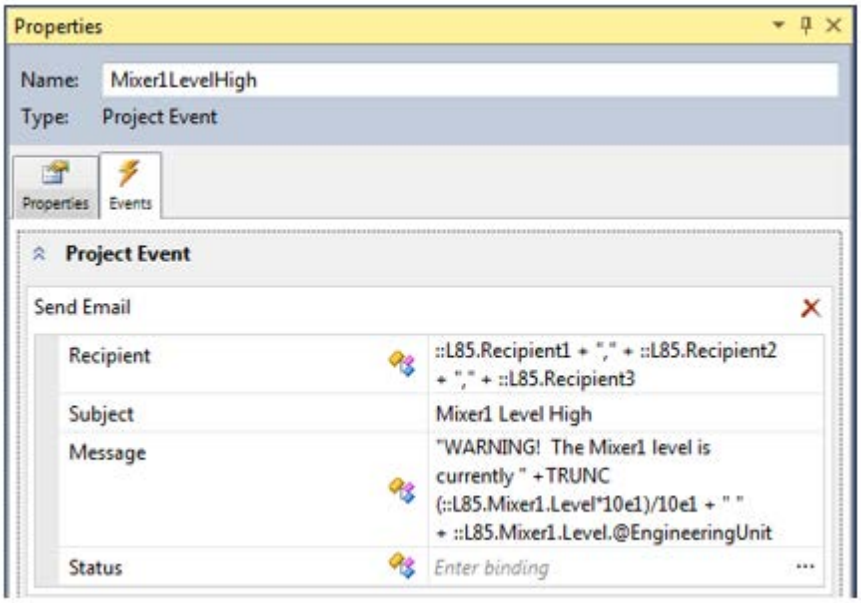
To configure the recipients from the PanelView 5000, create an email configuration screen where the user enters recipients. On the screen, use **Text Inputs** to write the recipients to string tags in the Logix controller. The string tags use **::L85.Recipient1**, **::L85.Recipient2**, and **::L85.Recipient3** 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.

A

C

D

E

G

I

L

N

P

R

S

T

V

Rockwell Automation support

Use the following resources to access support information.

Technical Support Center	Knowledgebase Articles, How-to Videos, FAQs, Chat, User Forums, and Product Notification Updates	https://rockwellautomation.custhelp.com
Local Technical Support Phone Numbers	Locate the phone number for your country.	http://www.rockwellautomation.com/global/support/get-support-now.page
Direct Dial Codes	Find the Direct Dial Code for your product. Use the code to route your call directly to a technical support engineer.	http://www.rockwellautomation.com/global/support/direct-dial.page
Literature Library	Installation Instructions, Manuals, Brochures, and Technical Data.	http://www.rockwellautomation.com/global/literature-library/overview.page
Product Compatibility and Download Center (PCDC)	Get help determining how products interact, check features and capabilities, and find associated firmware.	http://www.rockwellautomation.com/global/support/pcdc.page

Documentation feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete the How Are We Doing? form at http://literature.rockwellautomation.com/idc/groups/literature/documents/du/ra-du002_-en-e.pdf.

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

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Rockwell Automation Publication 9324-GR001B-EN-P - September 2019

Supersedes Publication 9324-GR001A-EN-P - September 2019