2025

FactoryTalk Optix Platform

My First Project

Rockwell Automation

1/1/2025

Important User Information

This documentation, whether, illustrative, printed, "online" or electronic (hereinafter "Documentation") is intended for use only as a learning aid when using Rockwell Automation approved demonstration hardware, software and firmware. The Documentation should only be used as a learning tool by qualified professionals.

The variety of uses for the hardware, software and firmware (hereinafter "Products") described in this Documentation, mandates that those responsible for the application and use of those Products must satisfy themselves that all necessary steps have been taken to ensure that each application and actual use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards in addition to any applicable technical documents.

In no event will Rockwell Automation, Inc., or any of its affiliate or subsidiary companies (hereinafter "Rockwell Automation") be responsible or liable for any indirect or consequential damages resulting from the use or application of the Products described in this Documentation. Rockwell Automation does not assume responsibility or liability for damages of any kind based on the alleged use of, or reliance on, this Documentation.

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

Except as specifically agreed in writing as part of a maintenance or support contract, equipment users are responsible for:

- properly using, calibrating, operating, monitoring and maintaining all Products consistent with all Rockwell Automation or third-party provided instructions, warnings, recommendations and documentation.
- ensuring that only properly trained personnel use, operate and maintain the Products at all times,
- staying informed of all Product updates and alerts and implementing all updates and fixes; and
- all other factors affecting the Products that are outside of the direct control of Rockwell Automation.

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

Throughout this manual we use the following 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).

Table of Contents

© 20234 Rockwell Automation, Inc. All rights reserved.

© 2025 Rockwell Automation, Inc. All rights reserved.

Introduction to the New FactoryTalk® Optix™ Visualization Platform Hands-on Lab

Before You Begin

About this Lab

This is an introductory Lab that explores the features and functionality of FactoryTalk Optix.

In the first section of the lab, you will be creating a brand-new project to get hands-on experience with fundamental concepts and configure the framework for the rest of the lab. The remaining sections are independent of other lab sections so you may select what you wish to learn about in any order.

The lab sections contain an explanation and feature details for that particular section in a "Tip" section. It is not necessary to read these explanations, but a more thorough understanding of the features presented will be attained if time is taken to read through them.
In this lab, you will:

- Create your first project and configure the main window
- Configure panel and screen types
- Configure navigation

Select from the following:

- Configure alarms and alarm history
- Create security users and groups
- Configure a data logger, and display data in a Data Grid and Trend
- Configure Reports
- Configure Language switching
- Configure Recipes
- Create a Web Presentation Engine
- Develop a User Interface (UI) with scaled layout and responsive graphics

Other topics are also available as part of the appendences at the end of the lab.

© 2025 Rockwell Automation, Inc. All rights reserved.

Duration

Create Your First Project – 30 minutes

Web Presentation Engine - 15 minutes

Alarming – 15 minutes

Configure a Datalogger - 15 minutes

Reporting - 20 minutes

Configure Recipes - 15 minutes

Language Switching - 10 minutes

Security – 20 minutes

Aliases – 15 minutes

Prerequisites

There are no prerequisites required for this lab.

Materials

This lab requires the following items.

- FactoryTalk Optix Studio version 1.5 or newer

© 2025 Rockwell Automation, Inc. All rights reserved.

Introduction

Objectives

- Explain "What is FactoryTalk Optix"
- Understand the FactoryTalk Optix components.
- Learn about the Design environment.

What is FactoryTalk Optix?

FactoryTalk Optix is a new platform for creating Human-Machine Interfaces (HMI) developed by Rockwell Automation. It is an addition to the FactoryTalk suite of industrial automation software and is designed to provide an open and flexible platform for creating custom HMI applications that are tailored to the user-specific needs.

With FactoryTalk Optix, users can create:

- **Real-time monitoring**: FactoryTalk Optix provides real-time information about the status of industrial systems and processes, allowing operators to make informed decisions and quickly respond to issues.
- **Data visualization**: FactoryTalk Optix provides a wide range of visualization tools and features, including graphs, charts, and histograms, to help operators better understand and analyze data.
- **User-friendly interface**: FactoryTalk Optix features a modern and intuitive interface that is designed to be easy to use, even for non-technical users.
- **Customization**: FactoryTalk Optix provides a wide range of customization options, including the ability to create custom screens, dashboards, and reports, to meet specific customer requirements.
- **Connectivity**: FactoryTalk Optix is designed to connect to a variety of industrial systems and devices, including Rockwell Automation controllers, and third-party PLCs and automation devices.
- **Scalability**: FactoryTalk Optix is designed to scale from small to large-scale applications with multiple web clients.
- **Extensibility**: OPC UA is core to the platform with support of the OPC UA Companion Specs. Native IoT connectivity and a C# engine to fulfill every customer need.

© 2025 Rockwell Automation, Inc. All rights reserved.

FactoryTalk Optix consists of different software components:

FactoryTalk Optix Studio:

Integrated development environment with a framework of functional modules for designing and compiling HMI or Internet of Things (IoT) applications. FactoryTalk Optix Studio includes a library of predefined objects that support the modular design of graphical interfaces, features, and logic operations of an HMI application. By using specific C# language scripts, you can automate various actions in the design phase and add customized functions to projects. FactoryTalk Optix Studio comes in either:

- FactoryTalk Optix Studio Standard
  - This is available at no-cost to anyone without a license.
- FactoryTalk Optix Studio Pro
  - This is a subscription that enables cloud and collaboration functionality.

FactoryTalk Optix Application:

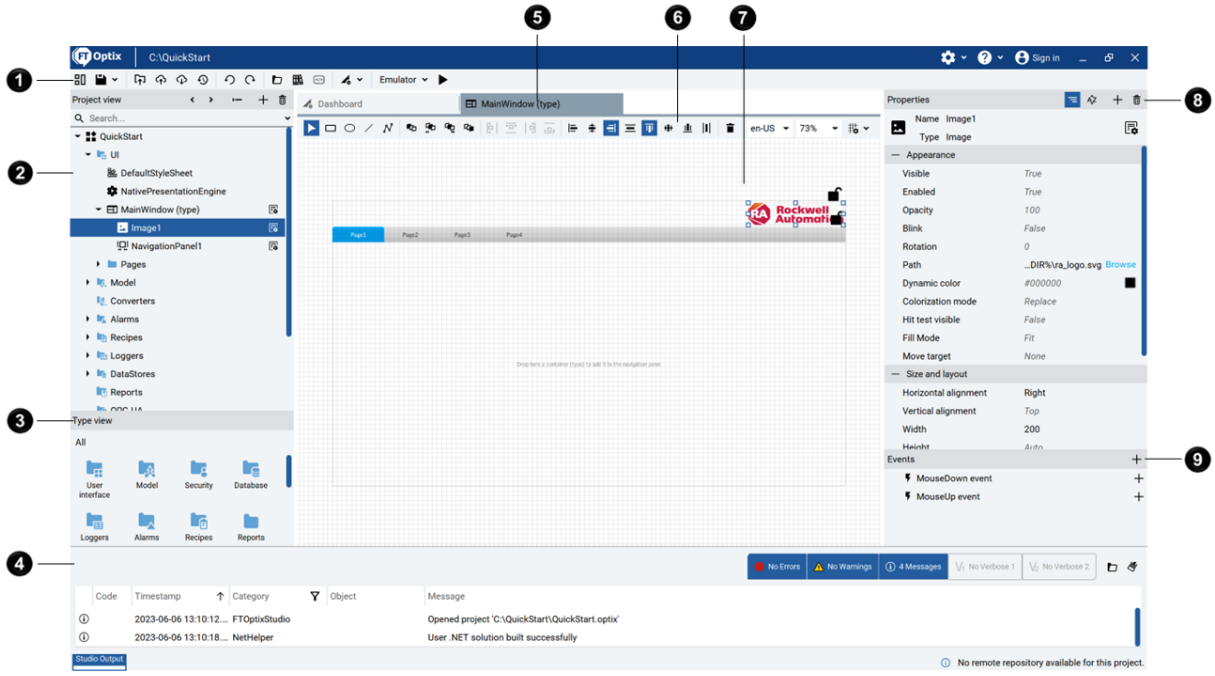
An HMI or IoT application developed and compiled in FactoryTalk Optix Studio.

FactoryTalk Optix Runtime:

Runtime software that is deployed to FactoryTalk Optix Application client systems. The FactoryTalk Optix Runtime installation package contains FactoryTalk Optix Application Update Service (Software for updating and deploying FactoryTalk Optix Applications from FactoryTalk Optix Studio to the client devices) and FactoryTalk Optix License Manager.

FactoryTalk Optix Studio Interface

The Design environment



© 2025 Rockwell Automation, Inc. All rights reserved.

1 Main toolbar

Commands that apply to all portions of the user interface.

2 Project view

Project information model that displays the content and structure of the project in nodes. A node can be a parent that has child nodes.

3 Type view

Object types and variable types on which instances of objects are based. Predefined types are grouped in folders according to their purpose. Custom types are grouped in folders that reflect the structure in **Project view**.

4 Output pane

Messages related to the operation of the FactoryTalk Optix Application. Applications that are running and are connected to FactoryTalk Optix Studio appear on **Emulator Output**.

5 Tabs

Tabs of open objects being modified in the editor. If you Mouse over a tab to select ✕ without saving the project, you are prompted to save your changes.

6 Editor toolbar

Commands that apply to the object type being modified in the editor.

7 Editor

Graphic editor for objects. Each editor opens in a new tab and is used for graphic objects and object types, such as tag importers and recipes.

8 Properties pane

Set the properties for the selected node in **Project view** or in the object editor.

9 Events pane

Associate methods with events generated by the currently selected node.

© 2025 Rockwell Automation, Inc. All rights reserved.

Languages supported in the design environment

New! Starting with FactoryTalk Optix 1.5, the following languages are supported at design-time

- English (United States)
- Italian (Italy)
- Chinese (China)
- French (France)
- German (Germany)
- Japanese (Japan)
- Korean (South Korea)
- Portuguese (Brazil)
- Spanish (Spain)

There are 3 ways to change the interface language:

- **Option 1**: Select the FactoryTalk Optix Studio Options icon located closer to the right side of the title bar. For Language and Locale, select the desired supported language.
- **Option 2**: Have a different user with a different associated locale language log on to your computer.
- **Option 3**: In Project view, select the project node and then in Properties, expand Localization and in Locales, select the desired supported language.

Note: FactoryTalk Optix Studio allows you to develop in 9 different languages. At runtime, a multilingual project uses the configured session locale to display the interface. You can select from approximately 250 locales and languages.

Supporting multiple versions of the design environment on the same computer

New! starting with FactoryTalk Optix 1.4, side by side installations of different version of the design environment is supported.

© 2025 Rockwell Automation, Inc. All rights reserved.

Create Your First Project (30 Minutes)

Objectives

- Create a new project
- Configure the main window
- Produce panel types and a navigation panel
- Add graphic objects and variables
- Set up Dynamic Links and Complex Dynamic Links using Key-value converters
- Explore responsive graphics
- Appendix A - Establish communications

Scenario

In this section of the lab, you will be introduced to some of the core aspects of basic HMI development using FactoryTalk Optix Studio. You will be starting from scratch to create a new project, where you will be guided in:

- Exploring the user interface (UI) responsive graphics capabilities of FactoryTalk Optix
- Creating some simple displays with easily configured navigation.

Along the way you will also be introduced some more optional advanced features and concepts such as:

- Object-oriented programming utilizing Types
- Complex Dynamic Linking
- Key-Value Converters

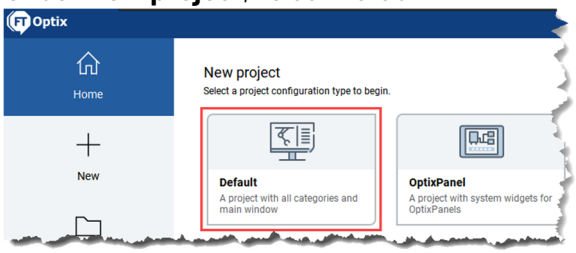
To learn more about setting up communications to both a Logix controller and an OPC UA Server, please check Appendix D.

© 2025 Rockwell Automation, Inc. All rights reserved.

Lab Procedure

1. Open FactoryTalk Optix Studio by clicking on the **FactoryTalk Optix Studio** icon from the **Windows Start** menu.

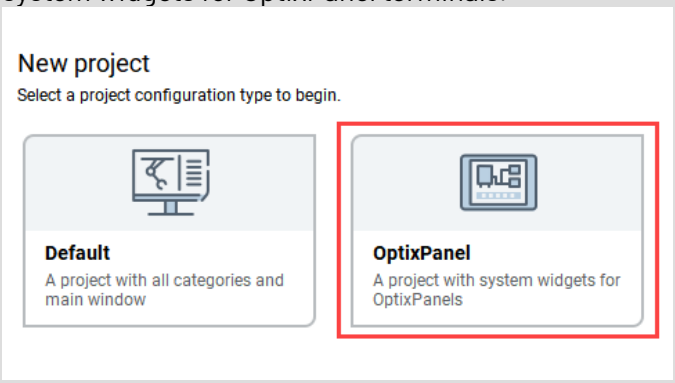
2. Under **New project**, select **Default**.



ADDITIONAL PROJECT OPTION

OptixPanel™ graphic terminals give you a PC-like user experience in an HMI appliance dedicated for Optix Applications. There is no operating system to secure and smaller applications can benefit from the superior price-performance ratio. These graphic terminals are also available in a wide range of screen sizes, bezel options, aspect ratios and touch screen technologies that support gestures, such as swipe and pinch, for easier integration on your factory floor.

OptixPanel is the 2nd option available. Select this option if you would like to create a project with system widgets for OptixPanel terminals.



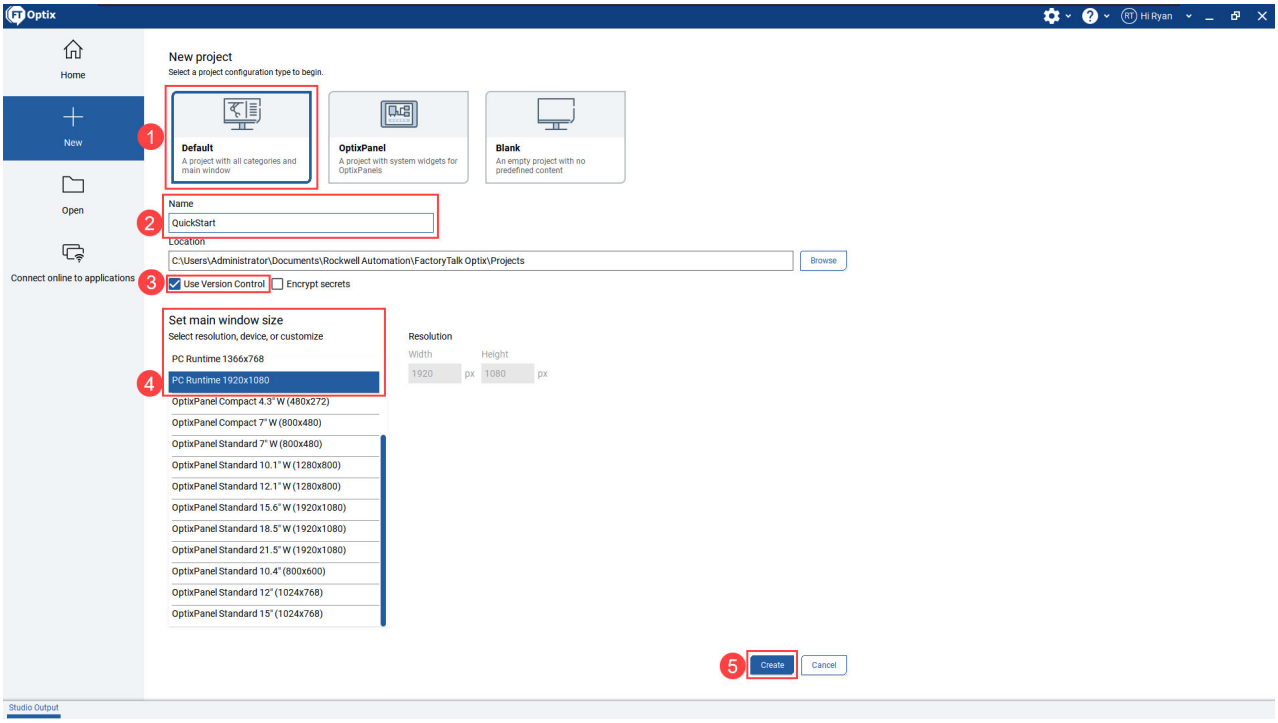
3. Under Name, enter the name of the project. For example, enter "***QuickStart***".
4. Verify that **Use version control** is selected.
5. We aim to run the Optix application full screen on the device of your choosing.

   Note: To run the project on your local computer, select PC Runtime 1920x1080 in Set main window size or the more appropriate resolution for your computer.

© 2025 Rockwell Automation, Inc. All rights reserved.

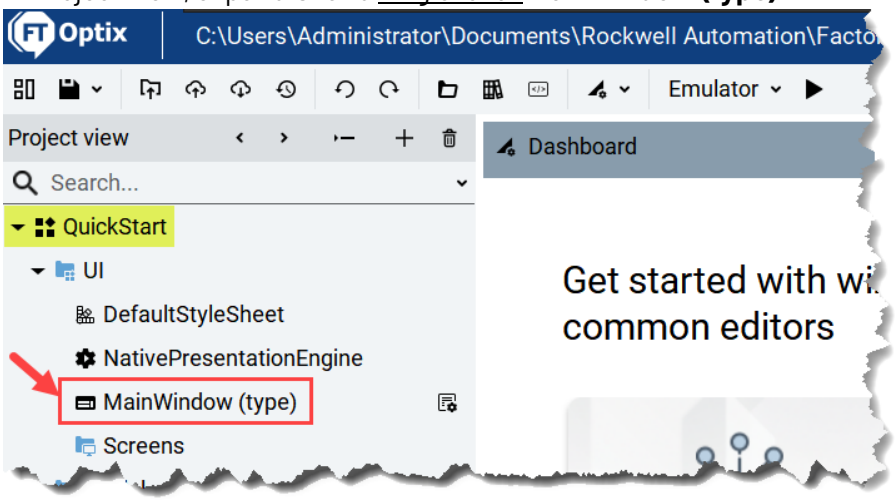
Note: To run the project on an OptixPanel, you can either specify the appropriate resolution, or simply repeat step 2, select OptixPanel, and select the OptixPanel you have so the resolution is set up automatically for you and OptixPanel specific system widgets are automatically added to the project.

6. Select **Create**.



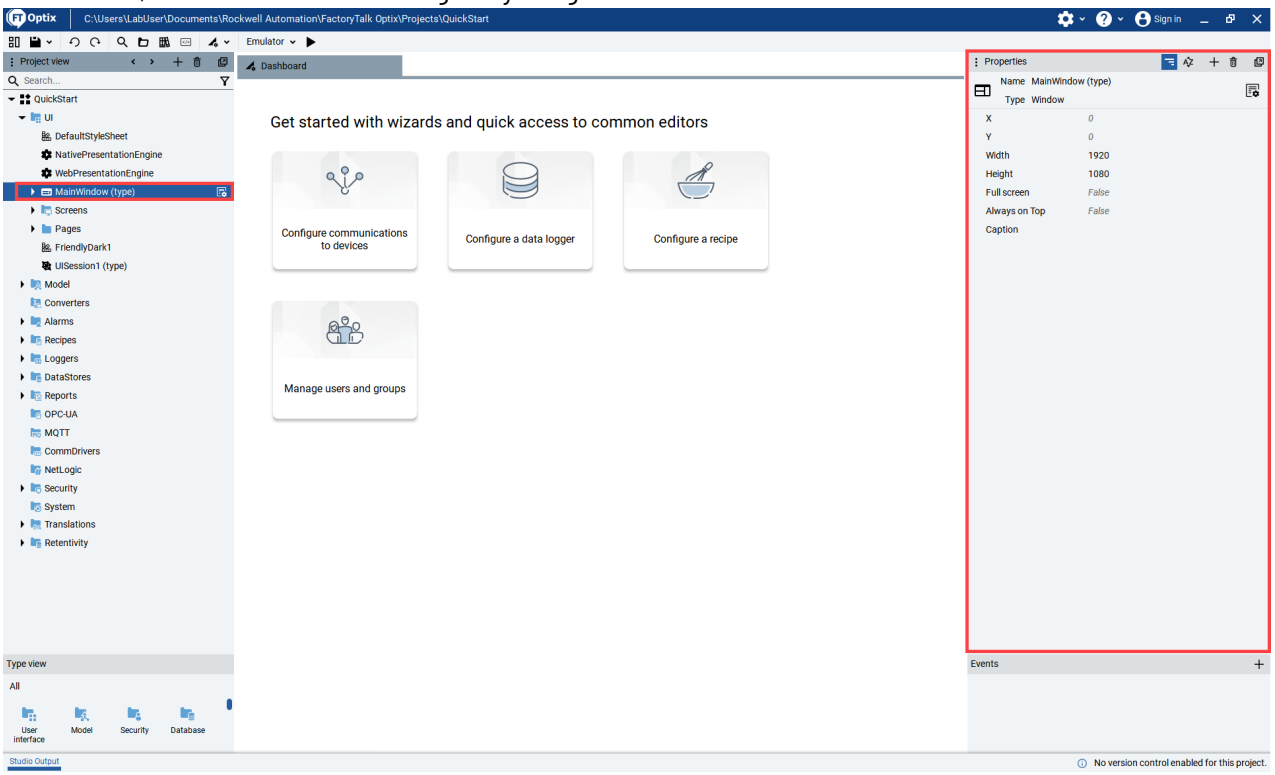
Basic navigation in FactoryTalk Optix Studio

1. In Project view, expand **UI** and single-click **MainWindow (type).**



© 2025 Rockwell Automation, Inc. All rights reserved.

2. Note that the middle pane called the Editor does not change. However, the Properties pane on the right displays the properties for MainWindow. Your Properties pane may show a different resolution from the screen capture below if you had selected an OptixPanel earlier.
For this lab, we do not have to change anything here.



Note: Please note how MainWindow is selected on the left under Project view, the middle pane called Editor displays the Dashboard and the Properties pane on the right is displaying the properties for MainWindow.

3. Double-click **MainWindow (type).** The MainWindow is now displayed in the editor.

Important Note: A single-click on a node in the Project view will give you access to its properties in the Properties pane. Double-clicking will open it for further editing in the center pane called Editor.

© 2025 Rockwell Automation, Inc. All rights reserved.

DEFAULT PROJECT COMPONENTS

A default project will provide the following 4 User Interface related items:

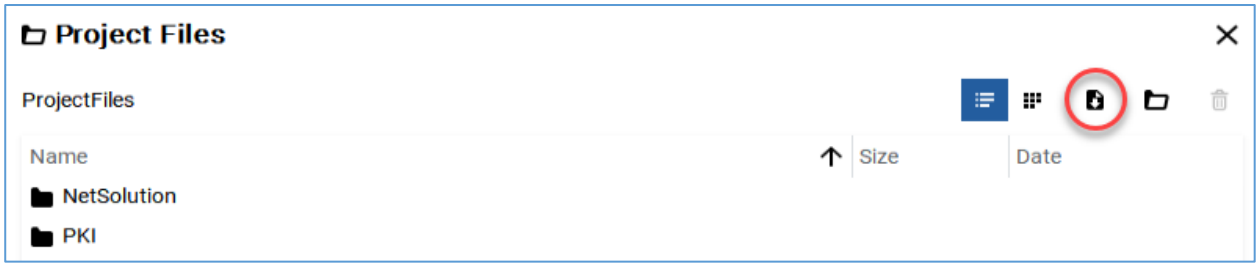
- **Default Style Sheet**: Makes it possible to globally set style properties of all graphical objects in the project. New! Starting with FactoryTalk Optix version 1.5, merging stylesheets is now available.
- **Native Presentation Engine**: Use the **Native Presentation Engine** for typical HMI applications that run on targets with an operator panel. There can be only one **Native Presentation Engine** in a FactoryTalk Optix Studio project.
- **Main Window**: This object is the root container of graphical objects. The main window contains all graphical elements displayed at design time in FactoryTalk Optix Studio and at runtime in your FactoryTalk Optix Application
- **Screens**: Folder where screen types can be created

You can set **Full screen** to **True**, however; it is easier to develop and preview applications with the default **False** settings.

(Optional) Create a Logo

1. (Optional) Minimize Optix Studio and use File Explorer to locate the **C:\Program Files\Rockwell Automation\FactoryTalk Optix\Studio 1.5.x.x \Help\en\downloads** folder
2. (Optional) Double-click **ra_logo.zip**
3. (Optional) Copy the **ra_logo.svg** file and paste it under the root of drive C:
4. (Optional) Close File Explorer.
5. (Optional) Back in Optix Studio, from the toolbar, click the **Browse project files** icon .

6. (Optional) In the Project Files dialog box, click the **Import file(s)** icon .

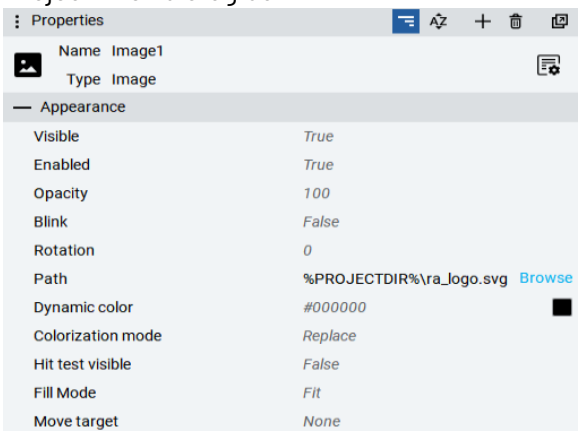


7. (Optional) Browse to **C:**, select **ra_logo.svg** and click **Open**. The file is now imported and can be used in the project.
8. (Optional) **Close** the Project Files dialog box.
9. (Optional) In Project view, right-click **MainWindow (type)**, and select **New** > **Drawings** > **Image**.

© 2025 Rockwell Automation, Inc. All rights reserved.

10. (Optional) Under the MainWindow, select the newly added Image object called **Image1** to access its Properties (on the right)

11. (Optional) In Properties, for **Path**, click **Browse**, select **ra_logo.svg** and click **Select** to close the Project Files dialog box.

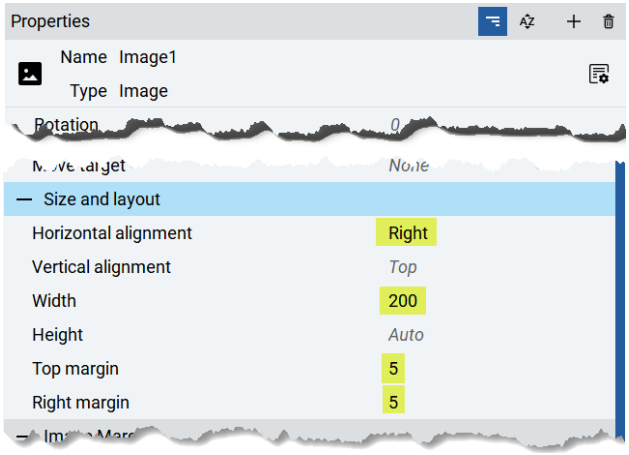


12. (Optional) Still in Properties, set **Horizontal alignment** to **Right**.

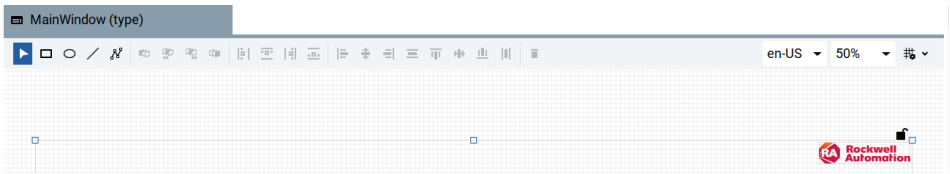
13. (Optional) Mouse over the **Width** property and notice the pencil icon appears  . Select this icon to type in new values for a property. Set Width to "**200**".

14. (Optional) Change the **Height** property from its default value to "**Auto**" (just type **Auto** please)

15. (Optional) Set **Top margin** and **Right margin** to "5".



16. (Optional) Double-click **MainWindow (type)** and scroll horizontally if needed to see the new location and size of the logo.



© 2025 Rockwell Automation, Inc. All rights reserved.

Screen Types and Instances

Configure screens based on a type to distribute interface elements over multiple pages in the application. It is common to have a folder that is used to store all your templates. In this lab, the Screens folder will contain the screen type (template or base object), and we will create another folder to store the instances of the template screen type.

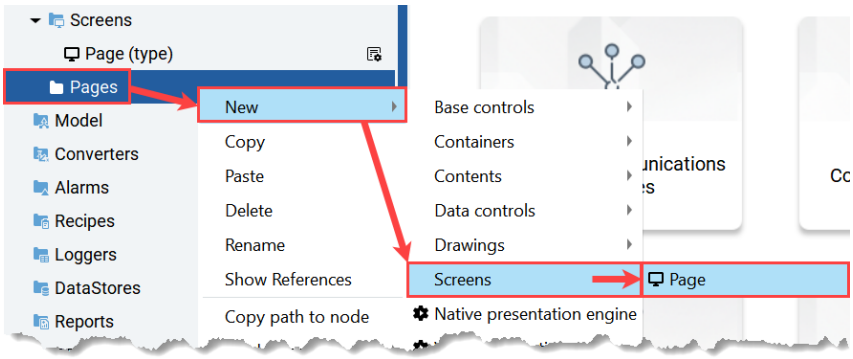
1. In Project view, right-click **Screens** and select **New > Screen**.
   Screen1 (type) appears under **Screens**.

2. Mouse-over **Screen1 (type)**, select the edit icon , and enter "**Page**" as the new name.

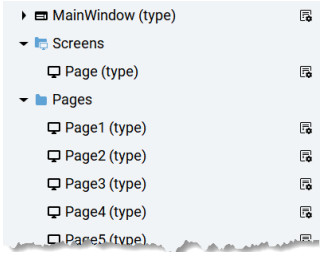
3. In Project view, right-click **UI** and select **New** > **Folder**.

4. Mouse-over **Folder1**, select the edit icon , and enter "**Pages**" as the new name.

5. Right-click **Pages** and select **New** > **Screens** > **Page**. Page1 (type) appears under Pages.



6. Now, with **Page1 (type)** under **Pages** still highlighted press **Ctrl+C** to copy the object. Select the **Pages** folder, then press **Ctrl+V** thirteen more times to create **Page2**, **Page3**, **Page4**, ..., **Page14**. Each Page# (type) inherits its properties from **Page (type).** There will be fourteen instances of the **Page (type)** object total under the **Pages** folder. They will be used as starting templates throughout the lab for the various sections.



Note: To properly instantiate child objects from a parent object type, only use the copy and paste method when the child type has not been modified. Otherwise, all specific changes to the child instance will be copied and it will not reflect the original parent type any longer. In cases where only the original type is meant to be instantiated, use the method described in step #5 above.

7. To see the instantiation in action, still in In Project view, double-click **Page** under the Screens folder. In the Properties pane for the screen called Page, set **Left margin**, **Top margin**, **Right margin** and **Bottom margin** to "10".

© 2025 Rockwell Automation, Inc. All rights reserved.

8. Set the **Background color** property to the lightest gray color you can select from the color picker with Hex #f0f0f0.

9. **Save** the project and note how these changes are inherited by all other pages (Page1 through Page14).

INSTANCES OF PARENT OBJECT TYPES

Create consistent user interfaces by inheriting properties from parent object types. For example, if you want to change the margins of **Page1 (type), Page2 (type), Page3 (type)**, **Page 4 (type)**, ..., edit the margin properties of **Page (type)**. The modification will propagate to each instance of this type.

Navigation Panel

Screen navigation in FactoryTalk Optix can be done in two ways

- Panel Loader container: Display panels based on the application logic. This is the more traditional way of designating an area on the main window as the location of where panels are loaded and then a button is configured to call the desired panel.
- Navigation Panel container: Automatically organizes panels into tabs that you can navigate to at runtime.

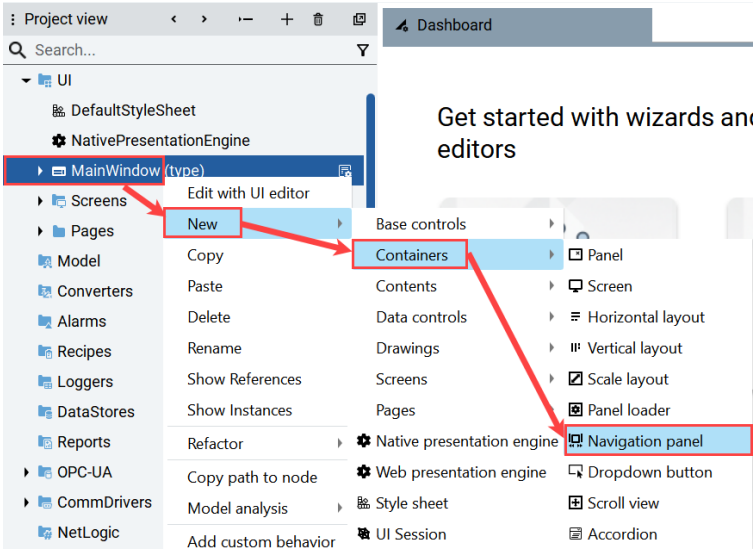
To see both in action, please check the sample project called **Training_UserInterface** available through the public git repository https://github.com/factorytalk-Optix

In this lab, we will configure the Navigation Panel.

1. In Project view, double-click **MainWindow (type).**

© 2025 Rockwell Automation, Inc. All rights reserved.

2. Right-click **MainWindow (type)** and select **New** > **Containers** > **Navigation panel**.



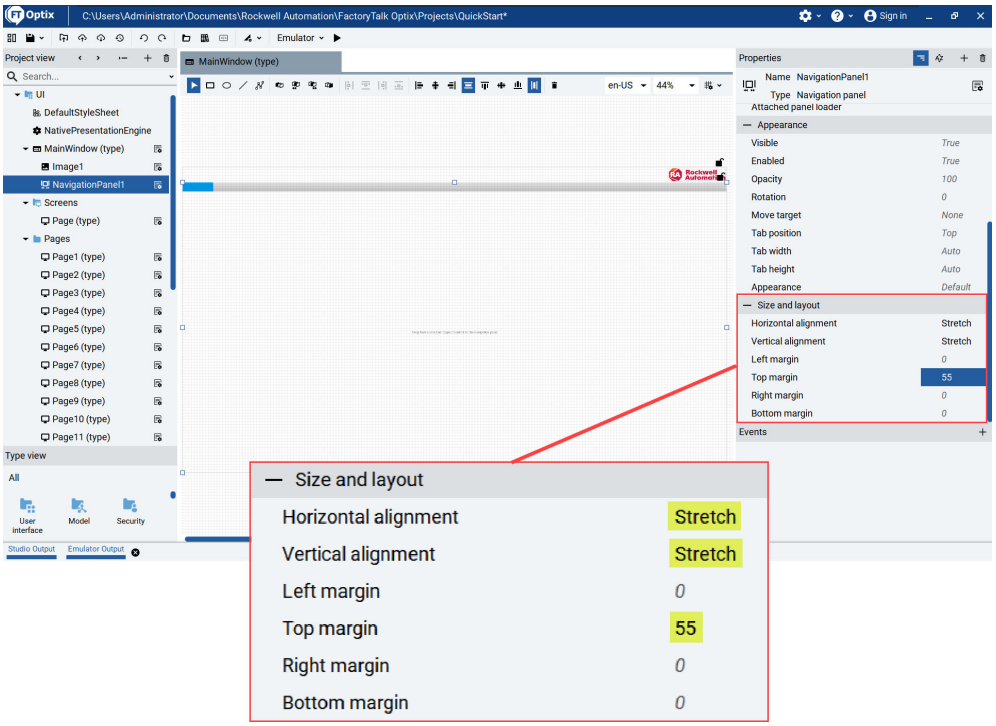
The navigation panel appears in the main window and is not positioned.

Note: To learn more about Containers see Appendix E – Container Objects.

3. In Properties, set **Horizontal alignment** and **Vertical alignment** to **Stretch**.

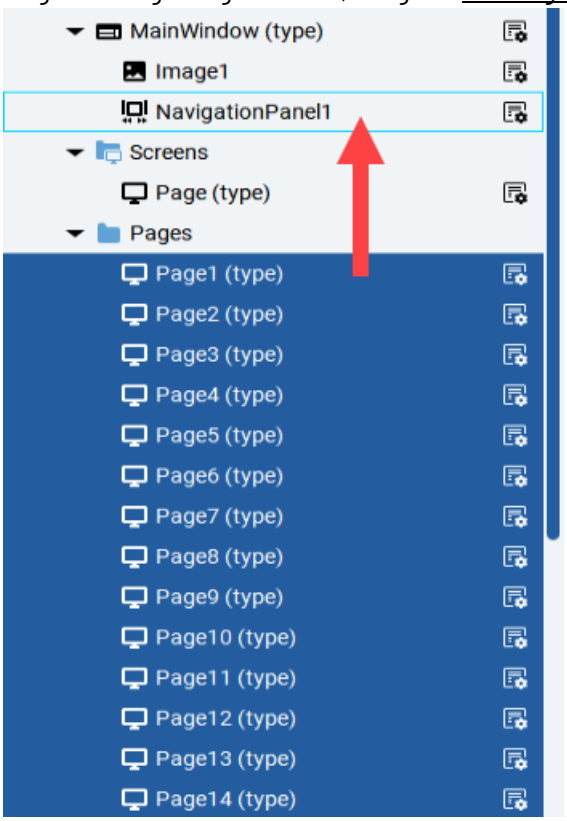
Now, the navigation panel occupies the entire height and width of the main window and covers the logo.

4. Set **Top margin** to **"**55" (you may need to scroll down the Properties pane to locate this property). The navigation panel no longer covers the logo.

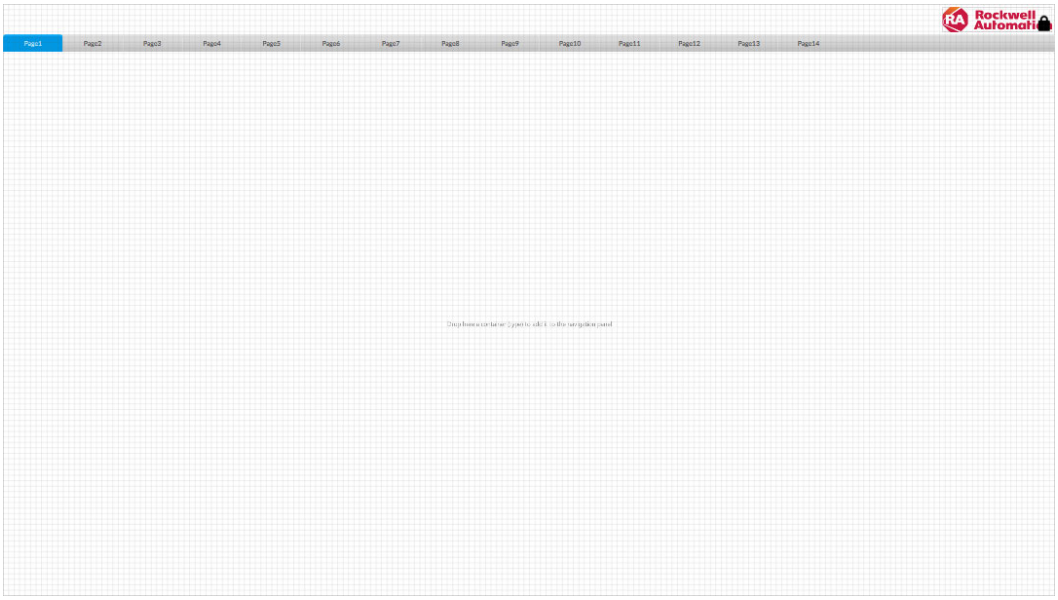


© 2025 Rockwell Automation, Inc. All rights reserved.

5. Click to select **Page1 (type)**, then hold **Shift** and click on **Page14 (type)**. This will select all screens Page1 through Page14. Now, drag the entire group of pages onto **NavigationPanel1**.



Double-clicking MainWindow will show a preview of the NavigationPanel



6. **Save** your project.

© 2025 Rockwell Automation, Inc. All rights reserved.

Creating Objects

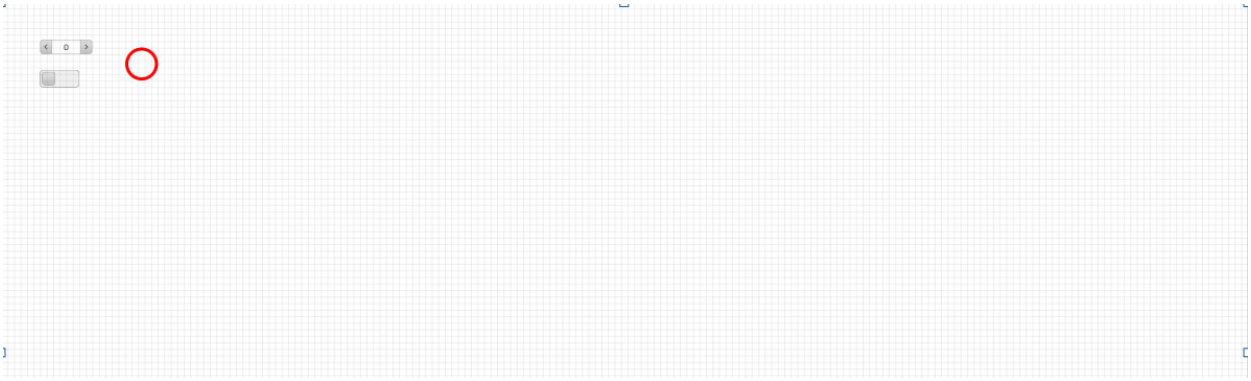
Add a Switch, Spin box, and LED objects to Page1 of your application.

1. In Project view, double-click **Page1 (type)**

   The currently blank Page1 displays in the editor.

2. Mouse-over **Page1**, select the edit icon , and enter "**Dashboard**" as the new name.
3. Right-click **Dashboard (type)** and select **New** > **Base controls** > **Switch**.
4. Right-click **Dashboard (type)** and select **New** > **Base controls** > **Spin box**.
5. Right-click **Dashboard (type)** and select **New** > **Base controls** > **LED**.
6. In the center pane of the editor, arrange the objects according to your preferences.

   Note: When building applications that you intend to deploy to a device with a resistive touch screen, we recommend that you **make your objects large enough to operate easily.** The touch sensitive area should be at least as large as your fingertip.



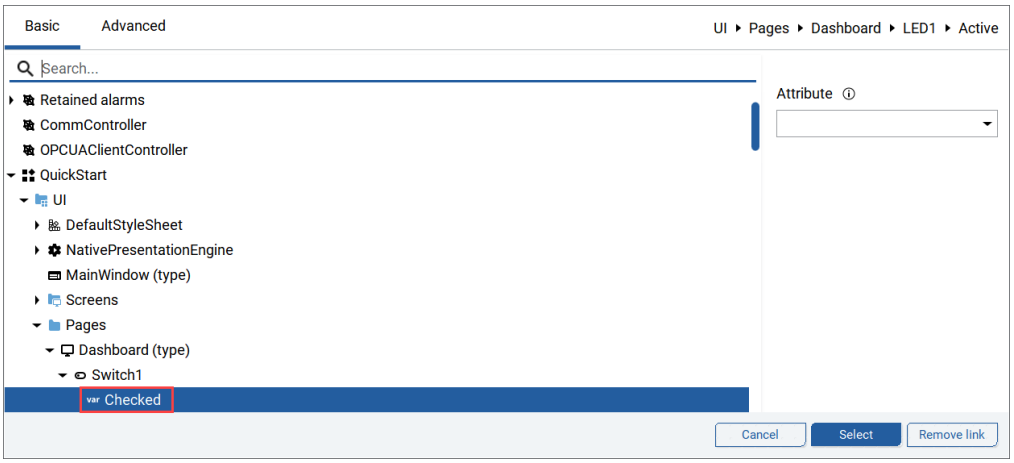
Associate the LED status with the switch

Turn the LED object on and off with the switch.

1. In Project view, select **LED1**.
2. In the **Properties** pane on the right, mouse-over the **Active** property and click the Add Dynamic Link icon .

© 2025 Rockwell Automation, Inc. All rights reserved.

3. In the dynamic links browser, navigate to **QuickStart** > **UI** > **Pages** > **Dashboard (type)** > **Switch1** > **Checked**.

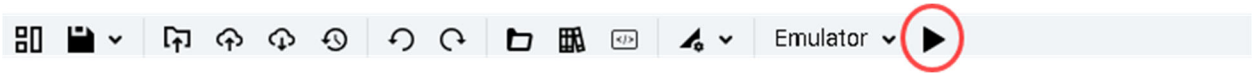


4. Choose **Select**.

   Note: A dynamic link can also be created by dragging the source from Project view to the value of the property that you want the dynamic link to set.

5. **Save** your project.

6. From the toolbar, with **Emulator** still displayed. Click the Run Emulator icon ▶.

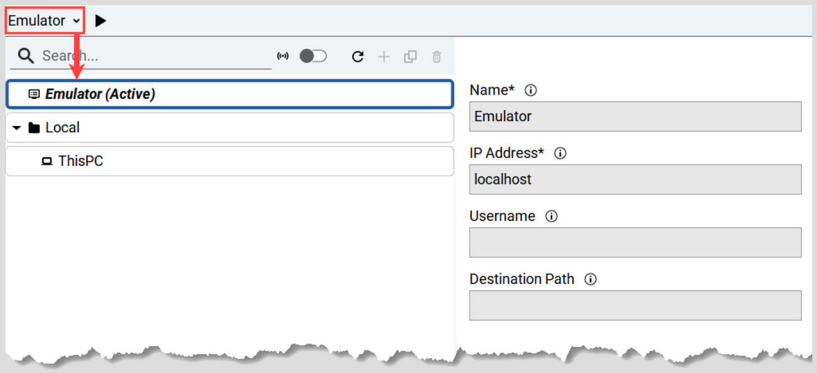


EMULATOR

FactoryTalk Optix Studio includes an **Emulator** to make developing and testing your runtime project easy.

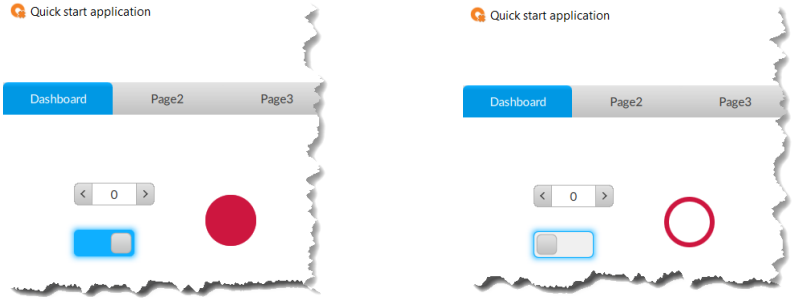
- The **Emulator** will run for two hours at a time, and it may be restarted as needed after the time limit expires.
- Multiple projects may be emulated at the same time.
- The **Emulator** is the default deployment option when pressing the play button within FactoryTalk Optix Studio.

Note: Click on **Emulator** to expand the runtime deployment configuration window and explore the options available there.



© 2025 Rockwell Automation, Inc. All rights reserved.

7. At runtime, in the **Emulator**, toggle the switch and observe the LED change color. Toggle the switch off and observe the LED turn off.



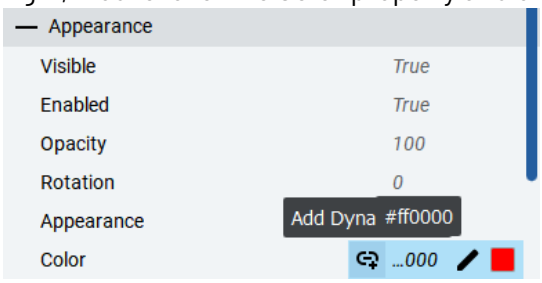
Note: Depending on the MainWindow resolution that was selected earlier when creating this project, you may have to scroll right or left to see all the screens we can navigate to using the Navigation panel.

8. Close the **Emulator**.

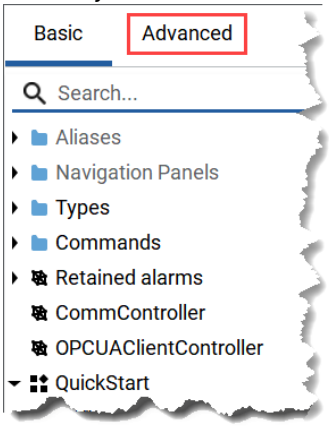
Key-value converter

Use a key-value converter to control the color of the LED object by changing spinbox values at runtime.

1. In Project view, select **LED1**.
2. Next, we will create a complex dynamic link for the Color property. In the Properties pane on the right, mouse-over the **Color** property and click the **Add Dynamic Link** icon .

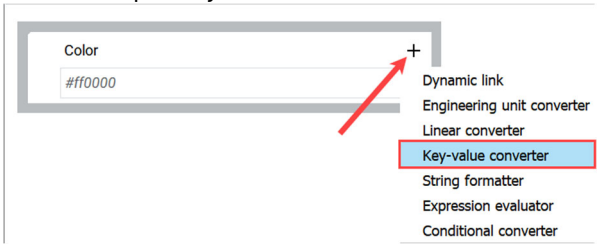


3. In the dynamic link browser, select the **Advanced** tab.

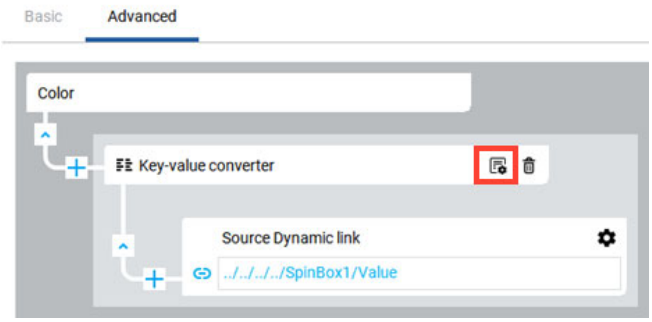


© 2025 Rockwell Automation, Inc. All rights reserved.

4. In the complex dynamic link editor, select the **Add new** icon + and select **Key-value converter**

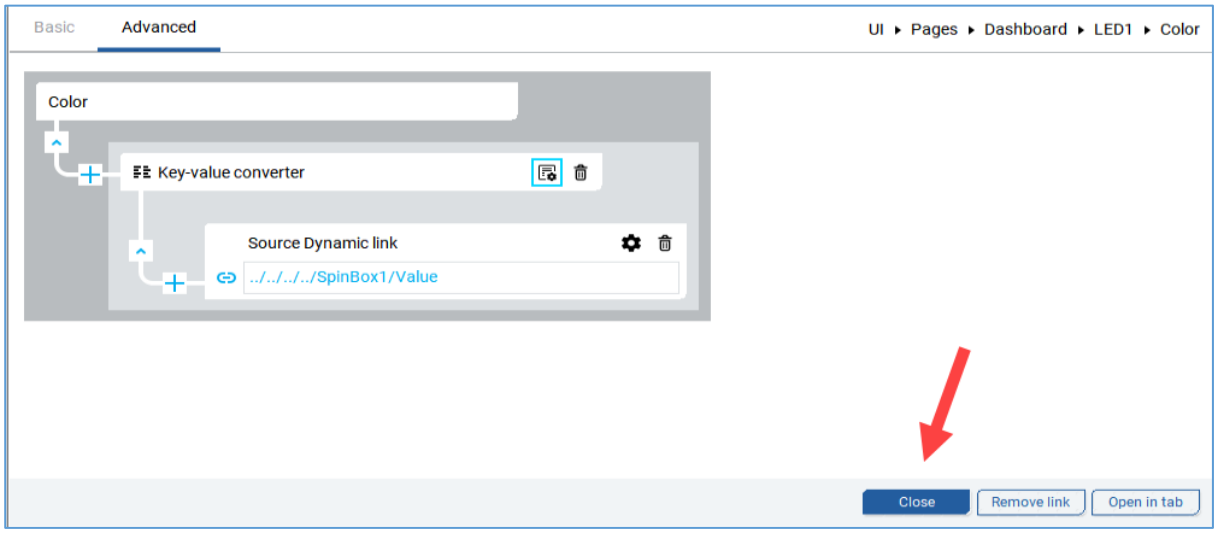


5. To specify the source and converter logic to transform the value, click the **Change Dynamic Link** icon, browse and select **QuickStart** > **UI** > **Pages** > **Dashboard (type)** > **SpinBox1** > **Value** and click **Select.**



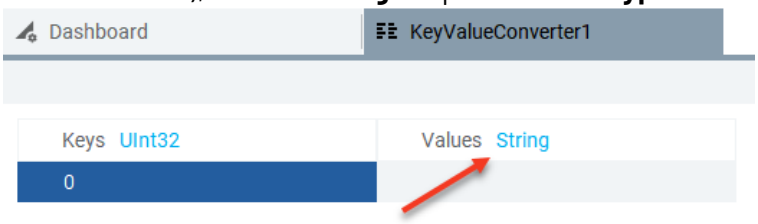
6. Next to **Key-value converter**, select the configure icon.

7. **Close** the dynamic link browser.

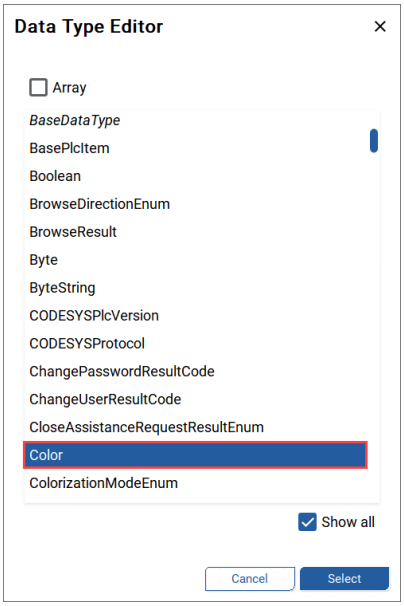


© 2025 Rockwell Automation, Inc. All rights reserved.

8. In the central pane of the editor, the Value Map editor is displayed. Next to **Values** (header of the second column), click on **String** to open the **Data Type Editor**.

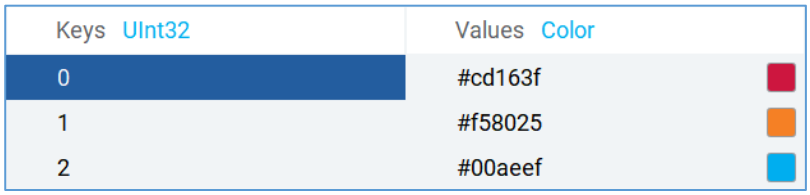


9. Select the **Show all** checkbox and change the data type to **Color**.



10. Click **Select**.

11. Add two rows by selecting the Add new icon + two times and set the values as follows:
Note: For this lab, you may type **Red**, **Orange** and **Blue** as the colors instead of the hex value of the color.



12. Close the **KeyValueConverter1** tab.

13. In Project view, select **SpinBox1**.

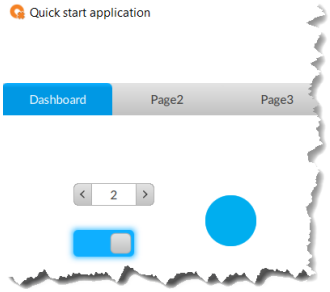
© 2025 Rockwell Automation, Inc. All rights reserved.

14. Let's set a minimum and maximum to prevent a user from providing values outside of a defined range at runtime. In **Properties**, set **Minimum value** to "0", and the set **Maximum value** to "2".

Note: When building applications that you intend to deploy to a device with a resistive touch screen, we recommend that you **make your objects large enough to operate easily.** The touch sensitive area should be at least as large as your fingertip.

15. **Save** the project and click the Run Emulator icon ▶.

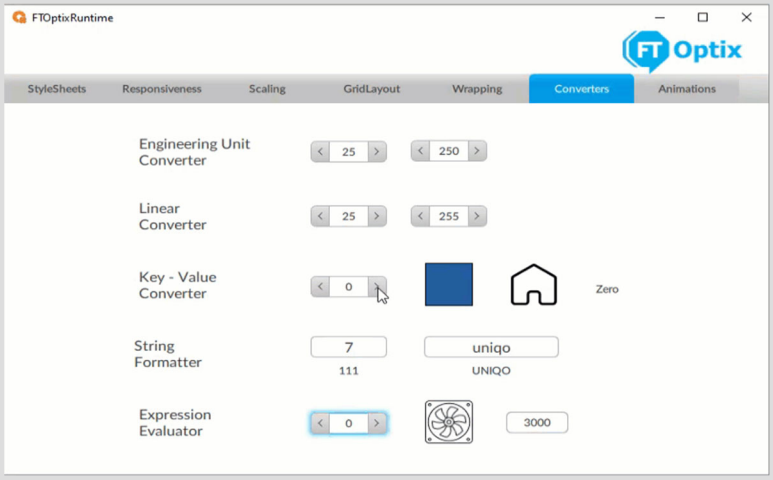
16. At runtime, in the **Emulator**, change the LED change color using the spin box.



17. Close the **Emulator**.

CONVERTERS

Converters can be used to create your own multistate indicator. To learn more on converters, check Appendix F. You can also check the sample project called **Training_UserInterface** available through the public git repository for Optix https://github.com/factorytalk-Optix



© 2025 Rockwell Automation, Inc. All rights reserved.

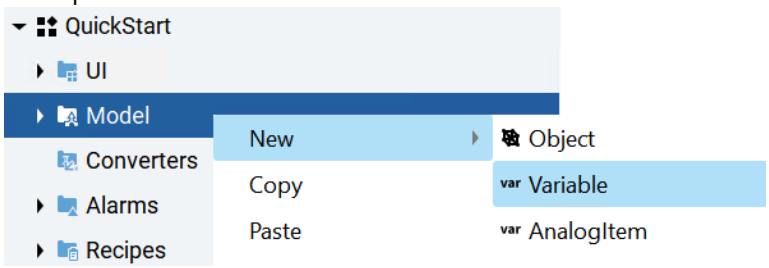
Create variables

Create variables that define a motor alarm, temperature, and individual ingredients.

Note: With Logix-based controllers, tags from the controller can be imported during design-time, using the tag importer in FactoryTalk Optix Studio. You can also import tags at runtime. Starting with Optix 1.5, Logix tag extended properties can also be imported.

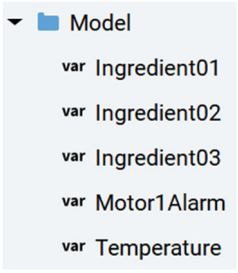
Note: For bulk changes of Model variables and their values (creating or updating) using a CSV file, use the design-time script available from the FactoryTalk Optix Studio Libraries. You may search Libraries for *Import Export Model Variables*.

1. In Project view, right-click the **Model** folder and select **New** > **Variable**. Repeat this step 4 times to end up with 5 variables in total.



Note: You may also copy and paste the variable you created 4 more times.

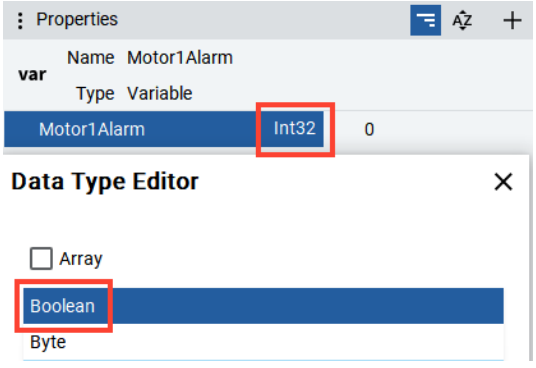
2. Mouse-over each variable, select the edit icon , and enter new names as ***Ingredient01***, ***Ingredient02***, ***Ingredient03***, ***Motor1Alarm***, and ***Temperature*** as shown below:



Note: You may consider creating folders and subfolders when working with many variables.

3. In Project view, select **Motor1Alarm**.

4. In **Properties**, change the data type from **Int32** to **Boolean**.



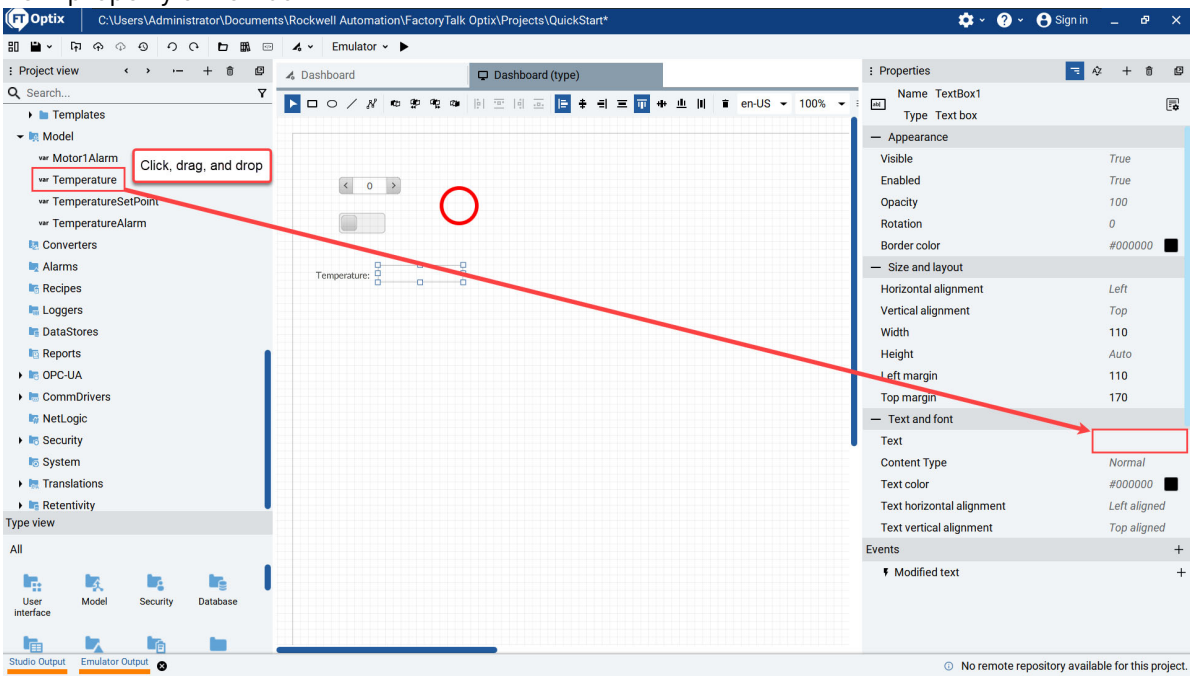
© 2025 Rockwell Automation, Inc. All rights reserved.

5. The variable value type changes to **Boolean** for the digital alarm that you will configure later. **Save** the project.

Configure temperature controls

Create **Label**, **Textbox**, and **Linear gauge** objects to visualize and control the temperature variable.

1. In Project view, under *UI*> *Pages*, double-click **Dashboard (type)**.
2. Right-click **Dashboard (type)** and select **New** > **Base controls** > **Label**.
3. In Properties, set Text to "***Temperature:***".
4. In Project view, right-click **Dashboard (type)** and select **New** > **Base controls** > **Text box**.
5. In Properties, use drag and drop to create a dynamic link. Drag the **Temperature** variable to the **Text** property of Textbox1.

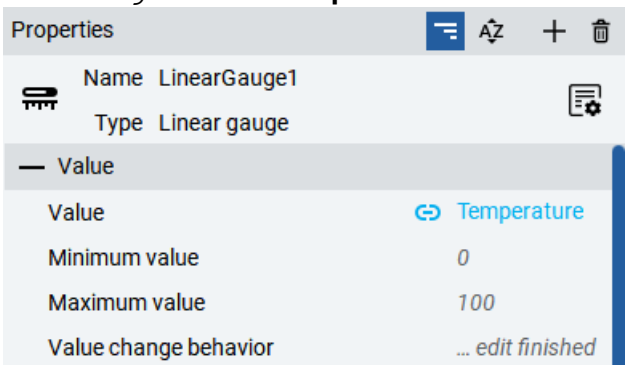


The Temperature variable is now linked to the text property of Textbox1.

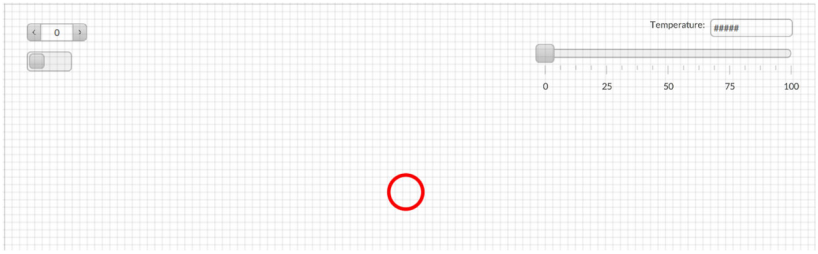
6. In Project view, right-click **Dashboard (type)** and select **New** > **Base controls** > **Linear gauge**.

© 2025 Rockwell Automation, Inc. All rights reserved.

7. Like step 5, use drag and drop to create a dynamic link between the **Value** property of *LinearGauge1* and the **Temperature** variable.



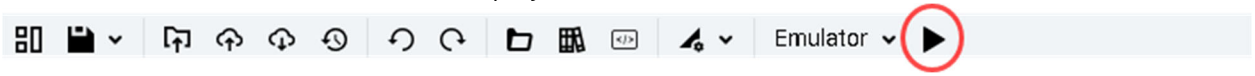
8. In Project view, double-click **Dashboard (type)** and arrange the objects according to your preferences.



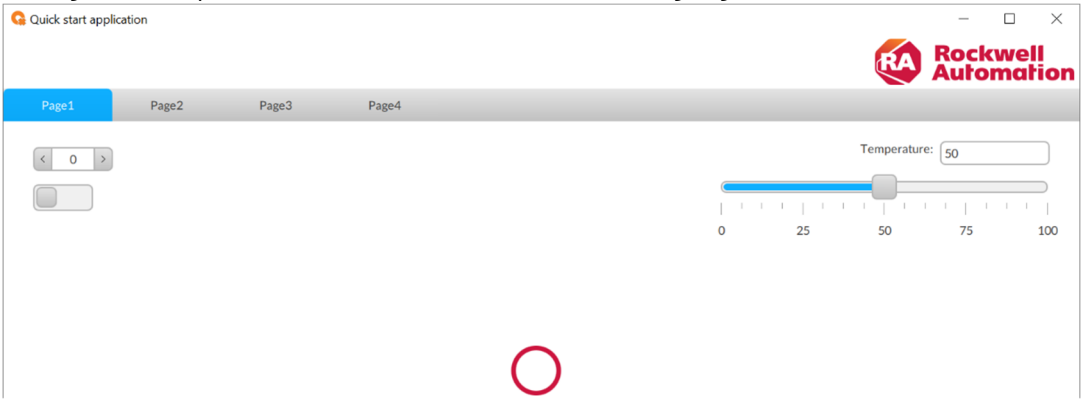
Note: When building applications that you intend to deploy to a device with a resistive touch screen, we recommend that you **make your objects large enough to operate easily.** The touch sensitive area should be at least as large as your fingertip.

9. **Save** the project.

10. From the toolbar, with **Emulator** still displayed. Click the Run Emulator icon ▶.



11. Change the temperature value and make sure that the gauge and the text box are linked.



Note: Min and Max values can be configured so new values are within the expected range.

12. Close the **Emulator**.

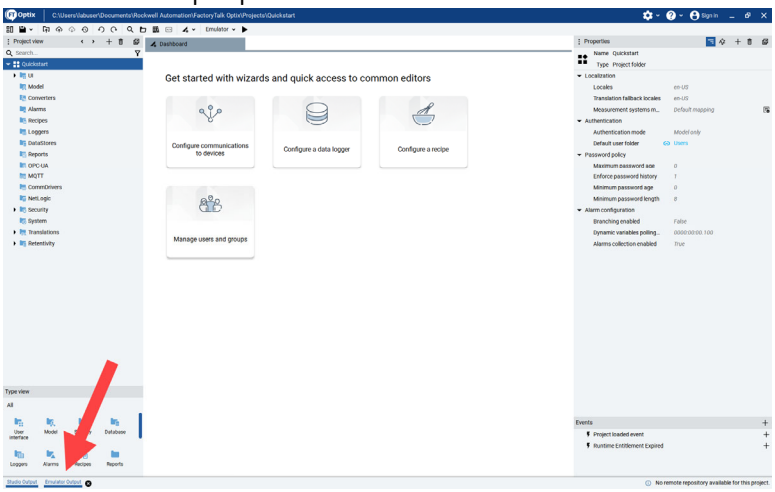
© 2025 Rockwell Automation, Inc. All rights reserved.

Runtime sizing in FactoryTalk Optix Studio

There are two ways to determine the runtime entitlement (license) that is needed for this project:

- Option 1: We can use the Runtime sizing calculator available through the Optix product page via FactoryTalk Hub. This is a great calculator to use before the project is started.
- Option 2: Use FactoryTalk Optix Studio to determine the runtime feature tokens used. Feature tokens function as a unit of currency and accumulate as you configure more features in a project.

1. Let's use FactoryTalk Optix Studio. Click the **Emulator Output** tab located closer to the bottom left to show the output pane.



2. Make sure that **Messages** is selected and mouse-over the log entry that states: **FactoryTalk Optix Runtime is currently using 1 feature token**. So far in this project, the native presentation engine is the only component that is impacting the runtime entitlement needed to run this project for more than 2 hours. Number of tags or screens do not impact the runtime license needed.

Note: Optionally, you can right-click a log entry, select **Copy selected logs**, and paste it in a text editor like Notepad to more easily get the details in the format and font size you prefer.

SUMMARY

In a few clicks we covered some basic and advanced topics. For example:

- We created one panel as a Type and created several instances of that panel. Any change to the template panel would propagate to the panel instances.
- We learned how a property of an object can be linked to a variable or a property of another object.
- We created dynamic links, advanced dynamic links and a converter.

You have completed the Create Your First Project lab.

© 2025 Rockwell Automation, Inc. All rights reserved.

Web Presentation Engine and Session Variables (10 Minutes)

Objectives

- Create a Web Presentation Engine
- Transfer application to a runtime device (Web Presentation Engine host)
- Connect a web client to the runtime host device
- Create and use Session variables

Scenario

Web Presentation Engine exposes a web user interface for connecting multiple users from different web browsers. When a user connects to the web server, an interactive session is generated for the authenticated user. The same user can generate additional interactive sessions by accessing the project from multiple devices.

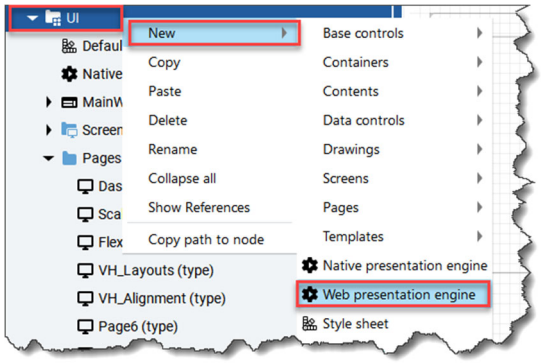
Lab procedure

Create a Web Presentation Engine

Note: Each FactoryTalk Optix Studio project can contain a single Native Presentation Engine and a single Web Presentation Engine. There is no limit to the number of web clients that can connect to the Web Presentation Engine.

Note: **New!** Starting with FactoryTalk Optix version 1.3, the web client IP Address property is available to be used as a client session property in your application logic. This allows you to perform specific actions, for example open specific screens, based on the web client IP Address.

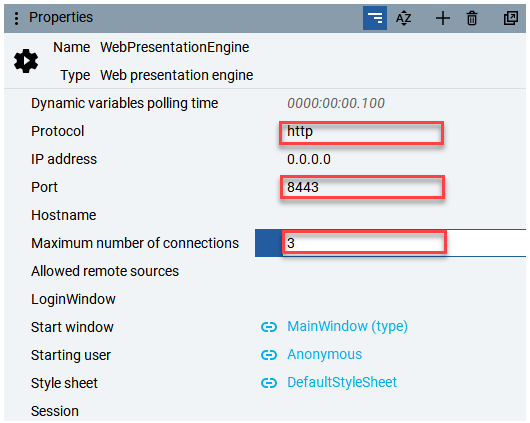
1. In Project view, right-click the **UI** folder (make sure it is expanded) and select **New** > **Web Presentation Engine**.



© 2025 Rockwell Automation, Inc. All rights reserved.

2. In the **Properties** pane on the right make the following changes:

   - Set Protocol to **http** (The recommendation is to use https and SSL certificates in production)
   - Set Port to **8443** (This is the port to be used for communication between web clients and the server and MUST be specified as part of the URL that clients will use)
   - Change the Maximum number of connections to **3** (This is the maximum number of concurrent connections to the web server that the application will allow. The number entered here will impact the license required for the application)

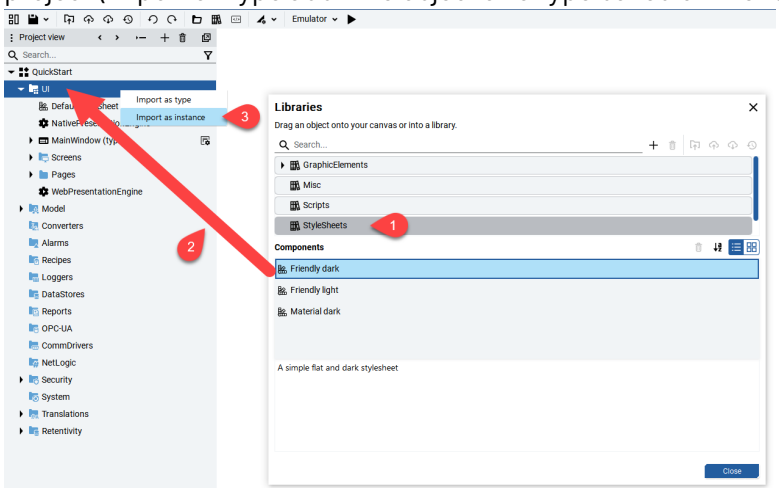


Note: You can set the IP address to 127.0.0.1 to host the server locally. Do not use the "localhost".

Add a new Stylesheet to the project

1. From the **Toolbar** select **Template Libraries** icon

2. From Libraries
   - Locate the **StyleSheets** section
   - Drag and drop the **Friendly dark** stylesheet to the **UI** folder.
   - Select **Import as Instance**. This will add the object as an instance based on the template into the project (Import as type adds the object as a type based on the template into the project)



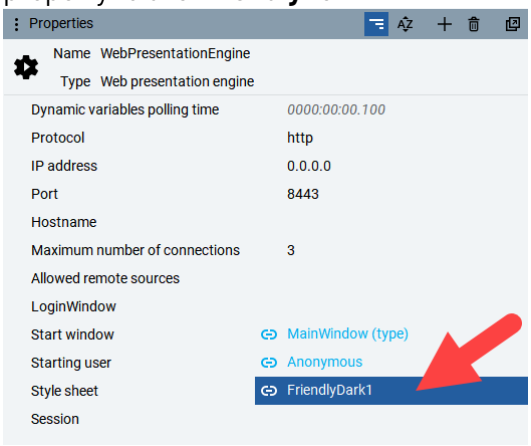
The FriendlyDark1 style sheet is now available to use in the project.

© 2025 Rockwell Automation, Inc. All rights reserved.

3. Select **Close** to close the **Libraries** window.

Set the Web Presentation Engine to use the new style sheet

4. In Project view, under the **UI** folder, select **Web Presentation Engine.**
5. In the **Properties** pane for the Web Presentation Engine on the right, change the **Style sheet** property to use **FriendlyDark1**.



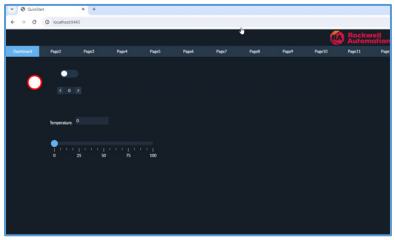
6. **Save** the project.

Connect a web client to the runtime host device

1. From the toolbar, click the **Run on Emulator** icon ▶.

2. Wait for the Optix Runtime application to launch
3. Open Microsoft Edge browser (or Chrome) from Windows Taskbar.
4. As a URL enter http://localhost:8443

   Once the web session is established, turn the LED object on and off with the switch. Note how the LED changes color only in the web presentation session and not the native presentation. This is because we associated the LED status with the toggle switch directly without using a tag.



5. **Close** the web browser session and the Optix Runtime application.

   Note: New! Starting with FactoryTalk Optix version 1.3, a disconnect alert is displayed on the web client if disconnected from the Optix Runtime, and will automatic reconnect when the connection is re-established.

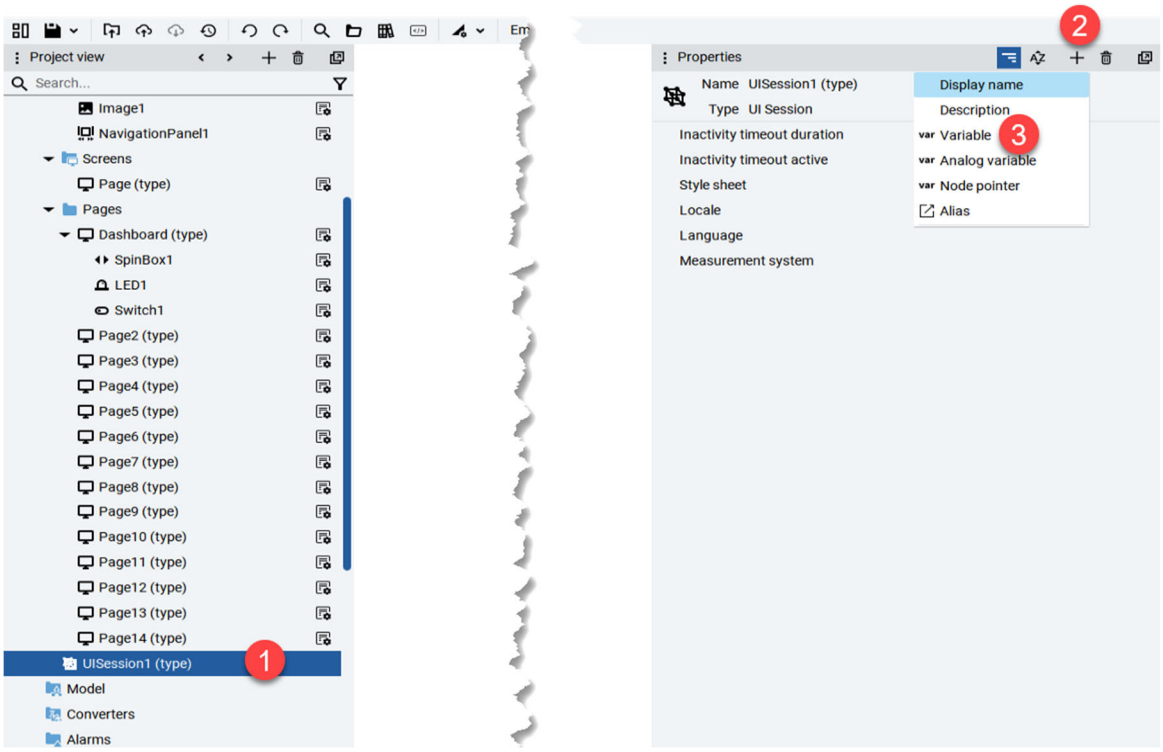
© 2025 Rockwell Automation, Inc. All rights reserved.

Session specific variables

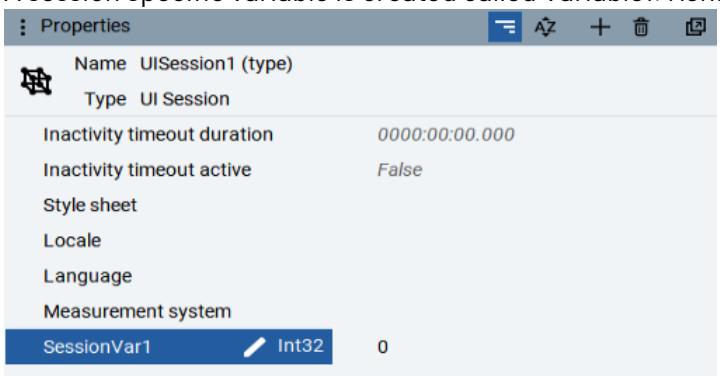
Variables we created earlier under the Model folder are considered global variables to the project. When the value changes, it is changed for the local user of the Native Presentation Engine as well as every remote user of the Web Presentation Engine. However, there are cases when using remote web clients that local variables to each user session is needed. Session variables are treated as variables that are unique to each user session.

1. In Project view, right-click the **UI** folder, select **New** > **UI Session.** UISession1 is added to the tree.

2. Double-click **UISession1** to access its properties on the right. Click the + icon and select **Variable**.



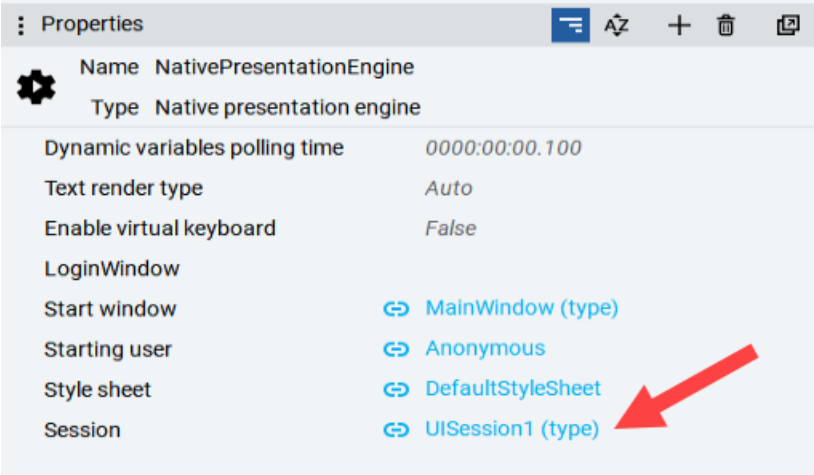
3. A session specific variable is created called Variable1. Rename it to **SessionVar1.**



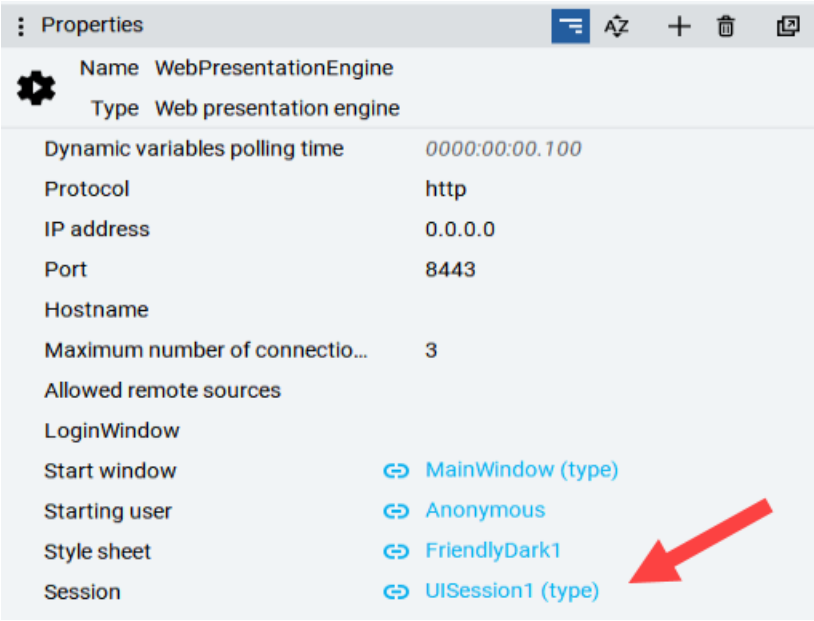
Let's specify UISession1 as the one to use for both Native and Web Presentation Engines.

© 2025 Rockwell Automation, Inc. All rights reserved.

4. Double-click **NativePresentationEngine** and set the **Session** property to **UISession1 (type)**.



5. Double-click **WebPresentationEngine** and set the **Session** property to **UISession1 (type)**.

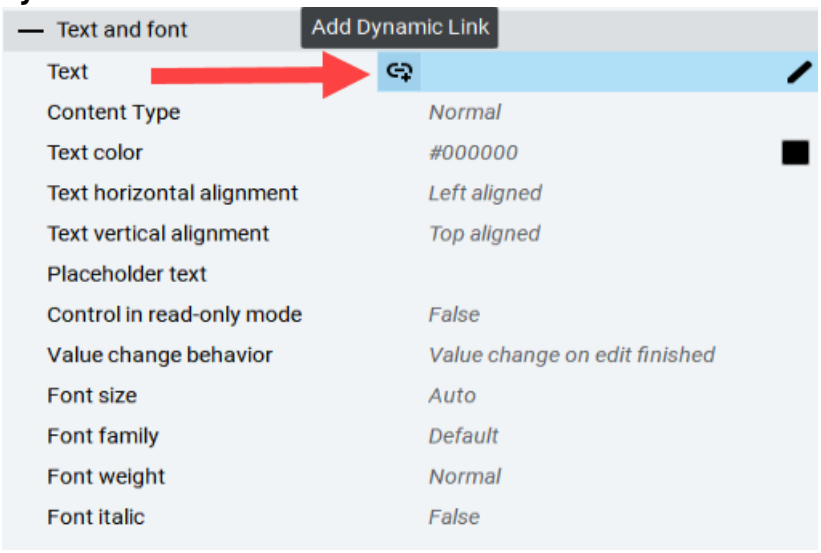


We are ready to use this session variable and see how it behaves compared to Model variables.

6. Double-Click **Dashboard (type)** to display it in the middle pane.
7. Right-Click **Dashboard (type)** and select **New** > **Base controls** > **Text box**
8. Drag and drop the newly added text box anywhere on the screen called Dashboard to position the textbox to your preference.

© 2025 Rockwell Automation, Inc. All rights reserved.

9. In the Properties pane (on the right) for *TextBox2*, mouse-over the **Text** property and click the **Add Dynamic Link** icon .



10. Browse to **Aliases** > **{Session}** > **Session** > **UI Session** > **UISession1** and select **SessionVar1**

11. From the toolbar, click the **Run on Emulator** icon .

12. Wait for the OptixRuntimeTarget application to launch. The NativePresentationEngine uses the default stylesheet with a white themed background. Open Microsoft Edge browser (or Chrome).
13. The URL to use is http://localhost:8443 to connect to the WebPresentationEngine, which uses dark mode. You must specify port 8443, which is the port number we specified earlier in the lab when configuring the web presentation engine.
14. Change the temperature value. The change is reflected in the NativePresentationEngine and the WebPresentationEngine. This is the behavior expected with a global model variable or a PLC tag.
15. In the browser (connected to the WebPresentationEngine), change the session variable we associated with TextBox2 on the Dashboard and note how the change only takes effect in the session it was changed in. Session variables behave as if each user session has its own SessionVar1. Switch to NativePresentationEngine and note how its session variable did not change.
16. Close the **Emulator** and web browser as well.
17. (Optional) Can you determine the total feature tokens needed to run this project in production? Which components in this project are impacting the runtime entitlement size?

You have completed the Web Presentation Engine lab.

© 2025 Rockwell Automation, Inc. All rights reserved.

Alarming (15 Minutes)

Objectives

- Configure a Digital alarm
- Configure an Analog alarm
- Configure Alarm Banner and Alarm Grid
- Configure Alarm History
- Simulate alarms

Scenario

In this section of the lab, you will configure alarms in FactoryTalk Optix and use the new alarm widgets from the libraries released with Optix 1.5.

Important Note

With FactoryTalk Optix and Logix-based controllers (ControlLogix and CompactLogix), using FactoryTalk Alarms and Events (instruction-based and tag-based alarms) in Logix are supported.

Lab Procedure

Note: Screenshots may differ depending on what optional sections have been completed previously.

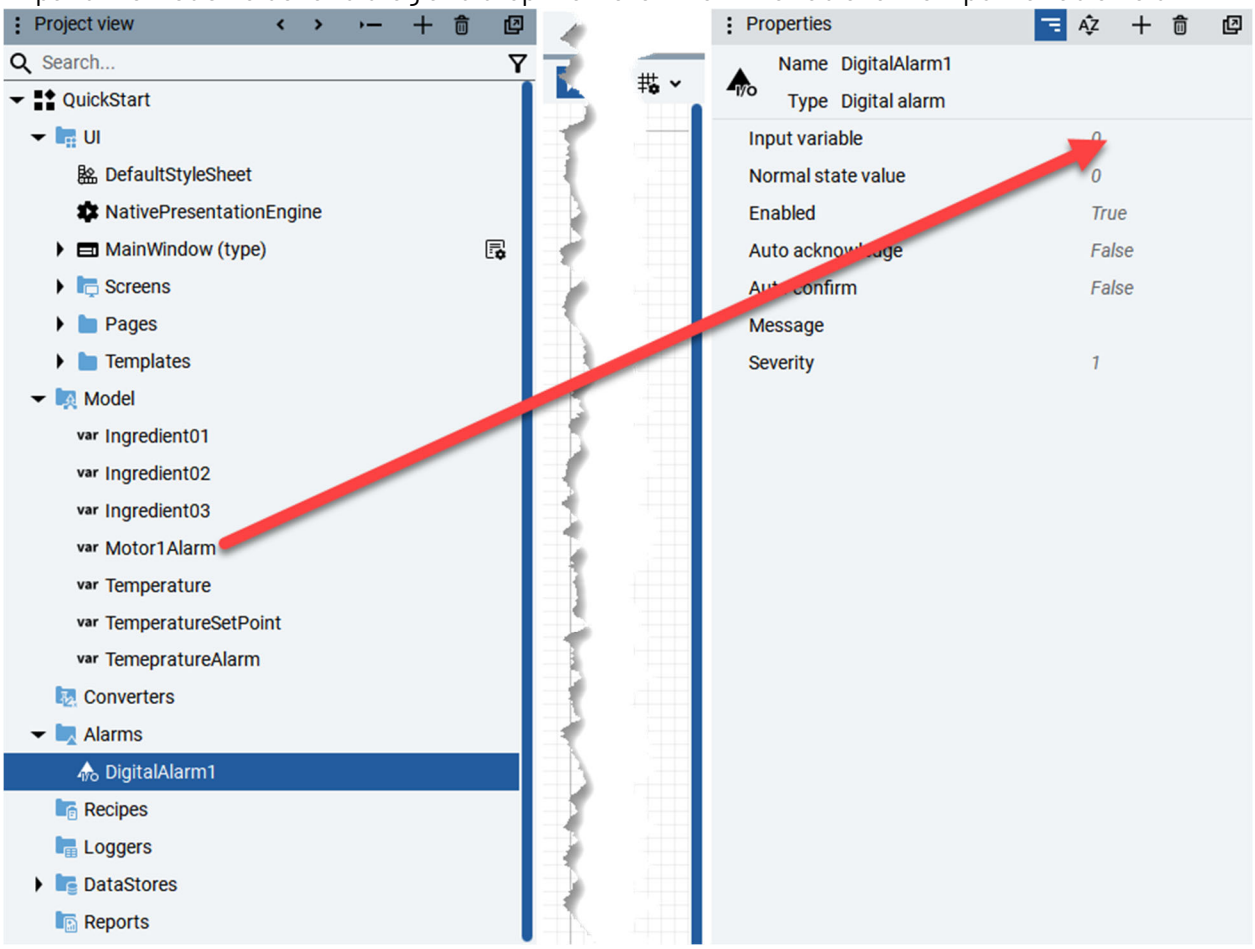
Configure a Digital alarm:

1. In Project view, right-click the **Alarms** folder and select **New** > **Digital alarm**.

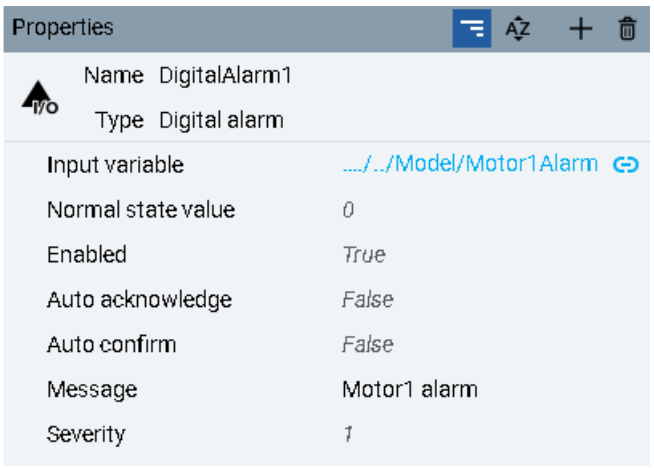
   DigitalAlarm1 will be created. Observe the Properties pane, on the right, associated with the new alarm. Next, you will create a dynamic link between the Input variable property and the Motor1Alarm variable in the Model folder.

© 2025 Rockwell Automation, Inc. All rights reserved.

2. Expand the **Model** folder and drag and drop the **Motor1Alarm** variable to the Input variable field.



3. In **Properties**, in the **Message** field, enter a warning message e.g. "***Motor1 Alarm***".



4. Set the **Auto confirm** property to **True**.

Note: In this lab, we only want to require an operator to acknowledge an alarm. In order to not require an operator to confirm, we can set the alarm to Auto confirm.

© 2025 Rockwell Automation, Inc. All rights reserved.

Configure an Analog alarm

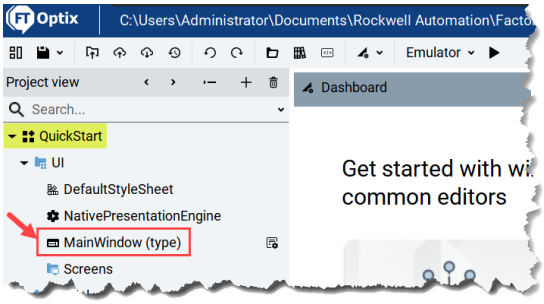
1. In Project view, right-click the Alarms folder and select **New** > **Exclusive level alarm**.
2. Rename ExlusiveLevelAlarm1 to **AnalogAlarm** by clicking on the pencil symbol on the right of the object.
3. Using the same method when creating dynamic link for the digital alarm, expand the **Model** folder and drag and drop the **Temperature** variable to the **Input variable** field in the **AnalogAlarm** object's **Properties** pane (make sure AnalogAlarm is selected in the Project view).
4. With the alarm's Properties pane still open, set the **High limit** to **80**.
5. Under **Message**, in the **Message high-state** field, type the message: "***Temperature is High***". Your configured alarm properties should look like this:

6. Set the **Auto confirm** property to **True**.

© 2025 Rockwell Automation, Inc. All rights reserved.

Configure an Alarm Banner

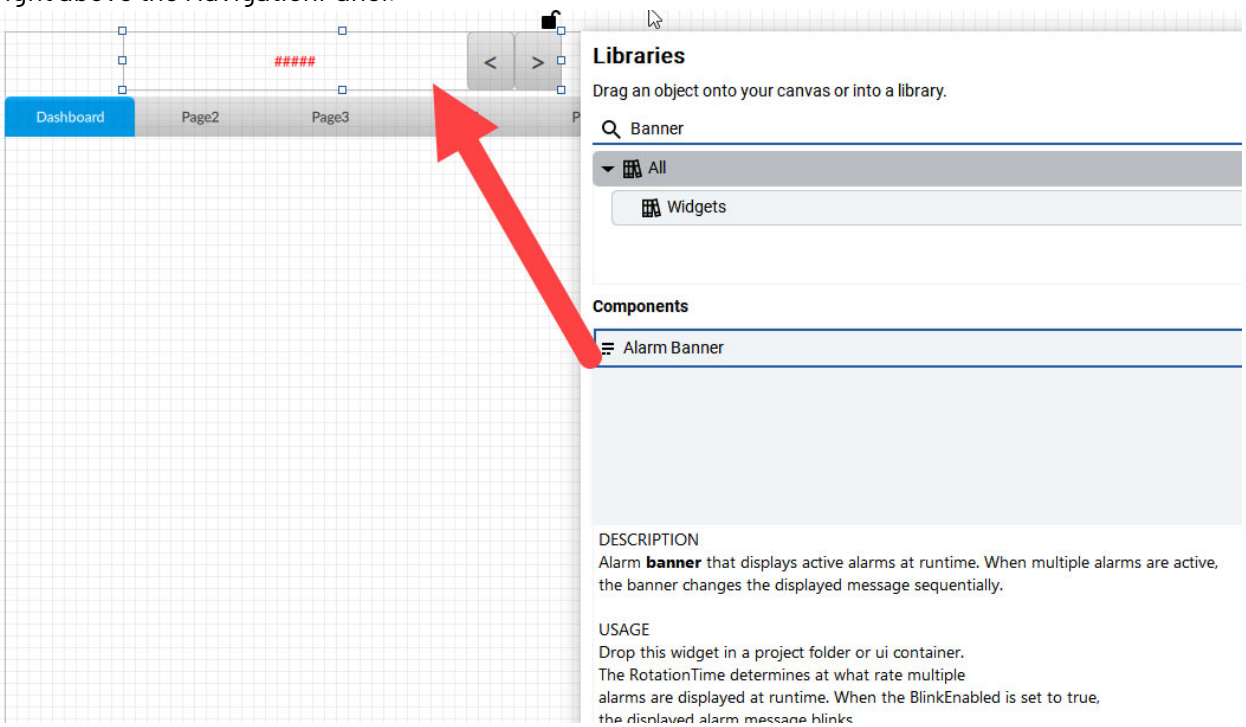
1. In Project view, expand **UI** and double-click **MainWindow (type).**



2. From the **Toolbar** select **Template Libraries** icon .

3. In the search field type "**banner**" to locate the **Alarm Banner** object.

4. Drag and drop the **Alarm Banner** widget onto the **MainWindow** and position it closer to the top right above the NavigationPanel.



5. Select **Close** to close the **Libraries** window.

© 2025 Rockwell Automation, Inc. All rights reserved.

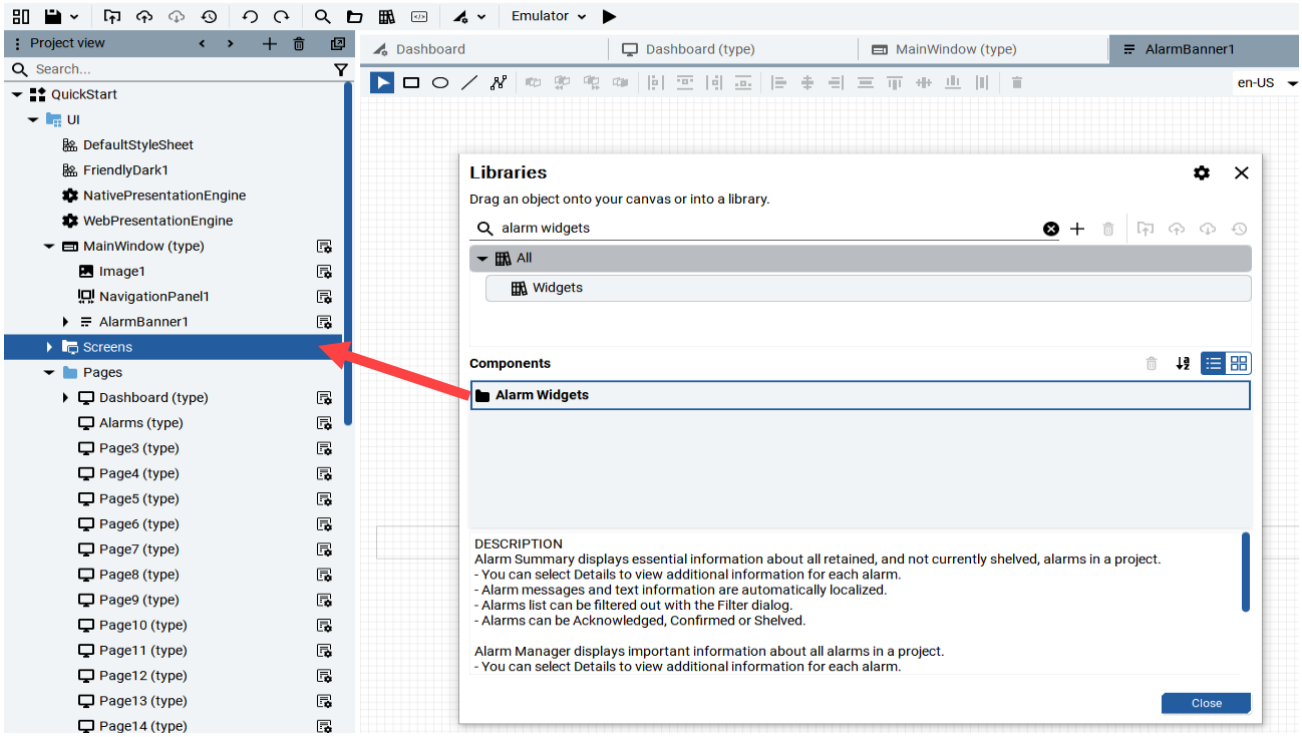
6. With the **Alarm Banner** widget selected, match the Size and layout properties as shown below.

Size and layout	
Horizontal alignment	Stretch
Vertical alignment	*Top*
Height	45
Left margin	5
Top margin	5
Right margin	250

Configure an Alarm Summary

Add an Alarm Summary that displays alarms status in real time.

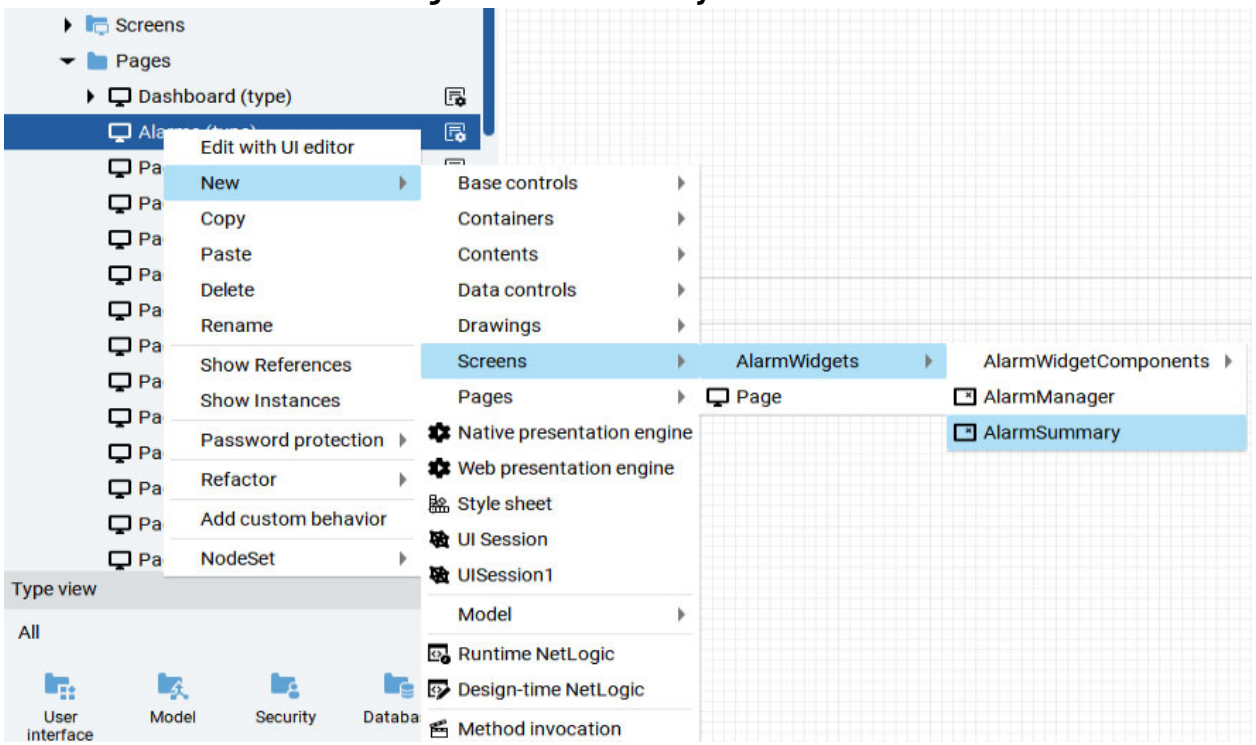
1. In Project view, expand the **Pages** folder and rename **Page2** to **Alarms**.
2. Double-click the **Alarms** page.
3. From the **Toolbar** select **Template Libraries** icon again.
4. Let's add a collection of alarm widgets called **Alarm Widgets** (type **Alarm Widgets** in the Search field to find it quicker). drag **Alarm Widgets** onto the **Screens** folder in Project view



5. Select **Close** to close the **Libraries** window.

© 2025 Rockwell Automation, Inc. All rights reserved.

6. In Project view, right-click on the newly renamed screen called **Alarms** under the *Pages* folder and select **New** > **Screens** > **AlarmWidgets** > **AlarmSummary.**

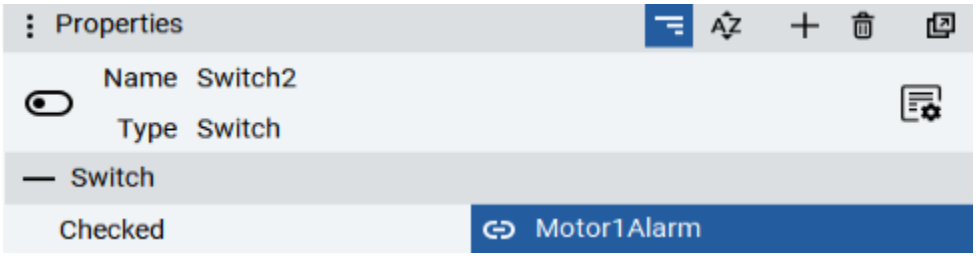


7. **Save** the project.

Simulate Alarm

Next, you will simulate these alarms and observe them in both the Alarm Banner and the Alarm Grid.

1. Right-click **Dashboard (type)** and select **New** > **Base controls** > **Switch**.
2. In Properties, use drag and drop to create a dynamic link between the **Checked** property and the **Motor1Alarm** variable.



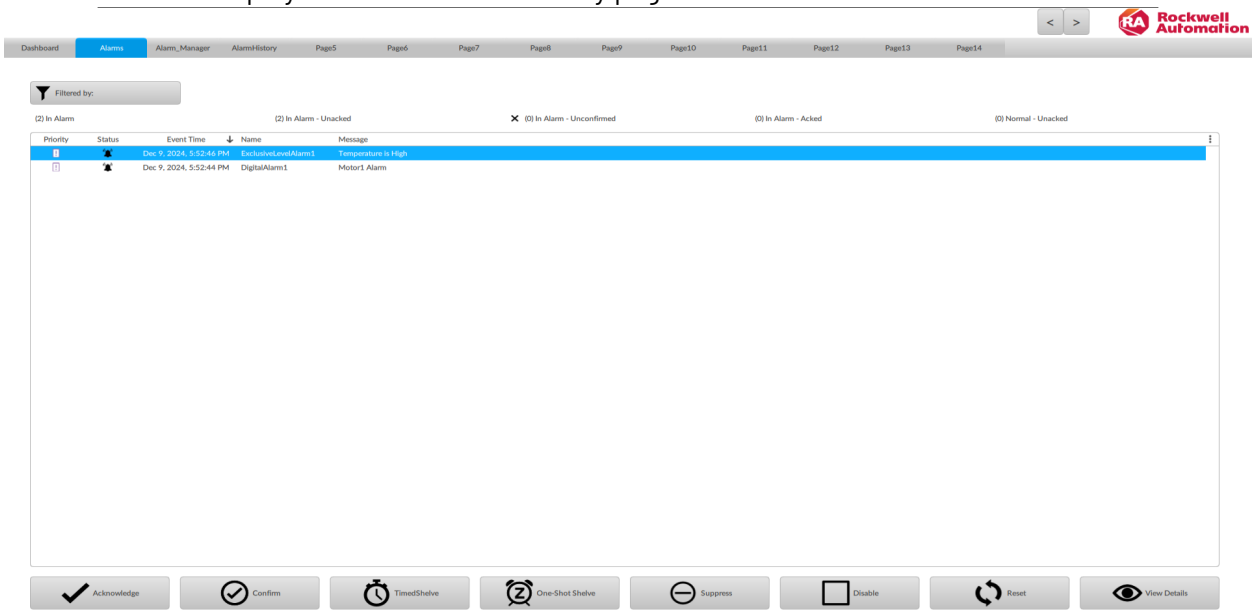
3. In the editor, arrange Switch2 so it is located below all other objects so you can easily locate it and simulate the alarm.
4. **Save** the project.
5. Next, you will simulate alarms and observe them in both the Alarm Banner and the Alarm Summary. Click the Run Emulator icon ▶.

© 2025 Rockwell Automation, Inc. All rights reserved.

6. From the Dashboard tab, click on the Motor 1 Alarm trigger switch.

Note: Observe how the alarm condition is displayed in the Alarm Banner.

7. Navigate to the Alarms page by clicking **Alarms** tab in the Navigation Panel top bar and confirm that the alarm is displayed in the Alarm Summary page.



8. Navigate back to Dashboard and move the Temperature Alarm trigger slider to a value higher than 80 and use the Next and Previous Alarm buttons as part of the Alarm Banner to scroll between the 2 active alarms.
9. Navigate back to the Alarms page and confirm that both alarms are displayed in the Alarm Summary page. **Acknowledge** each alarm using the alarm summary buttons.
10. To finish this exercise, deactivate both alarms. This is done by navigating to Dashboard, clicking the Motor 1 Alarm trigger switch off and moving the slider to a value less than 80.
11. Confirm the alarms no longer appear on the summary.
12. Close the **Emulator**.

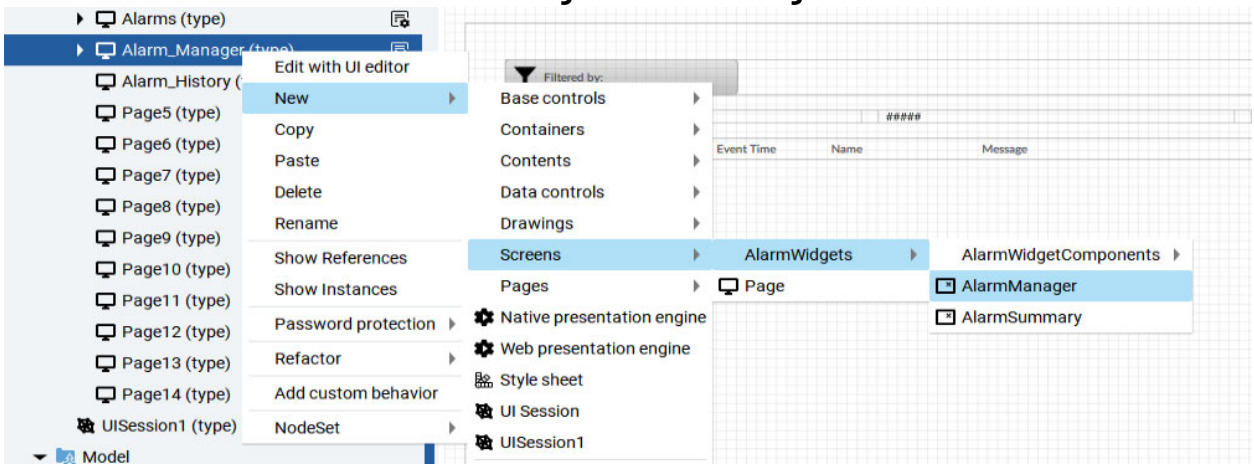
Configure an Alarm Manager

The Alarm manager widget lists all alarms configured in a system and their current states. Use it to monitor and interact with the alarms on the HMI device.

1. In Project view, expand the **Pages** folder and rename **Page3** to **Alarm_Manager**.

© 2025 Rockwell Automation, Inc. All rights reserved.

2. In Project view, right-click on the newly renamed screen called **Alarm_Manager** under the *Pages* folder and select **New** > **Screens** > **AlarmWidgets** > **AlarmManager.**

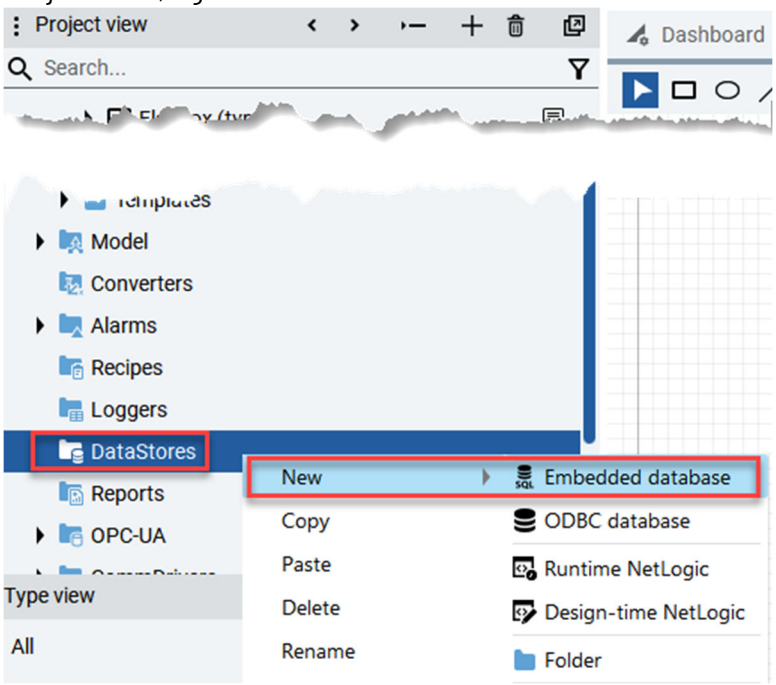


3. Click the Run Emulator icon ▶ and navigate to the Alarm Manager to see a list of alarms configured in the system. The Alarm manager widget lists all alarms regardless of their current states.

Configure Alarm History

In the next exercise you will configure a method that allows you to observe the alarm history. You will use simple steps to create a database that will store the alarm transactions and then associate this database with the Alarm Event Logger object. You will observe the alarm history in a ready-to-use widget.

1. Let's start with creating a local embedded database where the alarm history will be stored. In Project View, right-click the **DataStores** folder and select **New** > **Embedded database**.



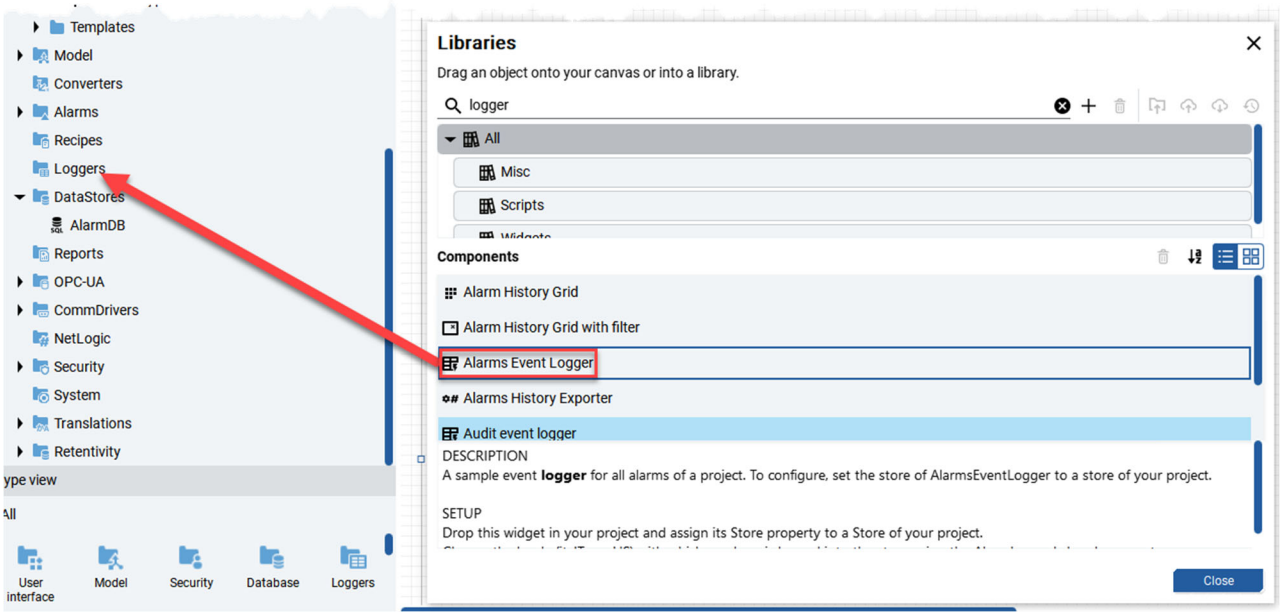
© 2025 Rockwell Automation, Inc. All rights reserved.

4. Mouse-over **EmbeddedDatabase1**, select the Edit icon , and rename it to **AlarmDB**. You just completed creating the database to use for alarm history in this lab. Please note that alarms can also be stored in an external ODBC database.

5. Next, we will use the Alarms Event Logger object to log alarms into the database we created earlier. From the **Toolbar**, select the **Template Libraries** icon .

6. In the Search field type **logger** to locate the **Alarms Event Logger** object.

7. Under **Components**, drag **Alarms Event Logger** onto the **Loggers** folder in Project view.

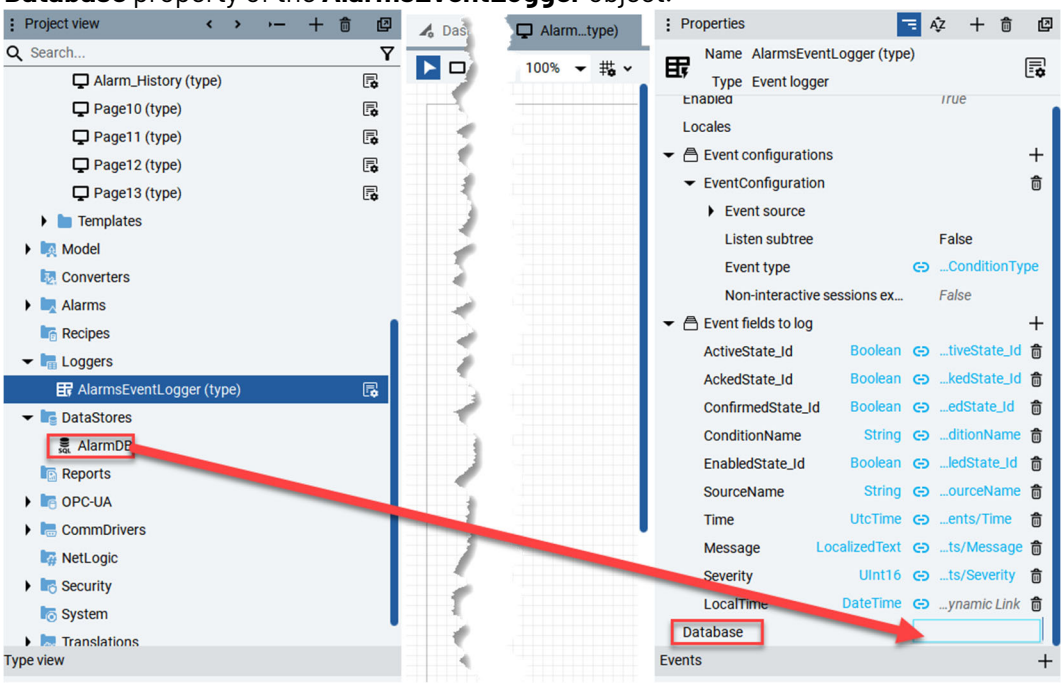


8. When dropping the object, select **Import as instance**. This will add the object instance based on the template into the project.

9. Select **Close** to close the **Libraries** window.

© 2025 Rockwell Automation, Inc. All rights reserved.

10. Let's link the logger we just created with the database. To create a dynamic link between Alarms Event Logger and the database AlarmDB, simply drag and drop the **AlarmDB** object to the **Database** property of the **AlarmsEventLogger** object.



11. So far, we have created the database and linked it to the Alarms Event Logger. The last step is to have a way to visualize the alarm history. In Project view, expand the **Pages** folder (if not expanded already) and rename **Page4** to **Alarm_History**.

12. Double-click the **Alarm_History** page. This page is currently blank.

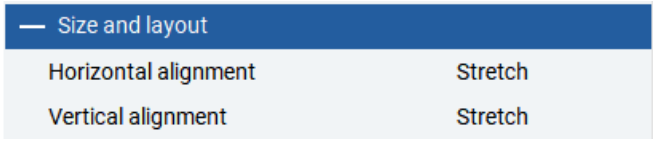
13. Select the **Template Libraries** icon again.

14. In the Search field type **grid** to locate the **Alarm History Grid with filter** object.

15. Drag and drop this object onto the **Alarm_History** page.

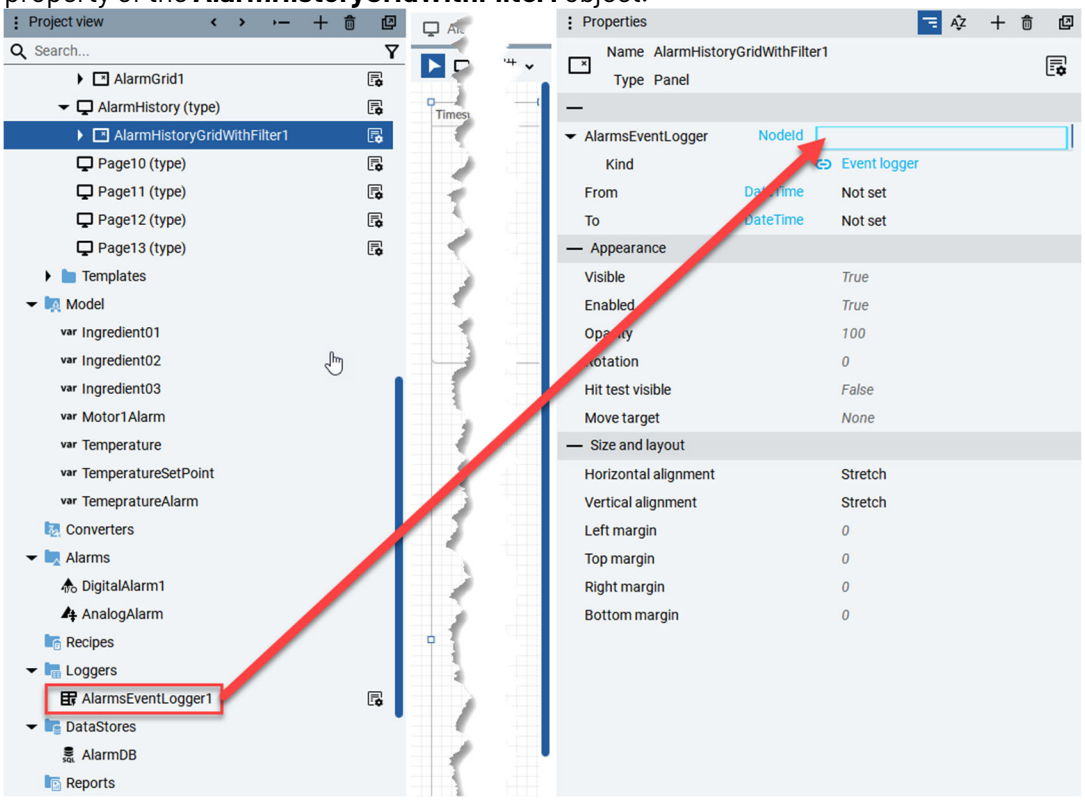
16. Select **Close** to close the **Libraries** window.

17. In **Properties**, for better viewing at runtime, adjust Size and layout of the object to stretch horizontally and vertically keeping all margins set to 0.



© 2025 Rockwell Automation, Inc. All rights reserved.

18. To create a dynamic link between Alarms History Grid and Alarm Event Logger that you created in previous steps, drag and drop the **AlarmsEventLogger1** object to the **AlarmsEventLogger** property of the **AlarmHistoryGridWithFilter1** object.



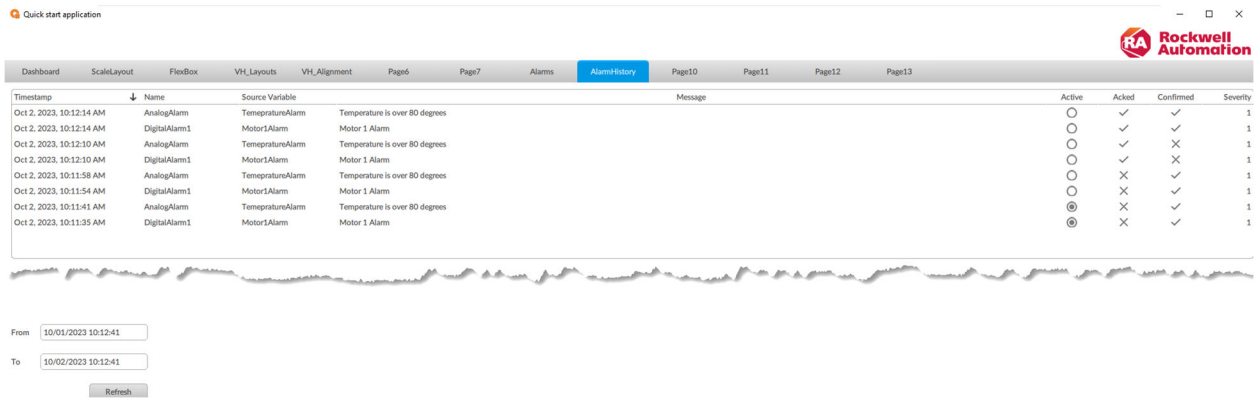
Visualize Alarm History

Next, you will generate the same alarm transactions as before and observe them in Alarm History.

1. Click the **Run Emulator** icon ▶.
2. Click on the Motor 1 Alarm trigger switch and move the Temperature Alarm trigger slider all the way to the right.
3. Click on the Motor 1 Alarm trigger switch again to deactivate the digital alarm.
4. Move the Temperature Alarm Trigger slider all the way to the left.
5. Navigate to the Alarms page by clicking **Alarms** tab in the Navigation Panel top bar and **Acknowledge** each alarm.

© 2025 Rockwell Automation, Inc. All rights reserved.

6. Click the **Alarm_History** tab in the Navigation Panel top bar to open the page with the Alarm History Grid object. Observe the sequence of actions that you performed in the previous steps e.g. the timestamped records indicate when the alarms went off and when they were deactivated as well as the acknowledgment and confirmation of alarms.



7. Close the **Emulator**.

8. (Optional) Can you determine the total feature tokens needed to run this project in production? Which components in this project are impacting the runtime entitlement size?

You have completed the Alarming lab.

© 2025 Rockwell Automation, Inc. All rights reserved.

Configure Datalogging (15 Minutes)

Objectives

- Use a wizard to create a datalogger
- Use a DataGrid to show the data on the screen
- Create a trend and configure as Realtime Data
- Create a trend and configure as Historical Data

Scenario

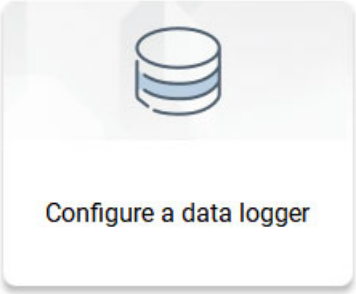
In this section you will create a data log using a wizard and visualize the data using a DataGrid and trend objects.

Note: Screenshots may differ depending on what optional sections have been completed previously.

Lab Procedure

Configure Data logger:

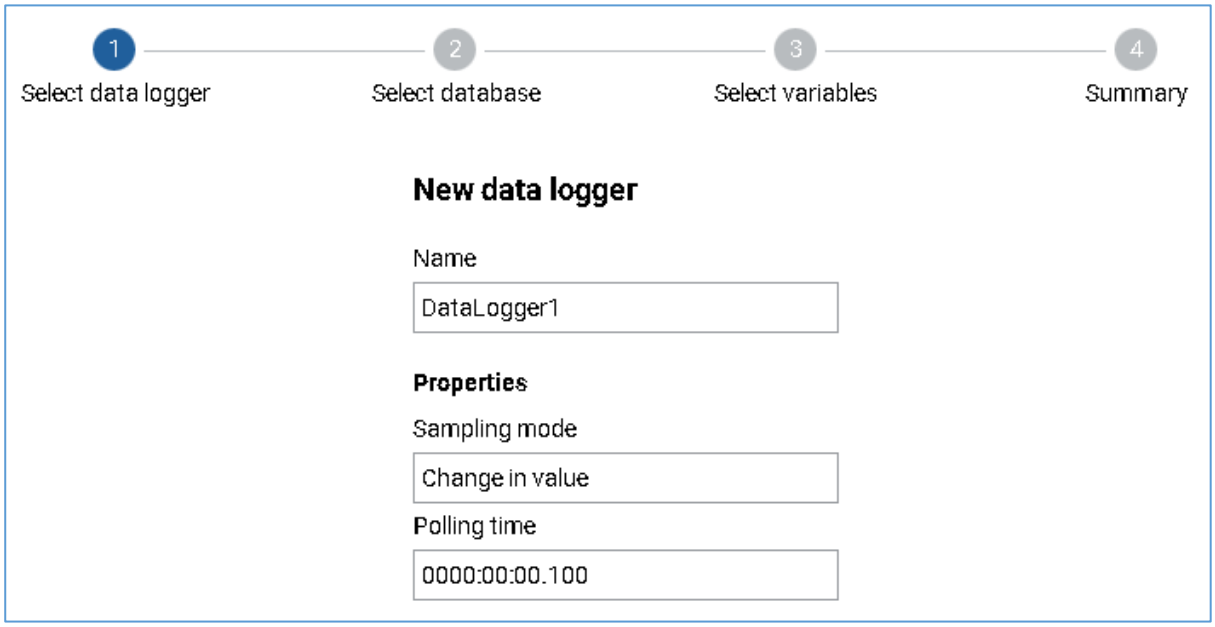
1. From the FactoryTalk Optix Studio toolbar, select the **Open dashboard page** icon .
2. In the central pane, select **Configure a data logger**.



3. Select **New data loggers** to create a data logger.

© 2025 Rockwell Automation, Inc. All rights reserved.

4. Select a *Sampling mode* of **Change in value**. Sample all monitored variables with values that changed from the previous sampling.
5. For *Polling time*, confirm that the sampling properties is set to: **0000:00:00.100**.



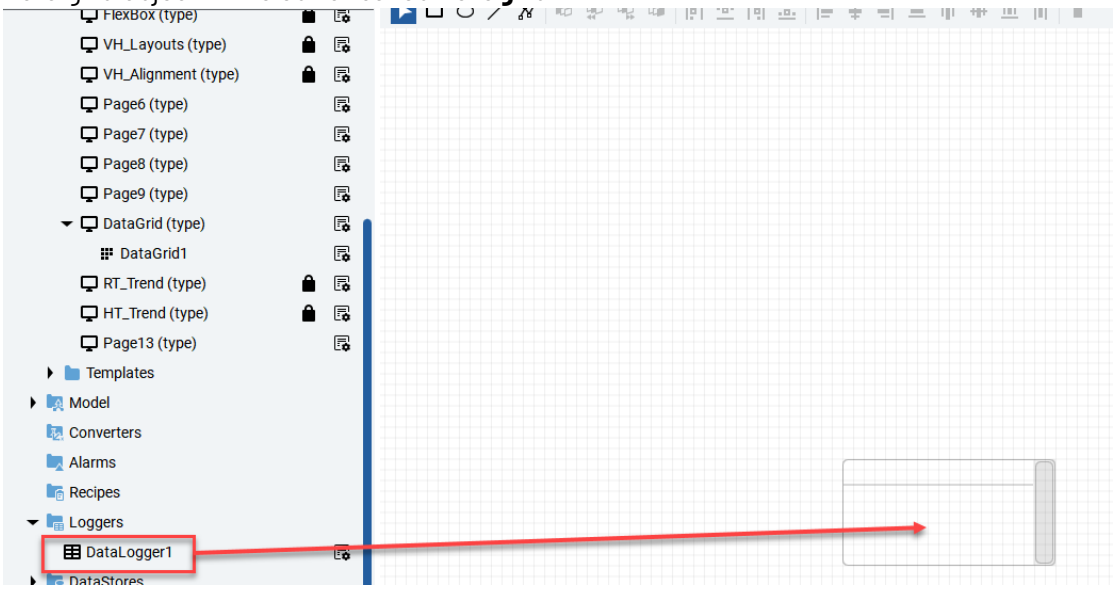
6. Select **Next** and **Next** again.
7. Select **New database.**
8. Change the Type as **Local InfluxDB Connection**
9. Edit **Name** and set it to **DataLogDB** and select **Next** and select **Next** again.
10. Click **Next** and select the **Temperature** variable under the Model folder to log.
11. Select **Next**.
12. Wait for the successful message and **Exit**.

You can display the data logger and event logger data in a table. The Data grid object shows the data stored by the logger in the database.

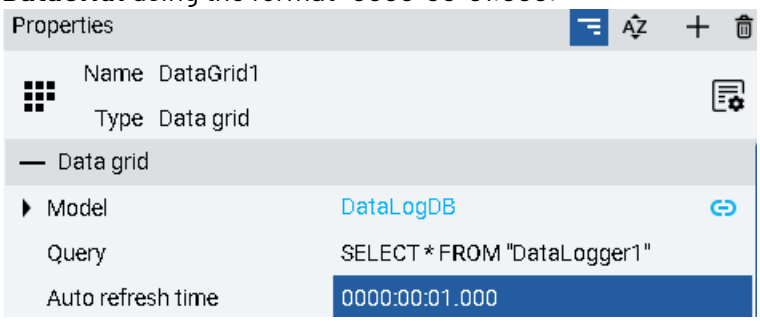
13. In the Project view pane, expand UI > Pages and rename **Page5 (type)** to **Data_Grid**
14. Right-click on **Data_Grid** page and select **New** > **Data controls** > **Data grid**.
15. Double-click the **DataGrid1** object to open it in the graphic editor pane (in the middle of FactoryTalk Optix Studio.)

© 2025 Rockwell Automation, Inc. All rights reserved.

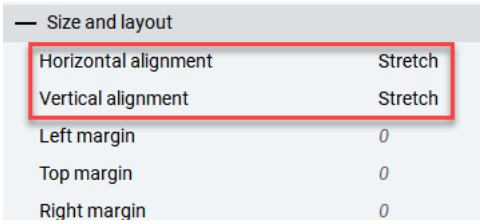
16. From the Project view pane, drag the data logger in the folder Loggers called **DataLogger1** to the Data grid object in the editor called **Datagrid1**.



17. To display the data updated in real time at runtime, configure the **Auto refresh time** property of **DataGrid1** using the format: 0000:00:01.000.



18. In **Properties**, scroll down **to Size and layout** section set **Horizontal alignment** and **Vertical alignment** to **Stretch** and keep all margins set to "0".

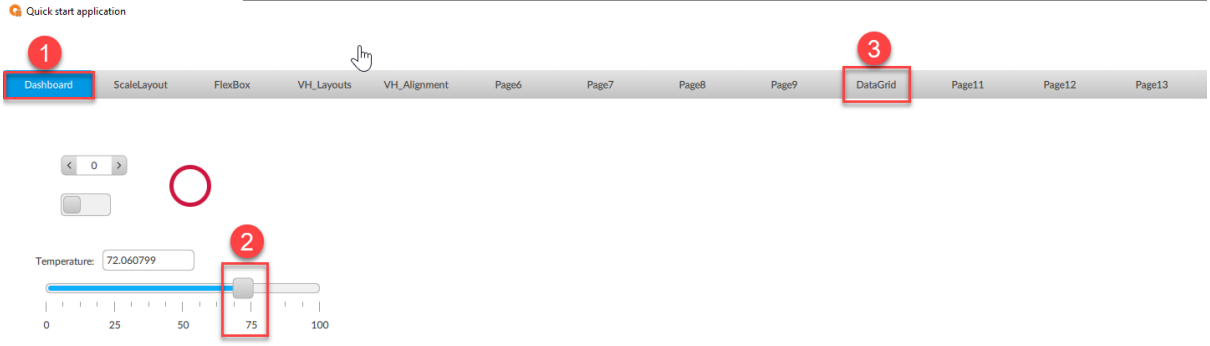


19. **Save** the project.

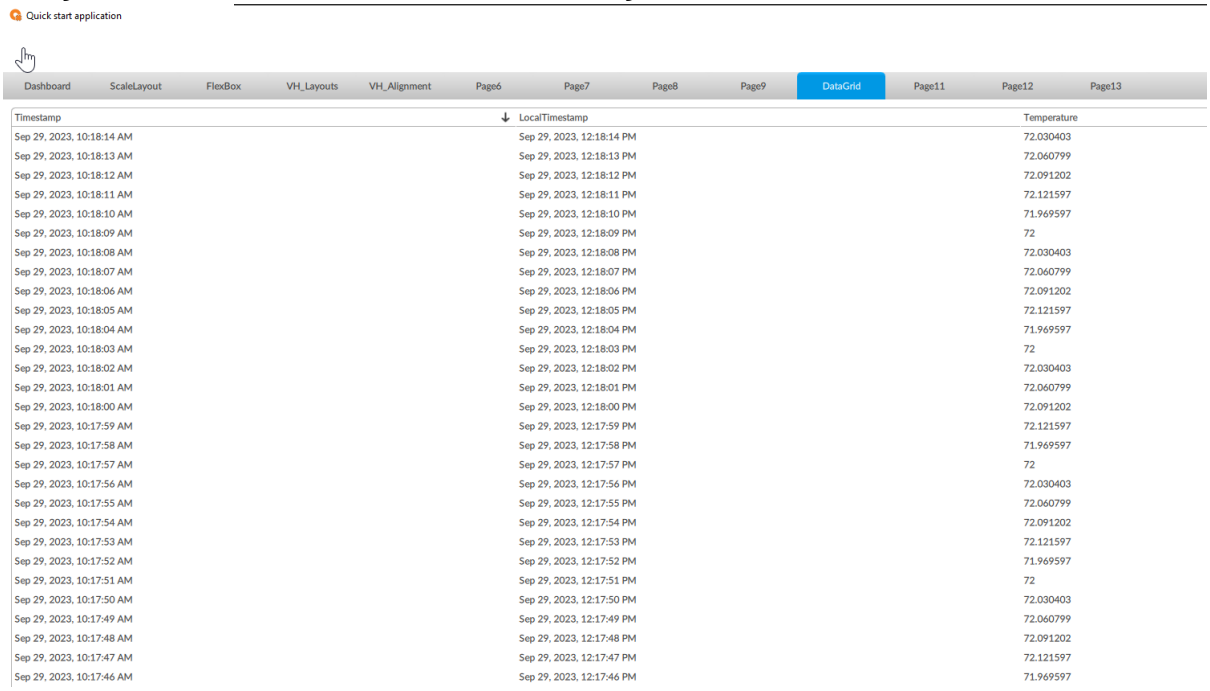
20. From the toolbar, with Emulator still displayed, click the **Run on Emulator** icon ▶.

© 2025 Rockwell Automation, Inc. All rights reserved.

21. At runtime, under **Dashboard**, change the temperature setpoint value and you will see the **temperature** change automatically.



22. Navigate to **Data_Grid** and observe the value changes in the Data Grid.



Timestamp	LocalTimestamp	Temperature
Sep 29, 2023, 10:18:14 AM	Sep 29, 2023, 12:18:14 PM	72.030403
Sep 29, 2023, 10:18:13 AM	Sep 29, 2023, 12:18:13 PM	72.060799
Sep 29, 2023, 10:18:12 AM	Sep 29, 2023, 12:18:12 PM	72.091202
Sep 29, 2023, 10:18:11 AM	Sep 29, 2023, 12:18:11 PM	72.121597
Sep 29, 2023, 10:18:10 AM	Sep 29, 2023, 12:18:10 PM	71.969597
Sep 29, 2023, 10:18:09 AM	Sep 29, 2023, 12:18:09 PM	72
Sep 29, 2023, 10:18:08 AM	Sep 29, 2023, 12:18:08 PM	72.030403
Sep 29, 2023, 10:18:07 AM	Sep 29, 2023, 12:18:07 PM	72.060799
Sep 29, 2023, 10:18:06 AM	Sep 29, 2023, 12:18:06 PM	72.091202
Sep 29, 2023, 10:18:05 AM	Sep 29, 2023, 12:18:05 PM	72.121597
Sep 29, 2023, 10:18:04 AM	Sep 29, 2023, 12:18:04 PM	71.969597
Sep 29, 2023, 10:18:03 AM	Sep 29, 2023, 12:18:03 PM	72
Sep 29, 2023, 10:18:02 AM	Sep 29, 2023, 12:18:02 PM	72.030403
Sep 29, 2023, 10:18:01 AM	Sep 29, 2023, 12:18:01 PM	72.060799
Sep 29, 2023, 10:18:00 AM	Sep 29, 2023, 12:18:00 PM	72.091202
Sep 29, 2023, 10:17:59 AM	Sep 29, 2023, 12:17:59 PM	72.121597
Sep 29, 2023, 10:17:58 AM	Sep 29, 2023, 12:17:58 PM	71.969597
Sep 29, 2023, 10:17:57 AM	Sep 29, 2023, 12:17:57 PM	72
Sep 29, 2023, 10:17:56 AM	Sep 29, 2023, 12:17:56 PM	72.030403
Sep 29, 2023, 10:17:55 AM	Sep 29, 2023, 12:17:55 PM	72.060799
Sep 29, 2023, 10:17:54 AM	Sep 29, 2023, 12:17:54 PM	72.091202
Sep 29, 2023, 10:17:53 AM	Sep 29, 2023, 12:17:53 PM	72.121597
Sep 29, 2023, 10:17:52 AM	Sep 29, 2023, 12:17:52 PM	71.969597
Sep 29, 2023, 10:17:51 AM	Sep 29, 2023, 12:17:51 PM	72
Sep 29, 2023, 10:17:50 AM	Sep 29, 2023, 12:17:50 PM	72.030403
Sep 29, 2023, 10:17:49 AM	Sep 29, 2023, 12:17:49 PM	72.060799
Sep 29, 2023, 10:17:48 AM	Sep 29, 2023, 12:17:48 PM	72.091202
Sep 29, 2023, 10:17:47 AM	Sep 29, 2023, 12:17:47 PM	72.121597
Sep 29, 2023, 10:17:46 AM	Sep 29, 2023, 12:17:46 PM	71.969597

23. Close the **Emulator**.

Note: **New!** Starting with FactoryTalk Optix version 1.4, you can add an external or remote InfluxDB time-series database for high-performance data operations.

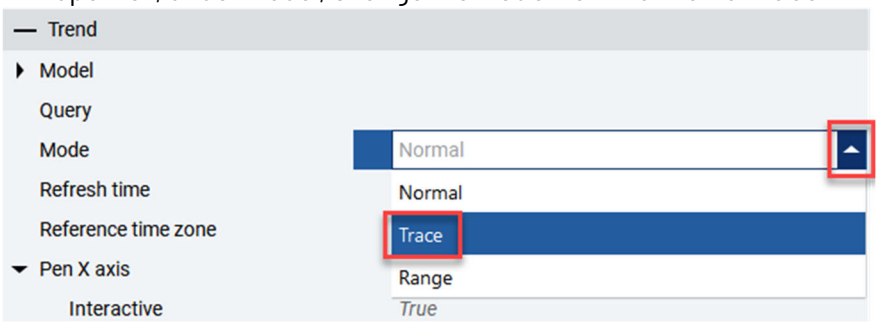
Note: **New!** Starting with FactoryTalk Optix version 1.5, support for store and forward was added to temporarily store data locally during connection disruption or when the external storage is unavailable

Note: **New!** Starting with FactoryTalk Optix version 1.5, support for InfluxDB as an internal/embedded storage solution was added, enabling time-series data management as a built-in capability.

© 2025 Rockwell Automation, Inc. All rights reserved.

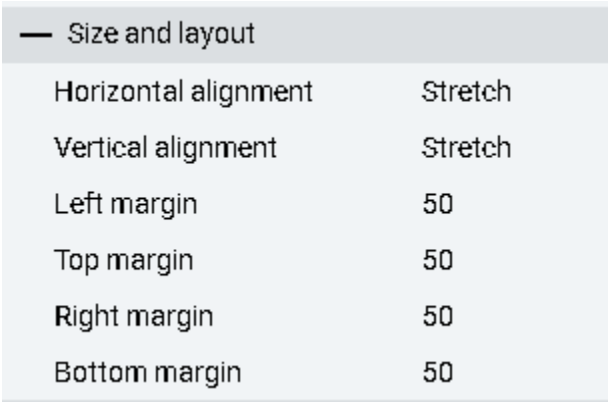
Realtime Trending:

1. In the Project view pane, under **UI** > **Pages**, rename **Page6 (type)** to **RT_Trend**
2. Right-click on **RT_Trend** page and select **New** > **Data controls** > **Trend**.
3. Double-click the **Trend1** object to open it in the editor and access its properties pane on the right.
4. In Properties, under Model, Change the **Mode** from Normal to **Trace**.



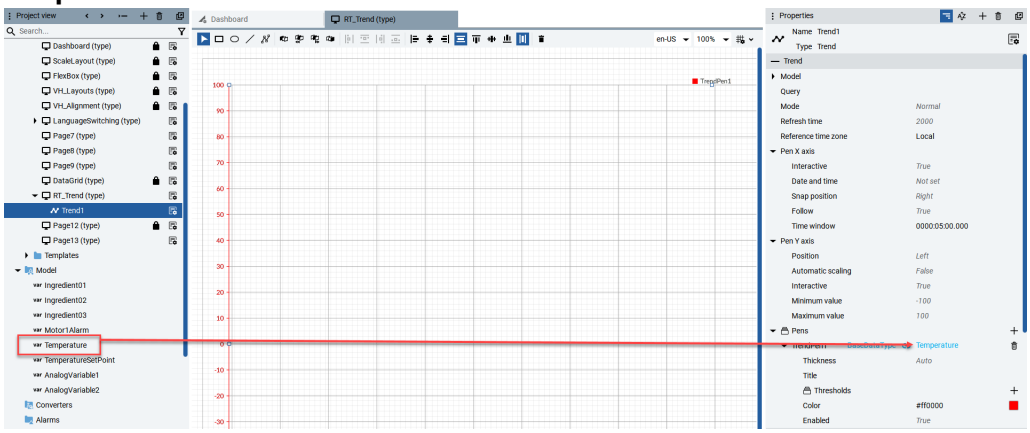
5. Change the Time window from 5 seconds to 5 minutes. This is down by setting Time window to 5 minutes using the following format **0000:05:00:000**

6. Set **Horizontal alignment** and **Vertical alignment** to **Stretch**.
7. Set **Left, Top, Right** and **Bottom margins** to **50**.

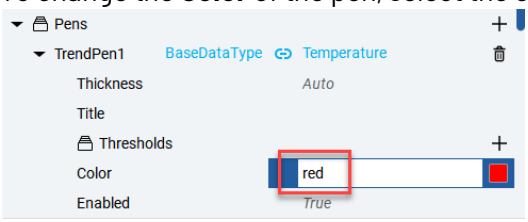


© 2025 Rockwell Automation, Inc. All rights reserved.

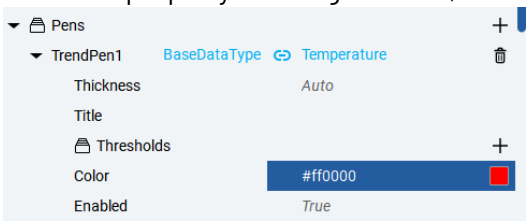
8. Under **Pens**, drag and drop to create a dynamic link between **TrendPen1** and the pen data source **Temperature** variable.



9. To change the **Color** of the pen, select the edit icon ✎, enter **Red**, and hit **enter** on the keyboard.



The Color property is changed to red, and the appropriate hex value of #ff0000 is used.



10. Change the **Thickness** property to **4**.

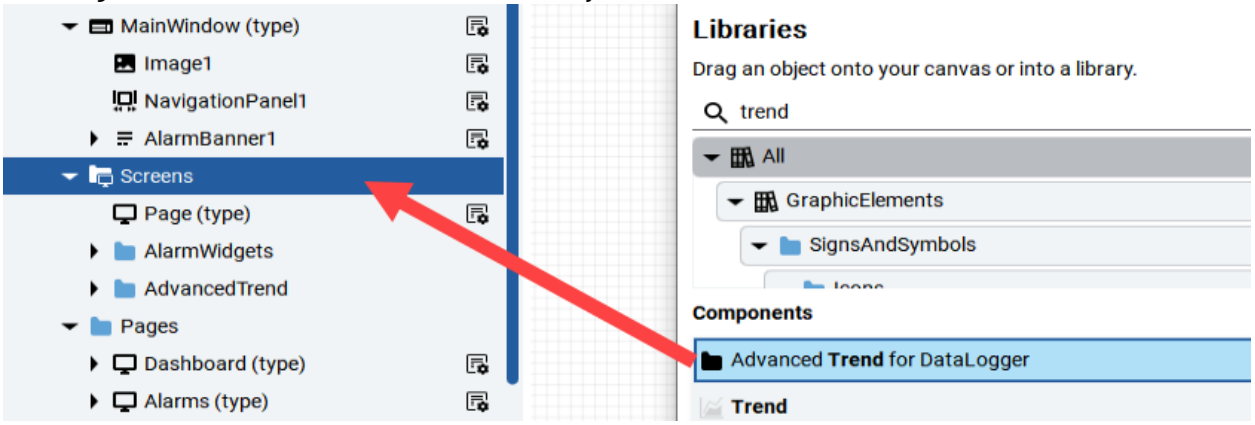
11. Click the **Run on Emulator** icon ▶.
In the emulator, change the temperature to any value and navigate to RT_Trend. The trend will start populating data. Since this is a real-time trend only, every time you navigate away and go back to RT_Trend, the trend object starts from the time you navigate to it. There is no history to show yet.

© 2025 Rockwell Automation, Inc. All rights reserved.

Historical Trending:

1. In this section of the lab, we will use the Advanced Trend widget available from Libraries. From the **Toolbar** select **Template Libraries** icon .

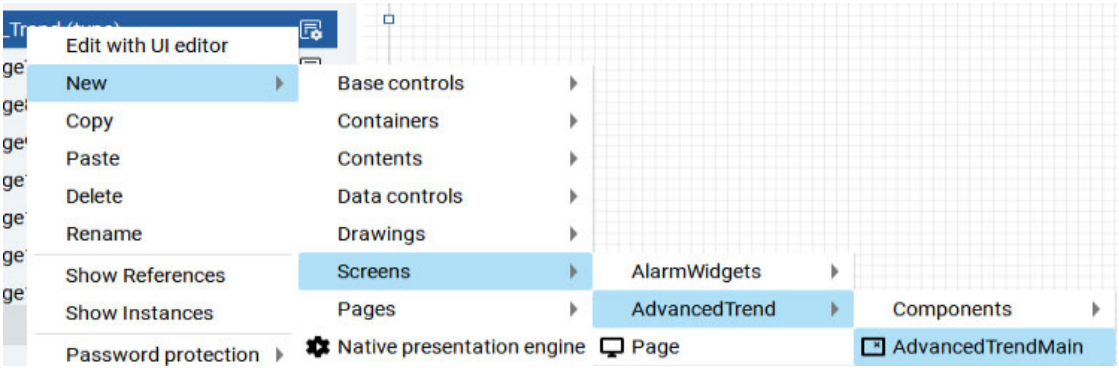
2. Locate the widget called **Advanced Trend for DataLogger** using the search field. Drag and drop the widget under onto the **Screens** folder in Project view.



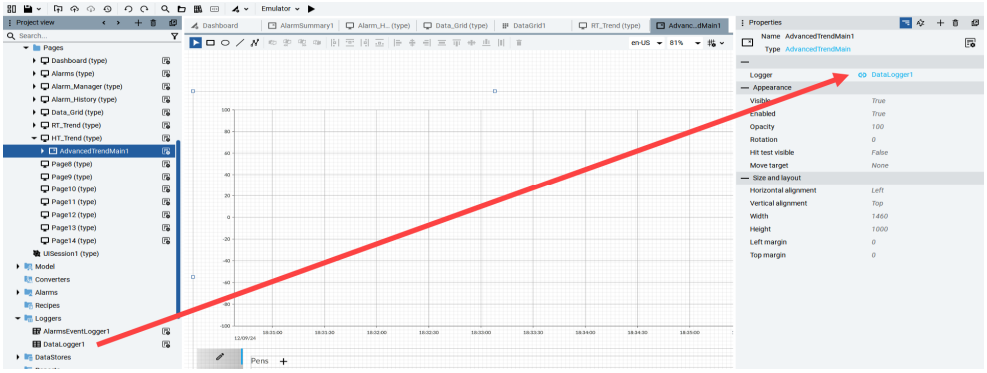
3. Select **Close** to close the **Libraries** window.

4. In the Project view pane, under **UI** > **Pages**, rename **Page7 (type)** to **HT_Trend**

5. Right-click on **HT_Trend** page and select **New** > **Screens** > **AdvancedTrend** > **AdvancedTrendMain**

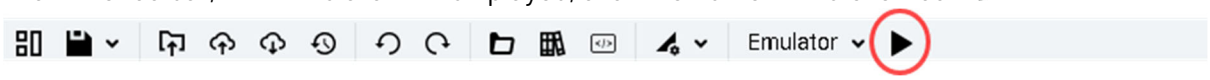


6. Use drag and drop to drag the **DataLogger1** we created earlier under Loggers and drop it on the **Logger** property of the Advanced Trend chart.



© 2025 Rockwell Automation, Inc. All rights reserved.

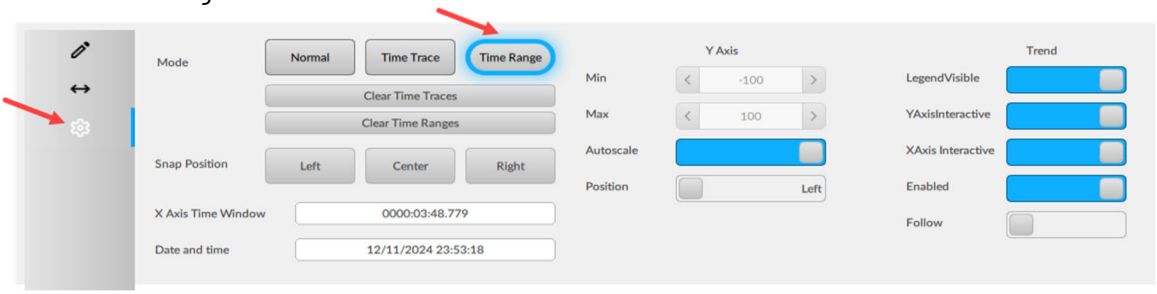
7. From the toolbar, with Emulator still displayed, click the **Run on Emulator** icon ▶.



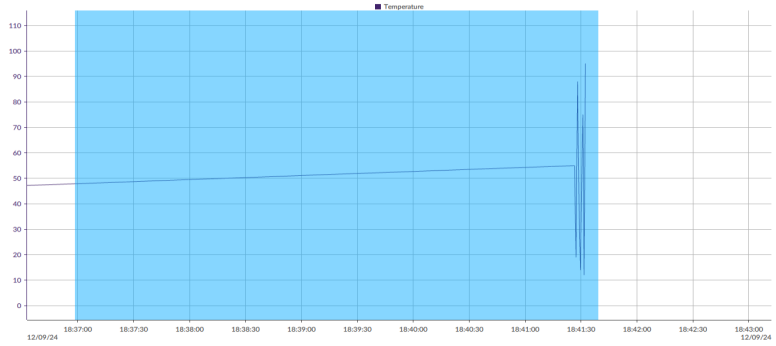
8. From Dashboard, change the temperature value using the gauge. Repeat this step by use different temperature values few times.

9. Navigate to **HT_Trend** to view the historical time trend. All historical values in the last few minutes are plotted.

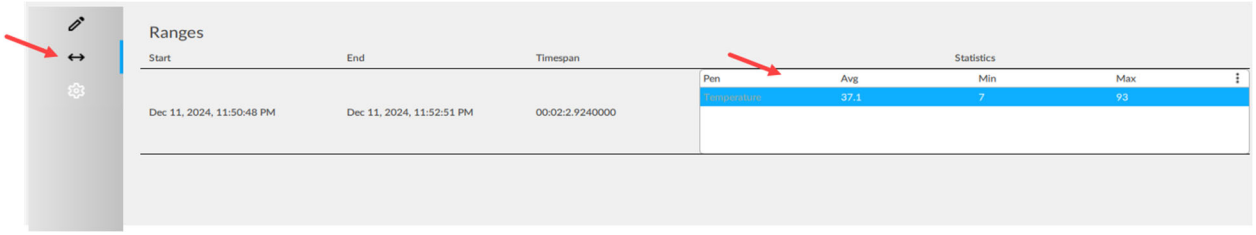
10. This widget enables extracting statistical data from a time range. Click the configure icon ⚙ and select Time Ranges.



11. On the trend, starting from the left, click, drag to the right and release when you have covered a large area with different values for the Temperature variable.



12. To see the statistical values, select the ranges tab show using the ↔ icon.



13. Close the **Emulator**.

14. (Optional) Can you determine the total feature tokens needed to run this project in production? Which components in this project are impacting the runtime entitlement size?

You have completed the Datalogging lab.

© 2025 Rockwell Automation, Inc. All rights reserved.

Reporting (20 Minutes)

Objectives

- Create stylesheets for your report
- Create a report with header, footer and main sections
- Generate a PDF report

Scenario

In this section you will learn how to create reports using standard objects within FactoryTalk Optix and how to create PDF reports based on the stylesheet that you selected.

Make sure to do the Datalogging section first because we will use the data logs in this section.

Use a Reports object to design, layout, and generate PDF reports. Reports can include:

- Data extracted from a FactoryTalk Optix Application to include process data and statistical data
- A database to include data recorded by an event logger.

Report components include:

1. **Header**: Specifies the height and content of the header in the PDF. Headers generally contain logos, contact information, print references such as page numbers, and the date and time the report was generated.
2. **Sections**: Sets the content of the report. Sections contain:
    - Panel section: This is a container for:
        - Static values such as labels, images, panels, and rectangles
        - Dynamic values such as controller variable values, and the values of alarm events collected in real time
        - Graphic objects
    - Data grid section: Represents the data read from a database and selected from the data in the application in the form of a table.
    - Page breaks: Introduces a page break between report sections.
3. **Footer**: Specifies the height and content of the footer in the PDF. Footers generally contain logos, contact information, page numbers, and the date and time the report was generated.

You can customize the appearance of report contents with the Report Stylesheet and save reports to a specific location.

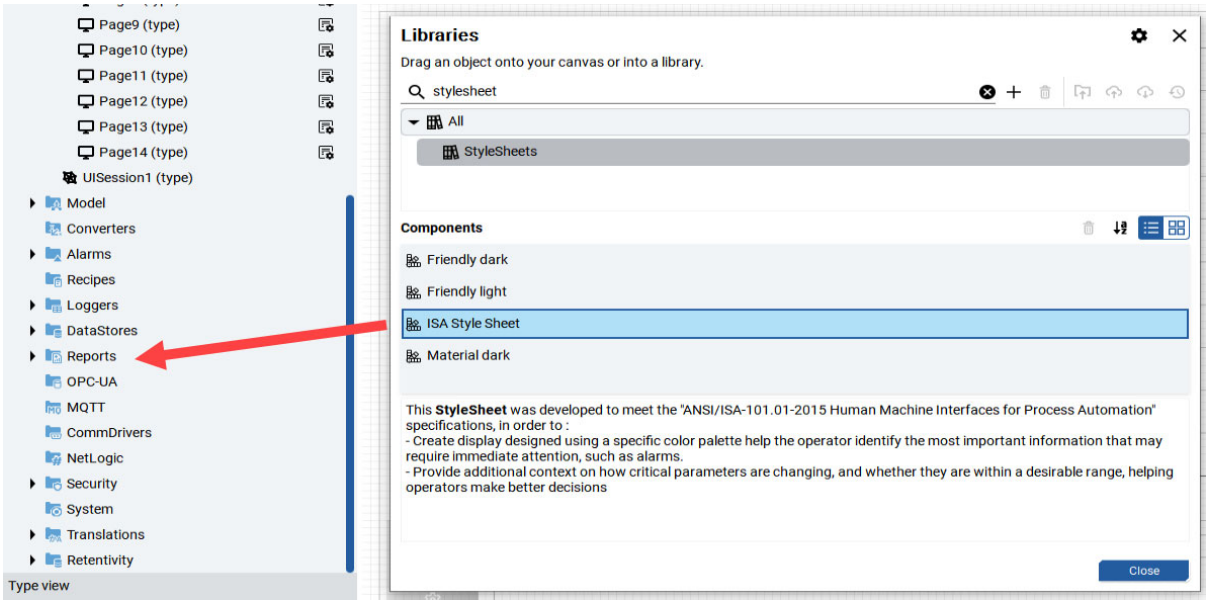
Lab Procedure

1. To have different custom style Reports we will need 2 different stylesheets. Instead of creating them from scratch, we will use an existing Stylesheet and bring another from Libraries. From the **Toolbar** select **Template Libraries** icon  .

2. Search for “stylesheets” to locate the ISA Style Sheet.

© 2025 Rockwell Automation, Inc. All rights reserved.

3. Drag and drop **ISA Style Sheet** onto the **Reports** folder in the Project view**.**



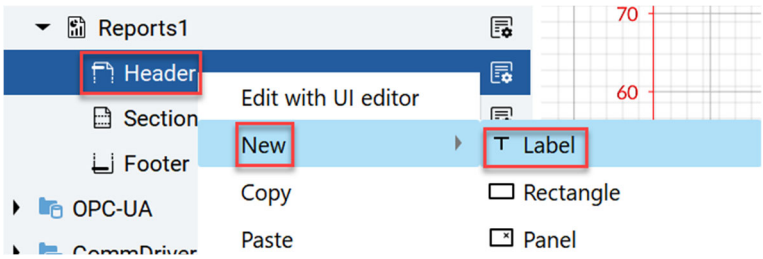
4. **Import as instance**.

5. Select **Close** to close the **Libraries** window.

6. Right click the **Reports** folder and select **New** > **Reports**.

7. In Project view, Expand **Reports1** and you will see the 3 components a Report contains. Header, Sections and Footer. Double Click on **Header** and you will see the Header in the Editor part of the window.

8. Right Click **Header** and add a **Label**.



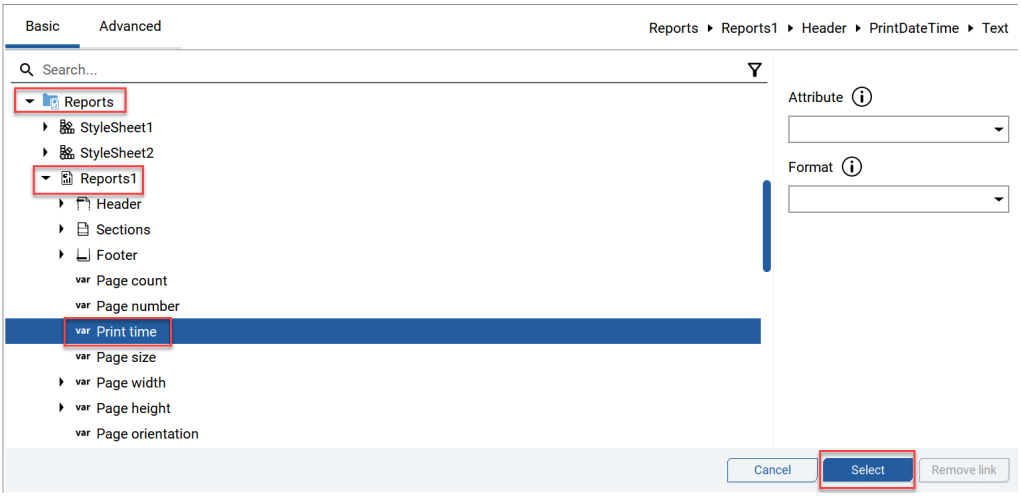
9. Click on the newly created **Label1** and change the Properties according to the following

Horizontal alignment: Right
Vertical alignment: Center
Width: 2.5
Right Margin: 0.5

© 2025 Rockwell Automation, Inc. All rights reserved.

10. Mouse-over the **Text** property and click the Add Dynamic Link icon .

11. Browse to **Reports** > **Reports1** > **Print time** and click **Select** (the field will be populated with *../../../PrintDateTime*)



The properties pane will look like the screen capture below.

12. We will now create content for the **Footer**. Right click **Footer** and add a **Label.**

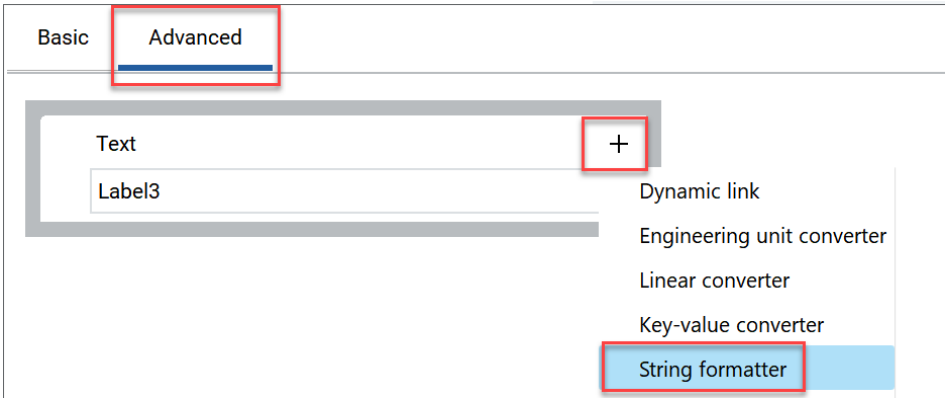
13. Click on the newly created **Label** and change the Properties according to the following

Horizontal alignment: center
Vertical alignment: Center
Width: 2.5
Text Horizontal alignment: Center aligned

14. Next, we will create a complex dynamic link for the Text property. Mouse-over the **Text** property and click the **Add Dynamic Link** icon .

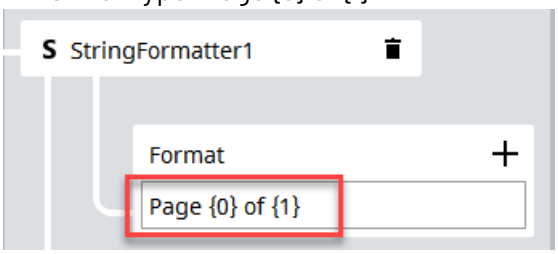
15. In the dynamic link browser, click the **Advanced** tab.

16. You are now looking at the complex dynamic link editor. Click + and select **String formatter**.



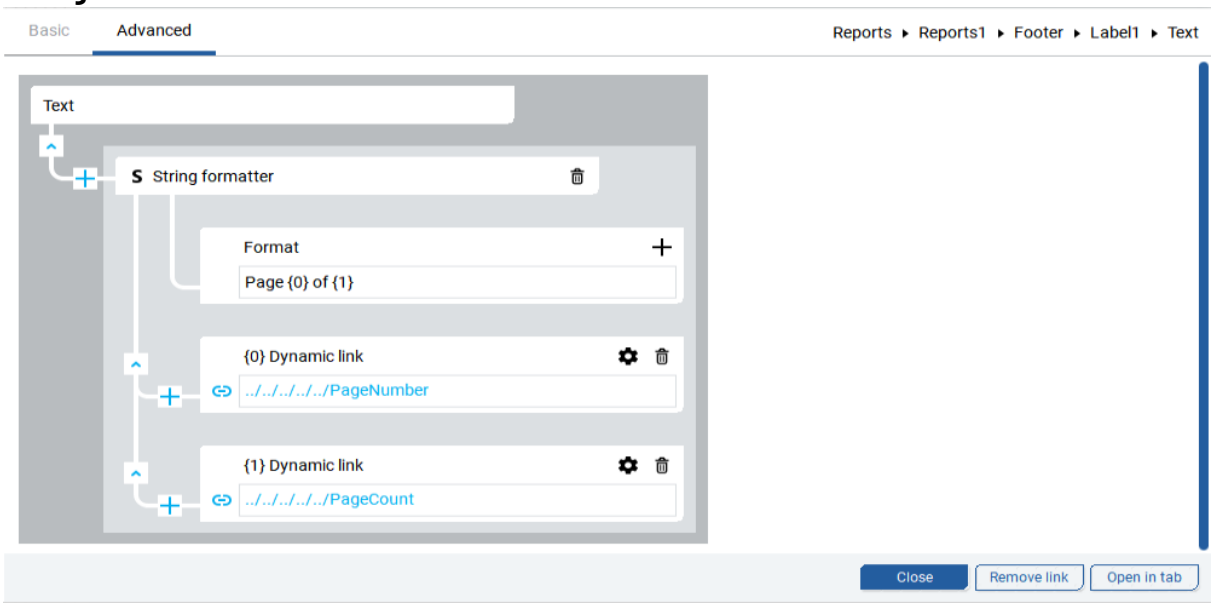
© 2025 Rockwell Automation, Inc. All rights reserved.

17. In **Format** type: Page {0} of {1}



18. For *{0} Dynamic link*, click the **Change Dynamic Link** icon and browse to **Reports → Report1 → Page Number.**

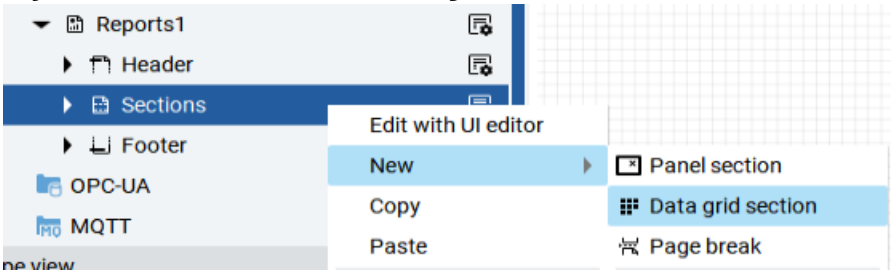
19. For *{1} Dynamic link*, click the **Change Dynamic Link** icon and browse to **Reports → Report1 → Page Count.**



20. **Close** when done.

Sections

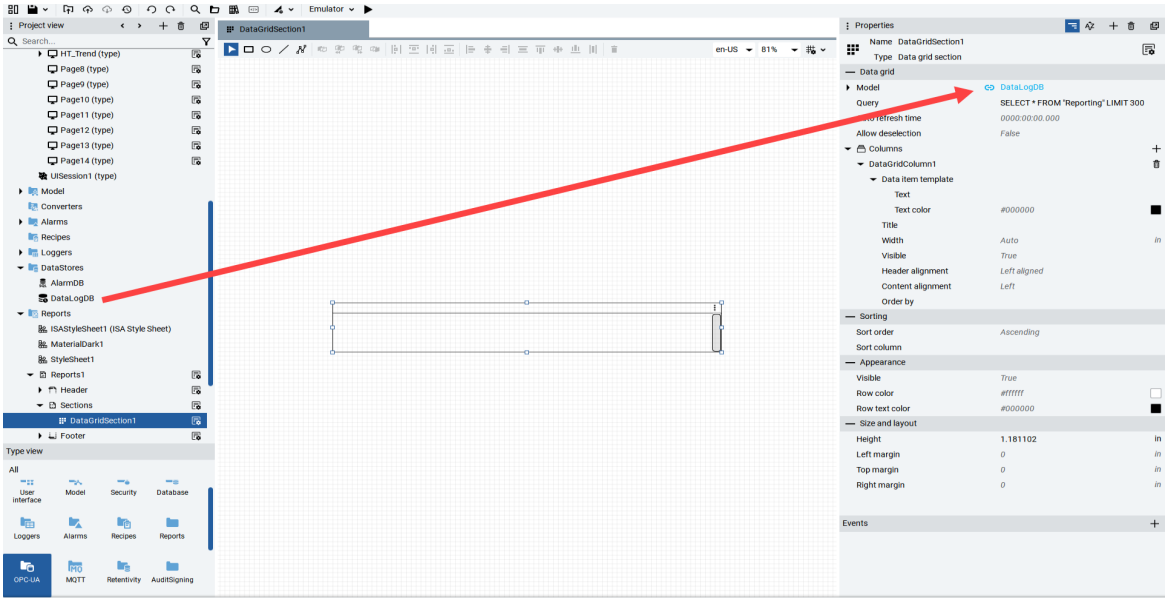
1. Right Click Sections and add a **Data grid section**.



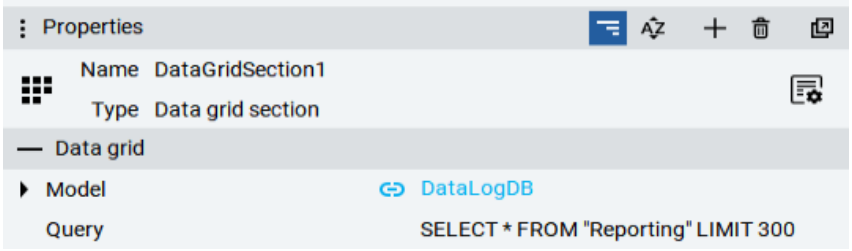
2. Select **DataGridSection1** to access its properties.

© 2025 Rockwell Automation, Inc. All rights reserved.

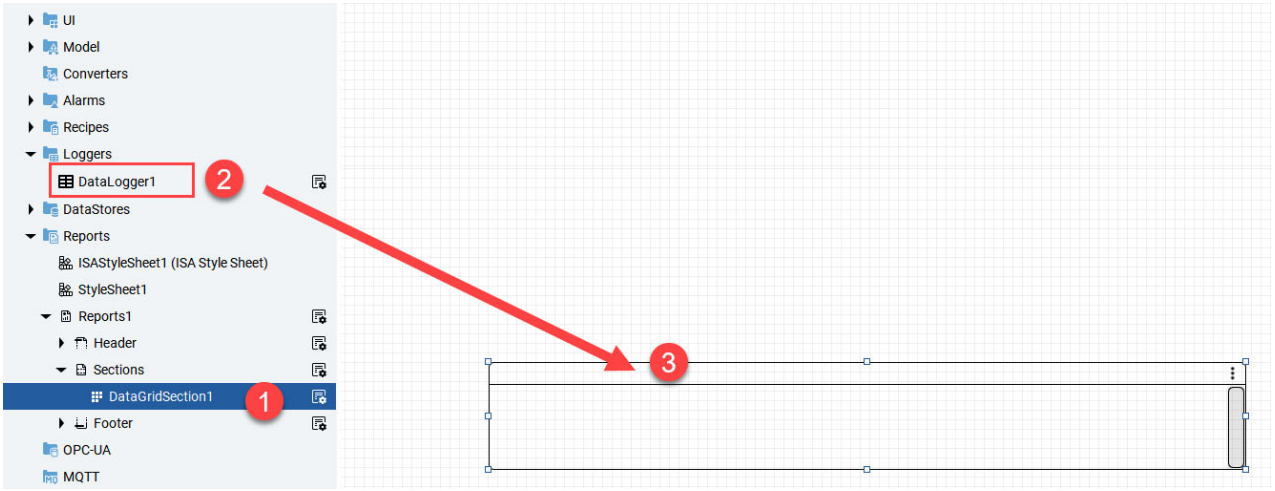
3. To get data in our Report first we must define where the data is coming from. We will use the local **Database** that you created in the Datalogging section. Expand **Datastores**, select **DataLogDB**. drag it to the **Model** property.



4. Update the **Query** property with **SELECT * FROM "Reporting" LIMIT 300**



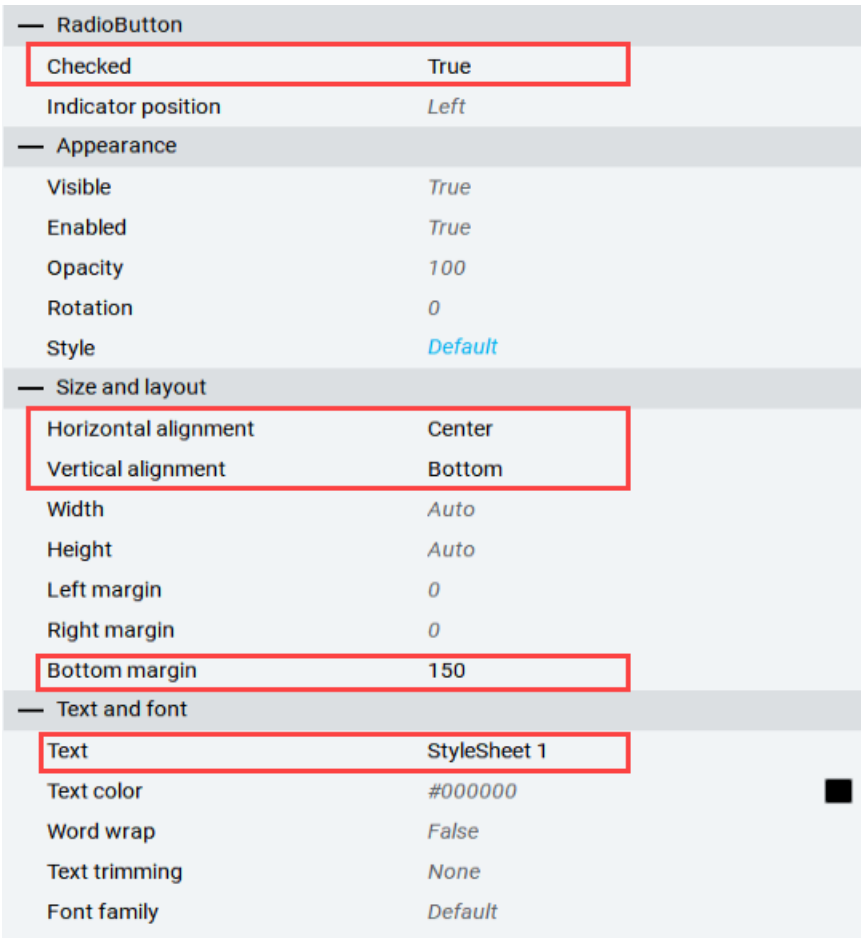
5. To automatically add the columns configured in the logger, double-click **DataGridSection1** so it opens in the editor, drag and drop the **DataLogger1** Logger into **DataGridSection1**.



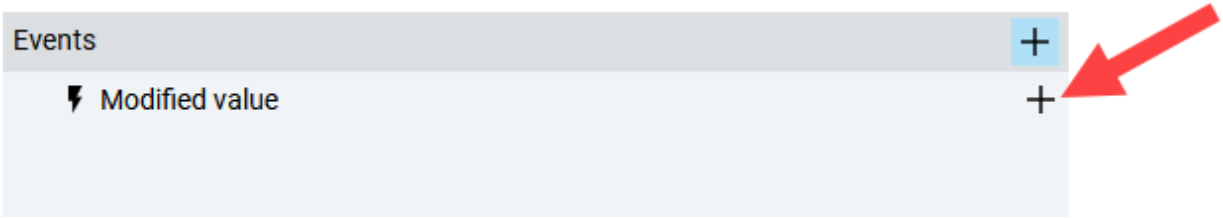
The report structure is now done.

© 2025 Rockwell Automation, Inc. All rights reserved.

6. Double click on the screen we created earlier called **Dashboard** page under the pages folder.
7. Right click on the **Dashboard** page in Project view and select **New** > **Base Control** > **Option Button**
8. Select **OptionButton1** at the bottom of the screen, change properties as below:



9. In the Events section at the bottom, Click + to add a new method.

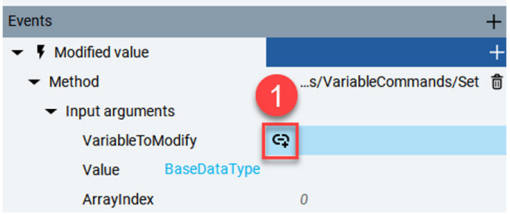


© 2025 Rockwell Automation, Inc. All rights reserved.

10. Browse to **Commands** > **Variable commands** > **Set Variable** value and click **Select**.

11. To change the stylesheet pointer to a new stylesheet

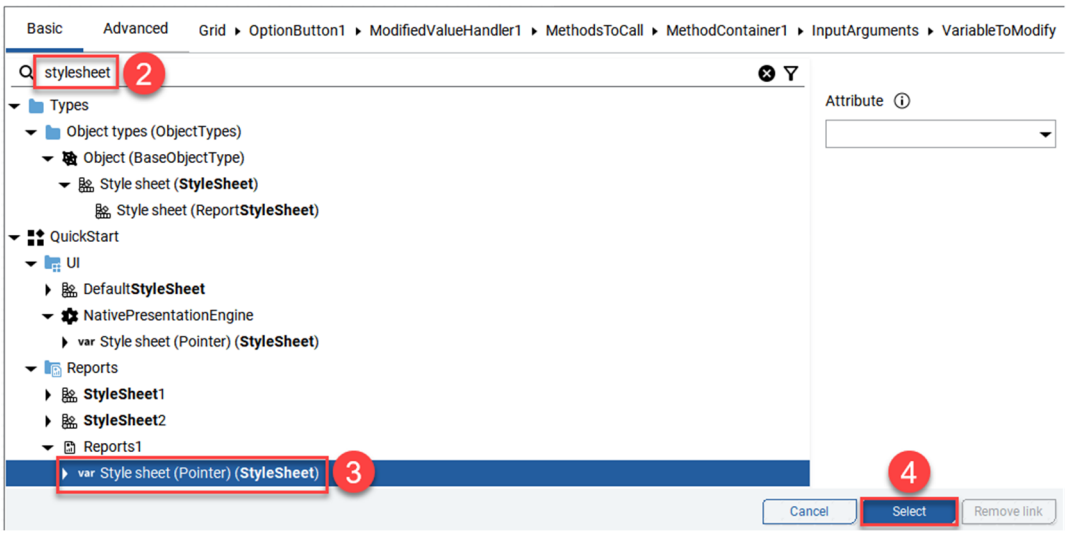
i. Mouse-over **VariableToModify** and click the Add Dynamic Link icon .



ii. In the **search** bar, type in “StyleSheet”

iii. scroll down to the bottom, select the **var StyleSheet (Pointer)**

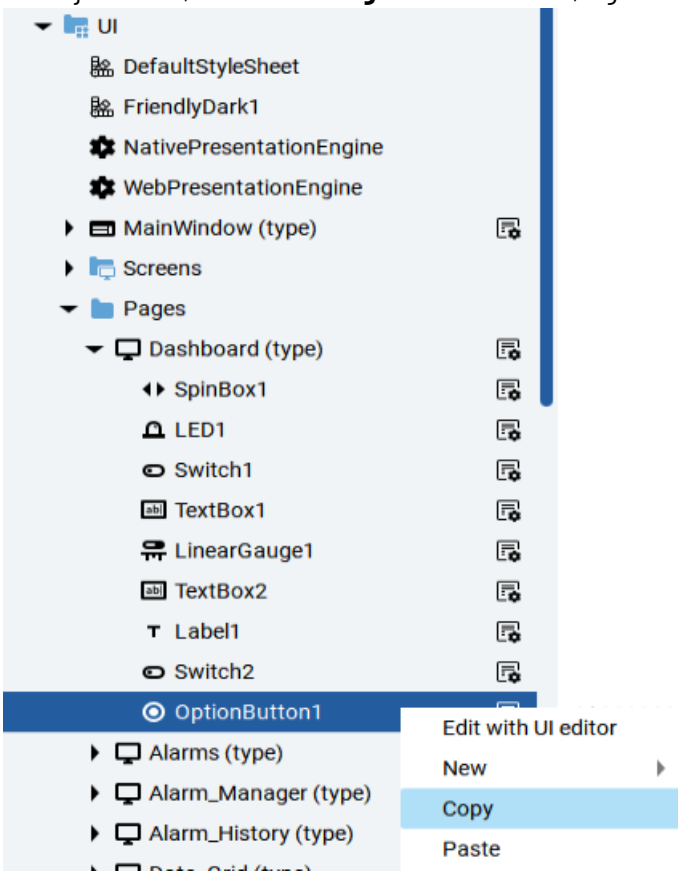
iv. and click on **Select.**



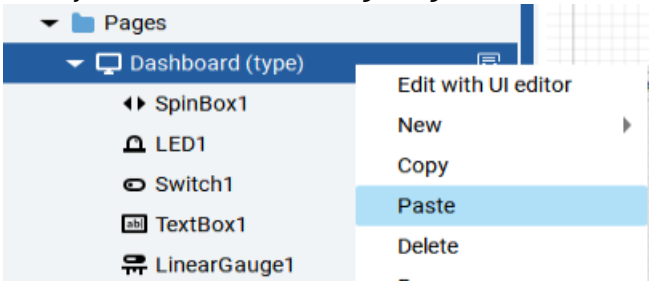
© 2025 Rockwell Automation, Inc. All rights reserved.

12. For the **Value** argument under the *Events* section, click the **Change Dynamic Link** icon to use the ISA Style Sheet (use the search field when browsing).

13. In Project view, under **UI** > **Pages** > **Dashboard**, right click **OptionButton1** and select **Copy**.

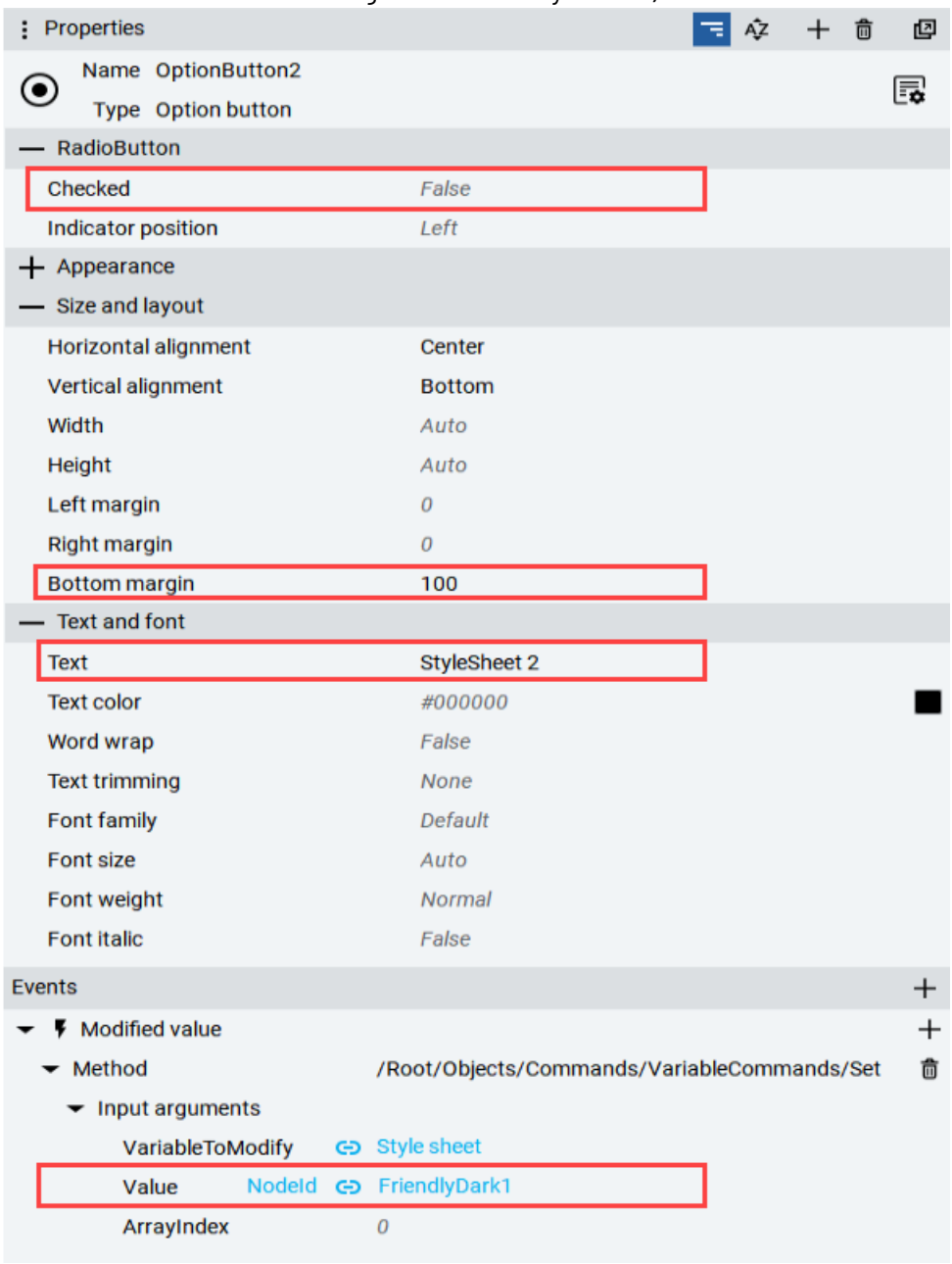


14. In Project view, under UI > Pages right-click **Dashboard** and select **Paste.**



© 2025 Rockwell Automation, Inc. All rights reserved.

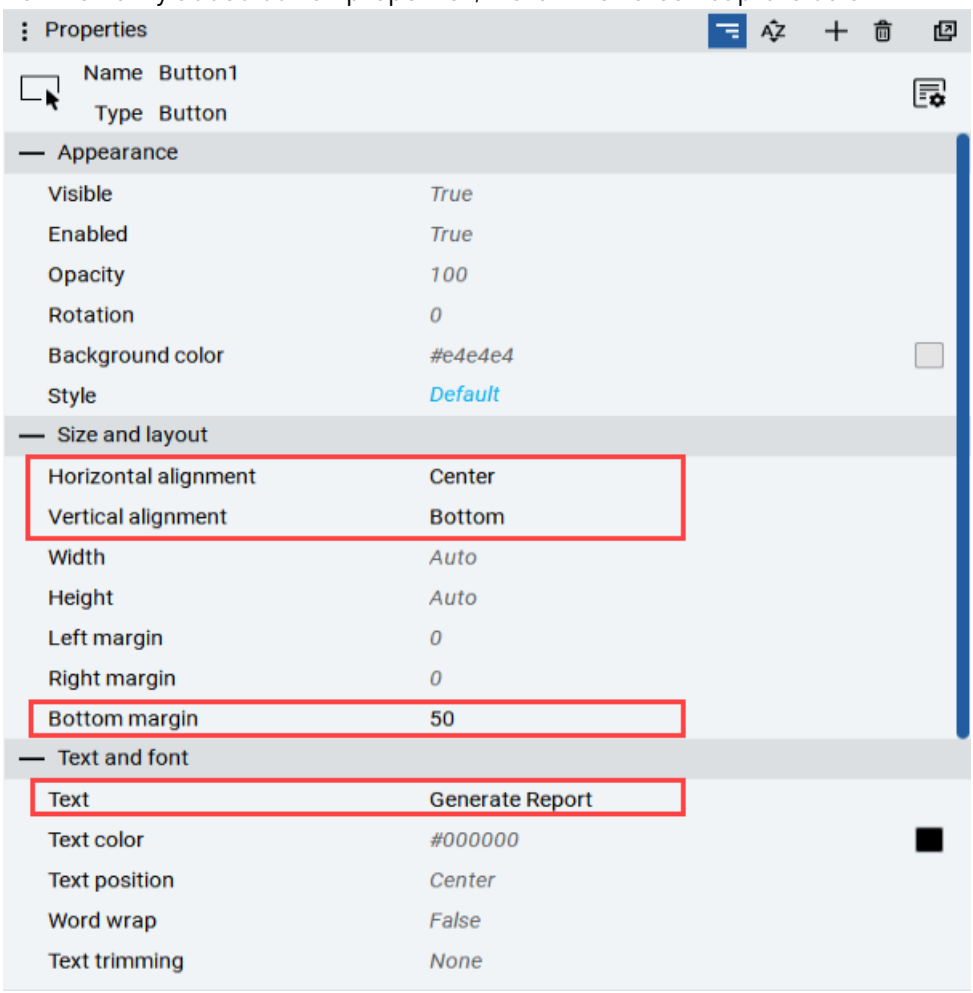
15. Change **OptionButton2** properties to match the screen capture below.
Change the **Checked** property to **False**
Change the **Bottom margin** to **100**
Change the **Text** property to **StyleSheet 2**
Change the Input argument **Value** to use the **FriendlyDark** stylesheet (as suggested before, use the search field when browsing for the new Stylesheet)



16. The last step is to add a new **Button** under Dashboard and change its properties. To add a button, right-click **Dashboard** and select **New** > **Base controls** > **Button**

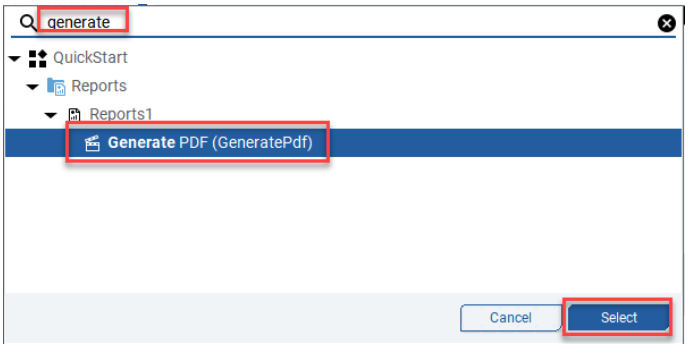
© 2025 Rockwell Automation, Inc. All rights reserved.

17. For the newly added button properties, match the screen capture below:



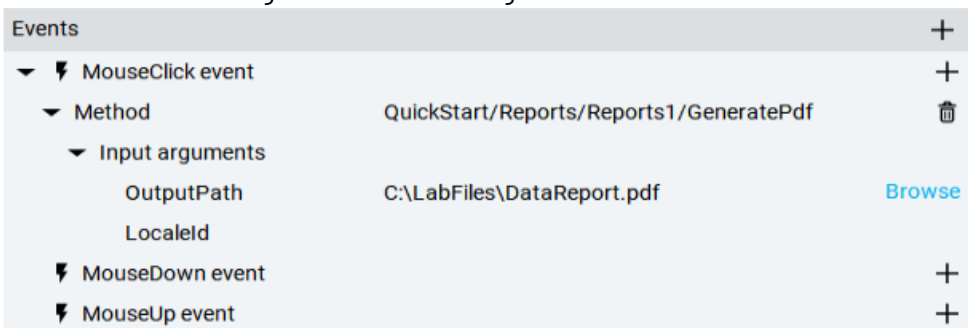
18. Still in the Properties pane, under the Events section, click + next to *MouseClick event.*

19. Type ***"generate"*** in the search bar and *select* ***Generate PDF***.



© 2025 Rockwell Automation, Inc. All rights reserved.

20. For the **OutputPath**, select the pencil and type in ***C:\LabFiles\DataReport.pdf***
Either make sure the folder exists (i.e. C:\LabFiles) or use another folder that already exists on your computer. This path will only work when emulating the project on a Windows computer or when the runtime target device is running on Windows.



Note: To create a file path with cross-platform compatibility when the runtime target device is either Windows or Linux, use forward slashes ' ***/*** ' as the directory separator in the path.

i. To generate a DataReport.pdf file in the ApplicationFiles directory of the project, use %APPLICATIONDIR%/DataReport.pdf

ii. %APPLICATIONDIR% is a placeholder used to refer to the directory where the Optix Application is located, allowing users to easily reference this location in the project without having to manually type the full path, which could be different depending.

iii. Deploying to an Emulator on your computer, the destination path of the application directory is: %localappdata%\Rockwell Automation\FactoryTalk Optix\Emulator\Projects*<Project Name>*

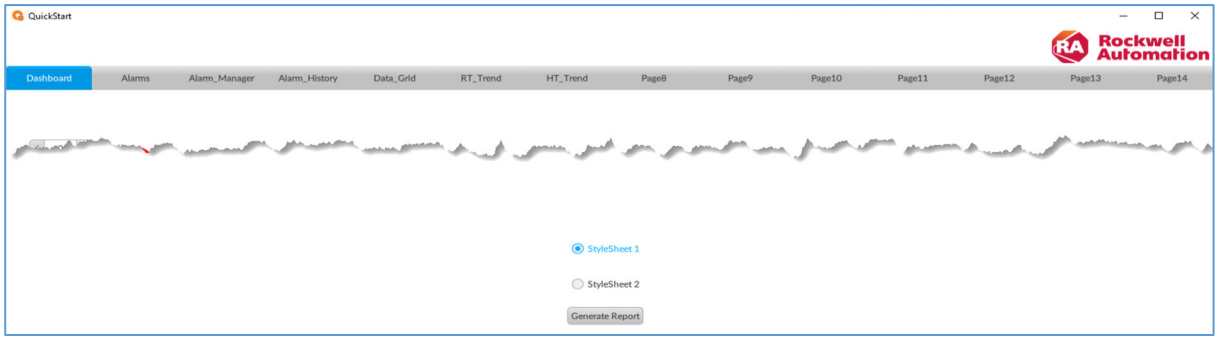
iv. Deploying to a runtime target device running Windows, the default destination path of the application directory is: %localappdata%\Rockwell Automation\FactoryTalk Optix\FTOptixApplication

v. Deploying to a runtime target device running Ubuntu Linux (i.e. OptixPanel), the destination path of the application directory is: /home/<username>/Documents/Project_1

Note: You may consider making your Optix application an FTP client or server or consider the use of the Email Sender library runtime script. To learn more about these options, search the Libraries included with FactoryTalk Optix Studio for FTP and Email.

21. **Save** your work and start the **Emulator**

22. Under the Dashboard page, select a **Stylesheet** and click on **Generate Report**.

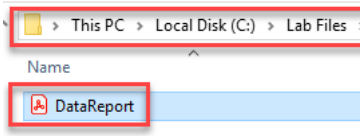


© 2025 Rockwell Automation, Inc. All rights reserved.

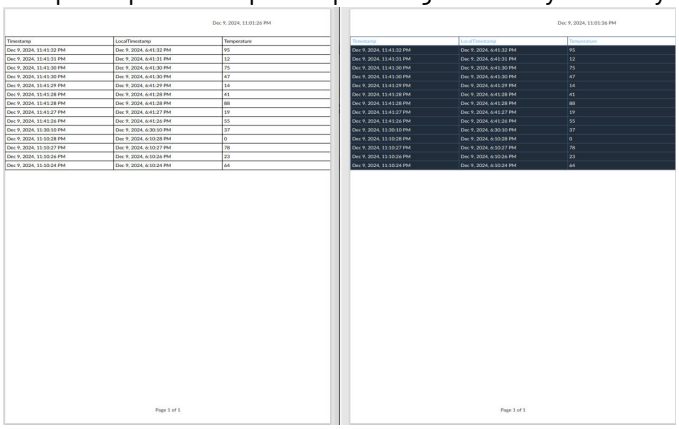
23. Open the folder **C:\Lab Files**.

    Note: As stated earlier, *C:\Lab Files* is not a valid path when using a Linux device like an OptixPanel. In this case, please use %APPLICATIONDIR%/DataReport.pdf in step 20 for the PDF file to be saved in */home/<username>/Documents/Project_1*

24. Double click on the **DataReport pdf**.



25. The pdf report will open depending on the stylesheet you selected



    Note: Make sure to close the PDF before creating a new Report.

29. Close the **Emulator**.

30. Close the pdf viewer when you have finished testing.

31. (Optional) Can you determine the total feature tokens needed to run this project in production? Which components in this project are impacting the runtime entitlement size?

    Note: **New!** Starting with FactoryTalk Optix 1.5, PDF Reports can also include the following: Pie chart, Histogram chart, XY chart, and Trend.

You have completed the Reporting lab.

© 2025 Rockwell Automation, Inc. All rights reserved.

Configure Recipes (15 Minutes)

Objectives

- Create a recipe schema using the Configure a recipe wizard
- Import and configure a Recipes Editor from the Template Libraries

Scenario

In this section of the lab, you will use the Configure a recipe wizard to create a recipe schema which will be used to define the recipe variables or ingredients. Then you will import a pre-built recipe editor from the FactoryTalk Optix library of widgets which will be used to create and interact with recipes.

RECIPE MANAGEMENT

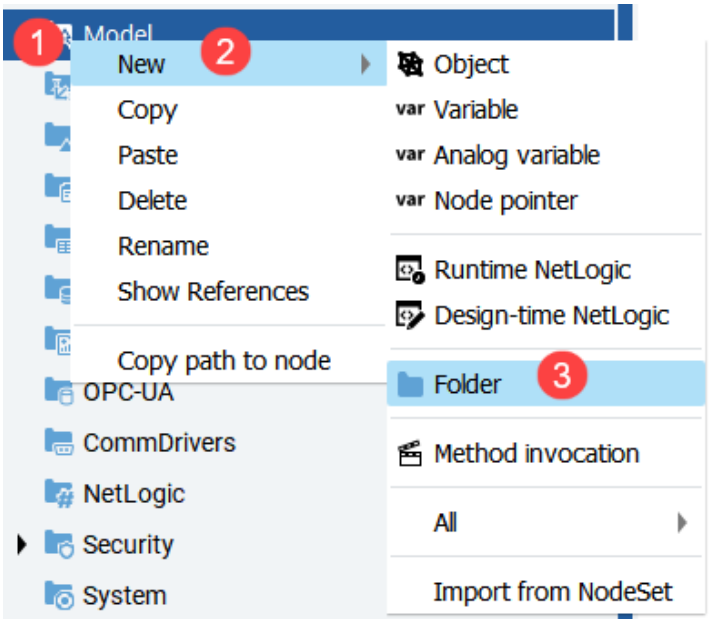
Introduction
In FactoryTalk Optix, recipe management is done through one or more Recipe schema objects.
A Recipe Schema object defines a set of variables, or ingredients, with which it is possible to configure different recipes, i.e. different sets of values for the same set of variables.
This object can be used to define the values of a configuration and save them to be reapplied as needed. It can be useful, for example, to restore the initial settings of the machine following changes that compromise its operation.
Ingredients of a Recipe Schema
To define the ingredients in a Recipe Schema, a destination node must be set at design time that contains, among all the child nodes, the variables of interest. It can be any node of the project, and the choice depends on the structure of the project and its complexity. For example, the destination node can be the Model folder, an object inside the Model folder, or a node that contains variables of one or more PLCs.
Hint
To define variables contained in different nodes of a complex project as ingredients of a Recipe Schema, create a dedicated object in the Model folder and add variables inside it, then create a dynamic link between the variables and desired ingredients. In fact, if the object is selected as destination node, the selection of the Recipe Schema ingredients is more intuitive, as only the variables referenced in the object are shown.
Recipe management at runtime
To design recipe management at runtime, the Recipes Editor widget included in FactoryTalk Optix Studio can be used.
The widget can be used as provided or some of its components can be reused to design a customized solution.
Edit model
When the user creates, edits or loads a recipe from the database at runtime, an Edit model node is automatically created in the project root node that can be referred to at design time. The Edit model node contains a temporary copy of the created/edited/loaded recipe data, until it is saved, deleted or sent to the PLC.
Important
If execution of the FactoryTalk Optix Application is stopped, the Edit model node and the data it contains are deleted.
To design custom recipe management, the mechanisms related to the Edit model node must be known. On the other hand, in the Recipes Editor widget, the references to the Edit model node are already correctly set to guarantee all features at runtime.

© 2025 Rockwell Automation, Inc. All rights reserved.

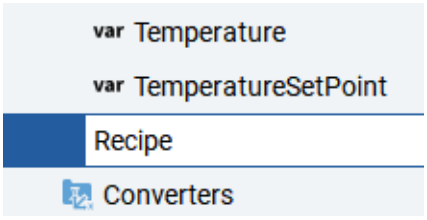
Lab Procedure

First, you will create a folder in the Model node. Then you will add some variables to this folder.

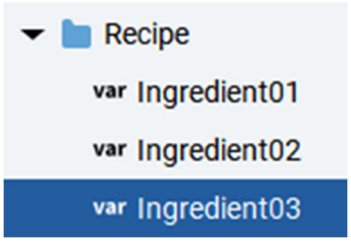
1. Right-click the **Model** folder, select **New**, and then select **Folder**.



2. Rename **Folder1** to "Recipe".

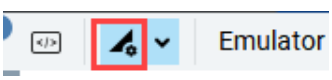


3. Drag and drop the 3 variables we created earlier called **Ingredient01**, **Ingredient02**, and **Ingredient03** under the **Recipe**. (Create the 3 variables if that step was skipped in section 1)



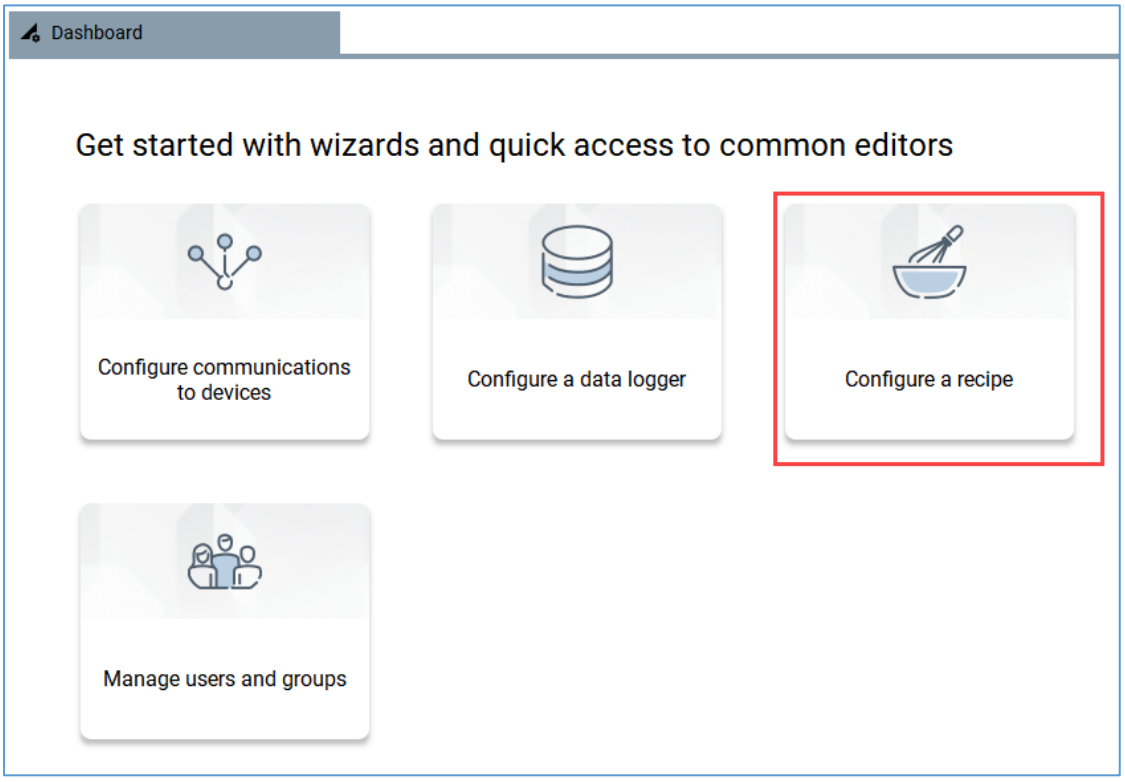
Now, you will use the **Configure a recipe** wizard to create the recipe schema.

4. Click the **Open dashboard page** icon.

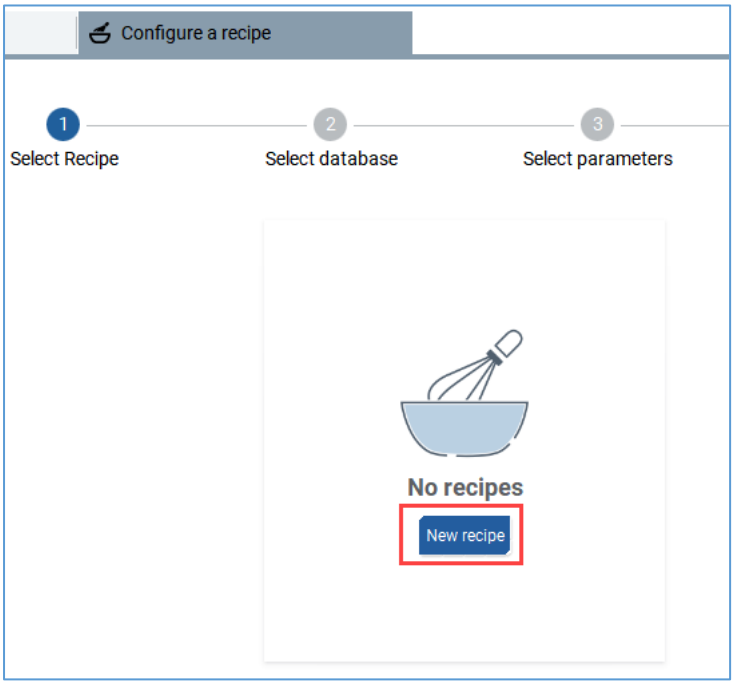


© 2025 Rockwell Automation, Inc. All rights reserved.

5. Select the **Configure a recipe** tile.

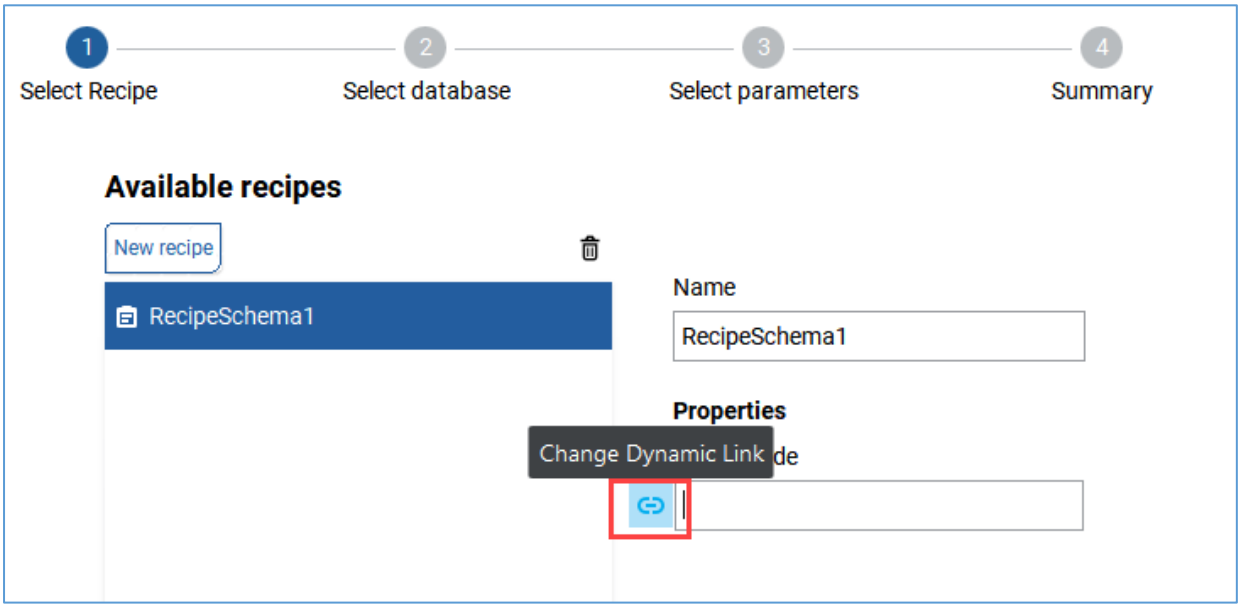


6. Click **New recipe**.

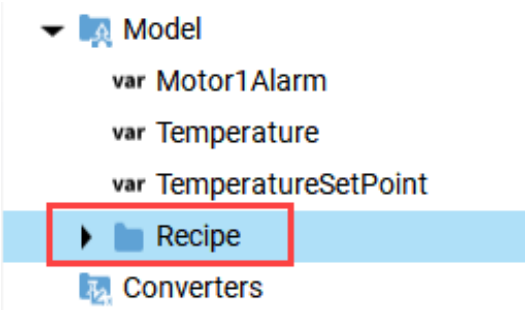


© 2025 Rockwell Automation, Inc. All rights reserved.

7. Keep the default name RecipeSchema1 and click the **Change Dynamic Link** icon for the **Target node**.



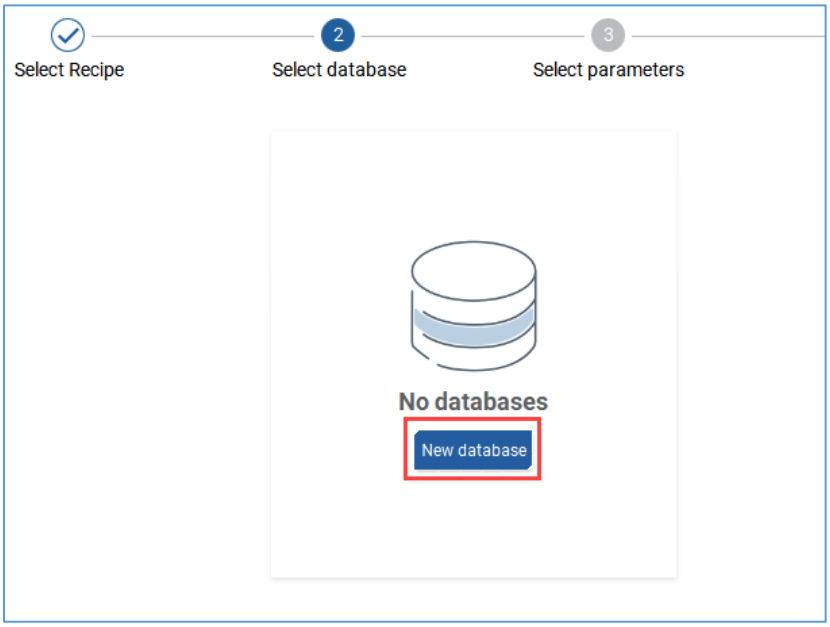
8. Click the **Recipe** folder under the Model folder and click **Select**.



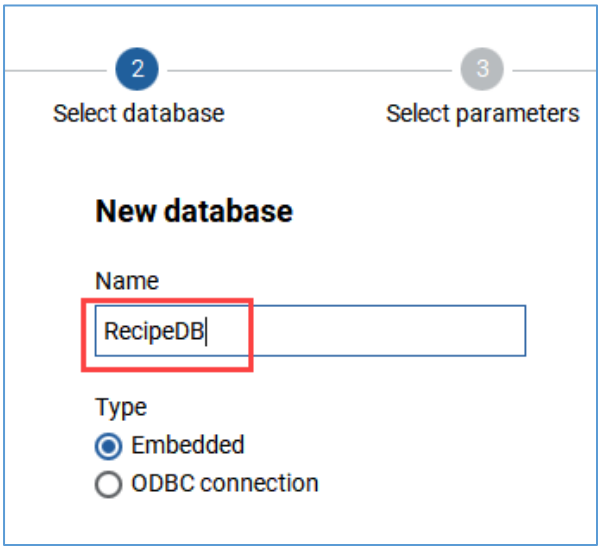
9. Click **Next**. Wait for the successful message and click **Next** again.

© 2025 Rockwell Automation, Inc. All rights reserved.

10. Click **New database** (This screenshot may look different as we already have databases created)

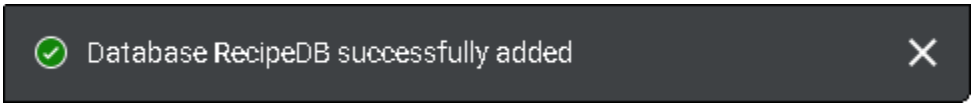


11. In the **Name** field, enter "RecipeDB" for the new embedded database and click **Next**.



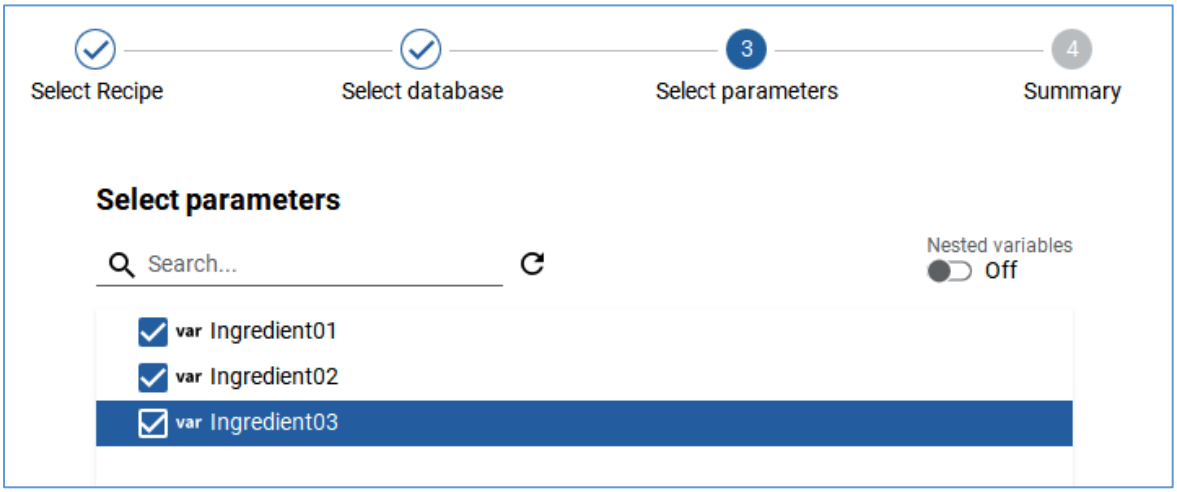
12. Keep the In Memory property set to False and click **Next**.

13. Wait for the successful message and click **Next**.

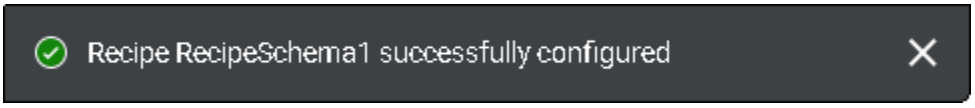


© 2025 Rockwell Automation, Inc. All rights reserved.

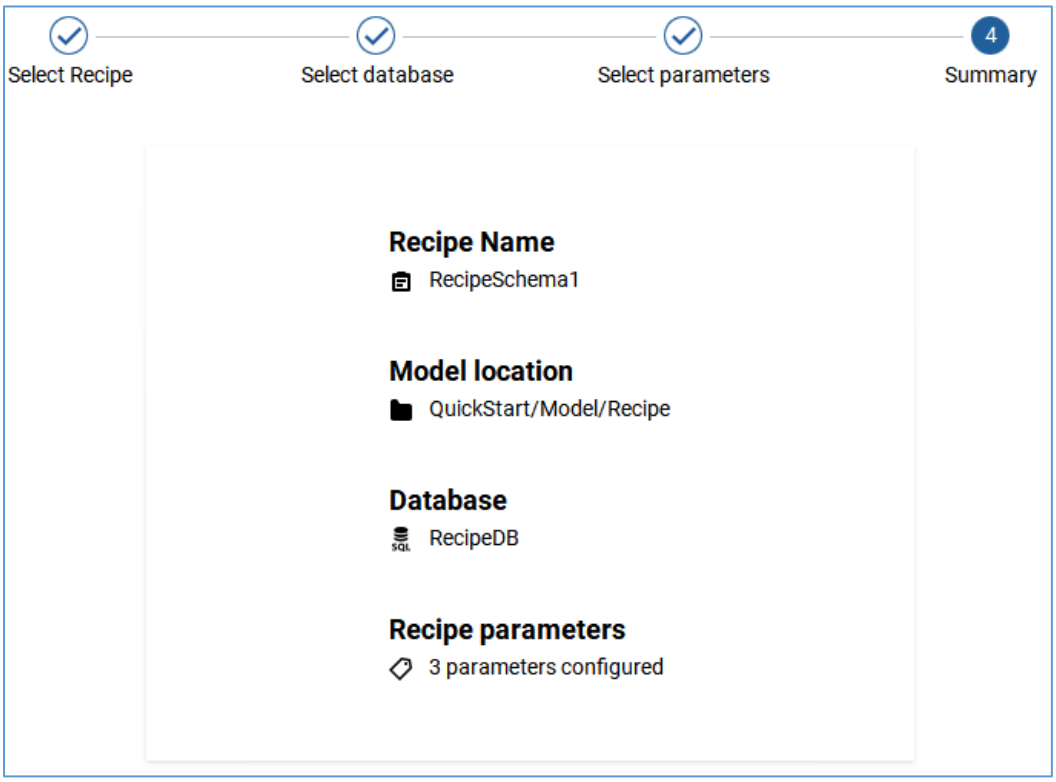
14. Select **Ingredient01**, **Ingredient02**, and **Ingredient03** and click **Next**.



15. Wait for the successful message.

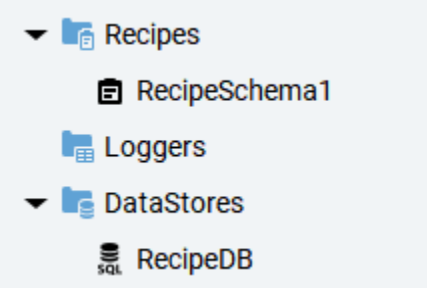


16. Review the **Summary** and then click **Exit**.



© 2025 Rockwell Automation, Inc. All rights reserved.

In Project view, **RecipeSchema1** has been created in the Recipes folder and **RecipeDB** has been created in the DataStores folder.

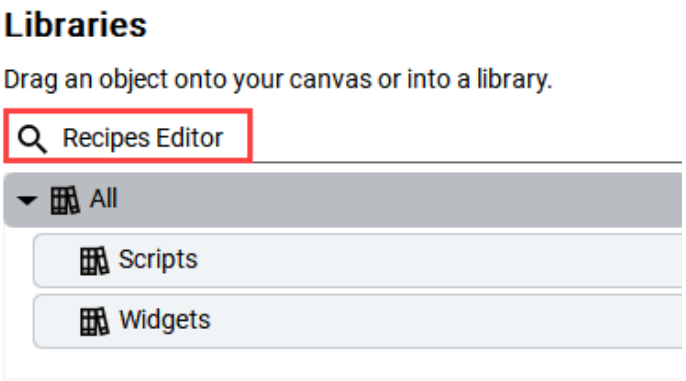


17. **Save** the project.

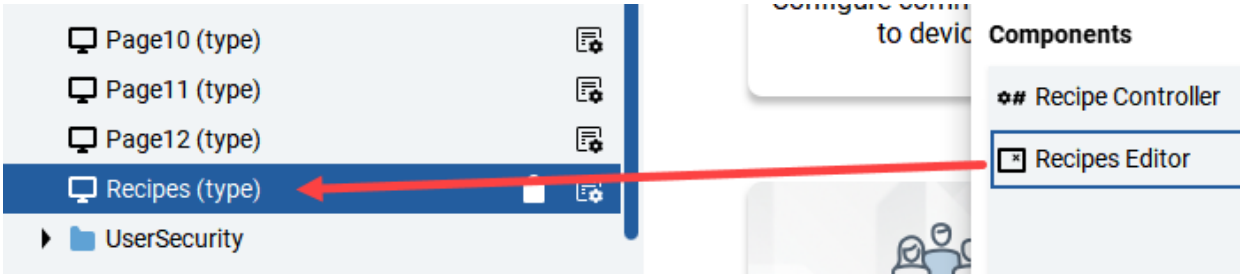
Configure a recipe editor

First, create a screen that you will add the recipe editor to.

1. Rename **Page8** to "**Recipes**".
2. From the toolbar, select the **Template Libraries** icon .
3. In **Libraries**, type "Recipes Editor" in the search field.



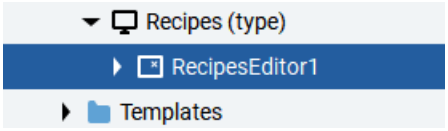
4. Under **Components**, drag and drop **Recipes Editor** onto **Recipes (type)** in Project view.



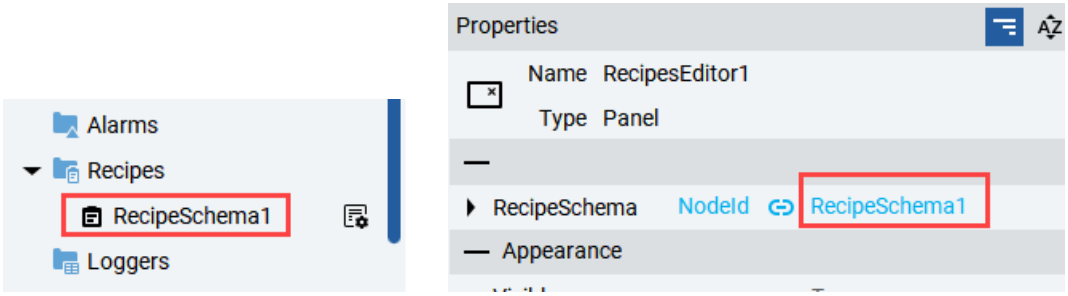
5. Select **Close** to close the **Libraries** window.

© 2025 Rockwell Automation, Inc. All rights reserved.

6. In Project view, ensure **RecipesEditor1** is selected.

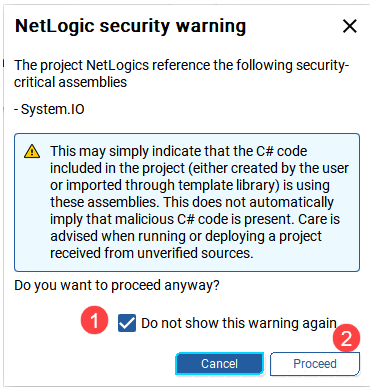


7. Drag and drop **RecipeSchema1** to the **RecipeSchema** property of **RecipesEditor1**.



8. In Project view, right-click **RecipesEditor1** and select **Execute Setup**.

9. Check the box **Do not show this warning again** and click **Proceed** at the NetLogic security warning. In this case, this simply indicates that the C# code included in this sample library object imported through the template library is using these assemblies. Care is advised when running or deploying a project received from unverified sources.

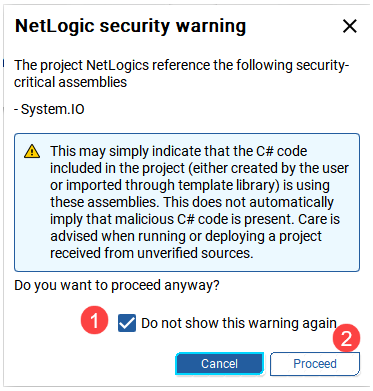


10. Save the project.

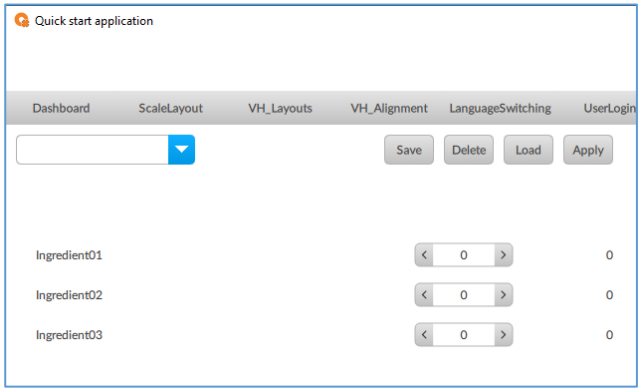
11. From the toolbar, click the **Run on Emulator** icon ▶.

© 2025 Rockwell Automation, Inc. All rights reserved.

12. Check the box **Do not show this warning again** and click **Proceed** at the NetLogic security warning if the warning appears during emulation.



13. Once the Emulator launches, navigate to **Recipes**.



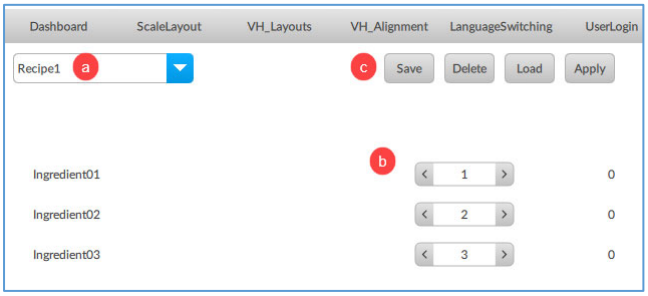
Use the recipes editor widget to create new recipes, to delete existing recipes, or to apply recipes on the controller.

14. To create a new recipe:

    a. In the recipe list, type “Recipe1” as the name for the recipe and **hit enter**.

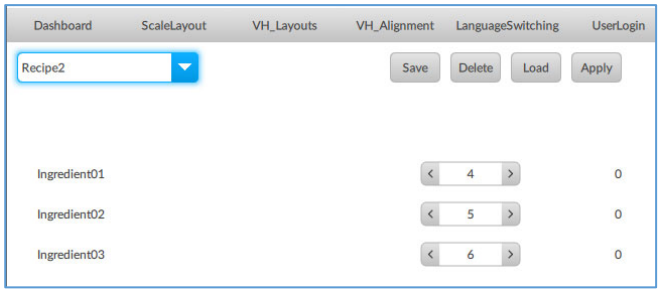
    b. Edit the values of the ingredients.

    c. Select **Save**.



© 2025 Rockwell Automation, Inc. All rights reserved.

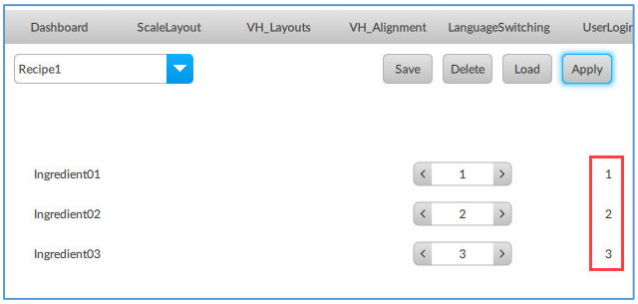
15. Repeat step 14 to create **Recipe2** with different values.



16. To apply a recipe:

    d. Select a recipe. Recipe values display in the spin boxes.

    e. Select **Apply**. Recipe values appear next to the spin boxes. These values are loaded into the assigned variables.



Note: The **Load** button loads the recipe values from the right column – the applied values – into the spin boxes. Selecting a recipe performs the Load operation.

17. (Optional) To delete a recipe, select a recipe and select **Delete**.

18. Close the **Emulator**.

19. (Optional) Can you determine the total feature tokens needed to run this project in production? Which components in this project are impacting the runtime entitlement size?

SUMMARY

FactoryTalk Optix provides recipe management capability. The recipes can be stored locally, in an embedded database, or in an external database. When developing the project, designers can use the **Template Libraries** to quickly and easily add pre-built recipe management functionality.

You have completed the Recipes lab.

© 2025 Rockwell Automation, Inc. All rights reserved.

Language Switching (20 Minutes)

Objectives

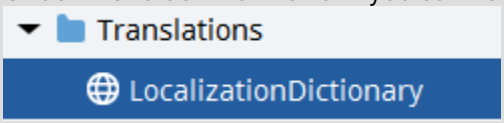
- Understand languages in FactoryTalk Optix.
- Configure localization dictionary and use translations in the project.
- Create buttons to switch languages during runtime.
- Learn how to create new dictionary entries from existing objects.
- Understand locales and learn how to change them during runtime.

Scenario

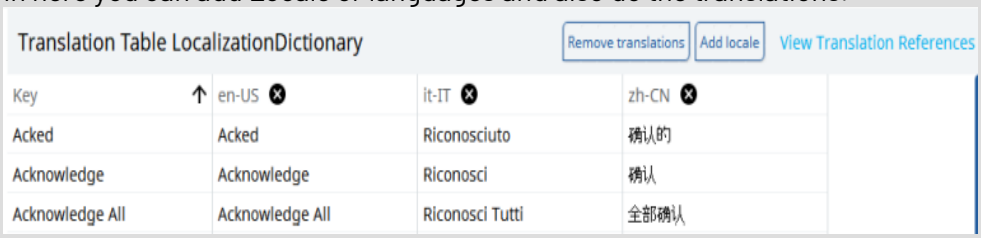
In this section of the lab, you will use the Localization Dictionary to add new translations and use them around the project. You will also be able to see how by using different locales the information on the screen will be displayed differently depending on the locale being used: language used, measuring system.

LANGUAGES

Read about Languages
Under the folder Translations you can see the Localization Dictionary.

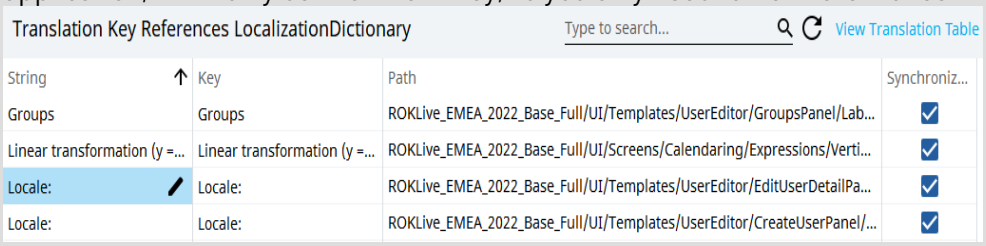


In here you can add Locale or languages and also do the translations.



Key	en-US	it-IT	zh-CN
Acked	Acked	Riconosciuto	确认的
Acknowledge	Acknowledge	Riconosci	确认
Acknowledge All	Acknowledge All	Riconosci Tutti	全部确认

Every text field in the application gets a unique name called Key which is the first column you see in the table.
When you click on **View Translation References**, you will see all the strings used in the application and each string has a unique Key. So even if for example Start is used multiple times in your application, it will only be shown as 1 key, so you only need to translate it once.



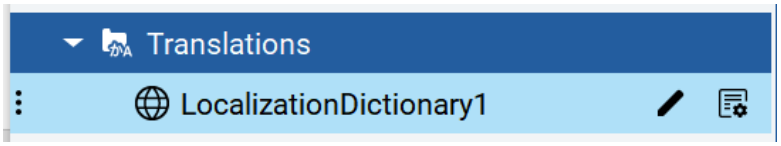
String	Key	Path	Synchroniz...
Groups	Groups	ROKLive_EMEA_2022_Base_Full/UI/Templates/UserEditor/GroupsPanel/Lab...	☑
Linear transformation (y =...	Linear transformation (y =...	ROKLive_EMEA_2022_Base_Full/UI/Screens/Calendaring/Expressions/Verti...	☑
Locale:	Locale:	ROKLive_EMEA_2022_Base_Full/UI/Templates/UserEditor/EditUserDetailPa...	☑
Locale:	Locale:	ROKLive_EMEA_2022_Base_Full/UI/Templates/UserEditor/CreateUserPanel/...	☑

© 2025 Rockwell Automation, Inc. All rights reserved.

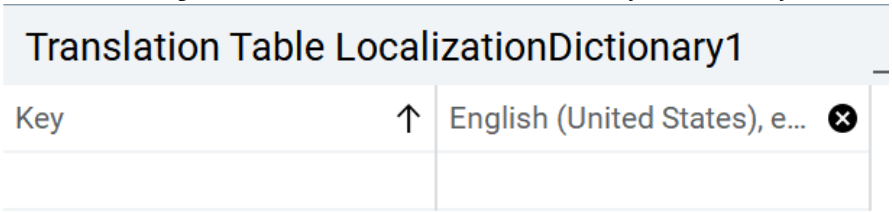
Lab Procedure

Add translations to the Localization Dictionary

1. In Project view, expand **Translations** and double click on **LocalizationDictionary1.**

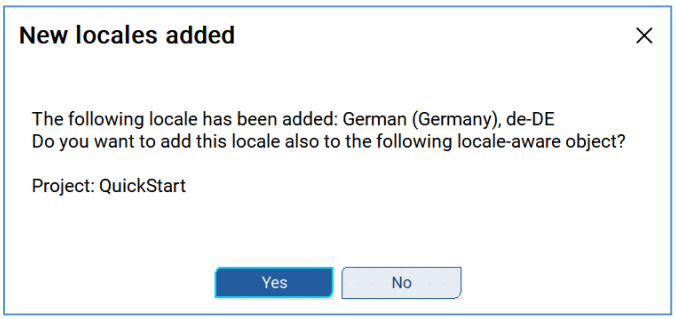


2. Notice that **English (United States)** locale is already available by default.



3. Click on **Add Locale**, choose **German (Germany) de-DE** from the dropdown list and add the selected locale:

4. You will get a confirmation message:



5. Click **Yes**
6. Now repeat the previous steps to add **Spanish (Spain) es-ES**
7. Under the **Key** column, scroll down to the first available row, and type “Hello World”
8. Fill in the columns of the first row (double click on the cell or click the pencil icon to edit)

© 2025 Rockwell Automation, Inc. All rights reserved.

9. Under the **English (United States)** en-US, type "Hello World"
10. Under **German (Germany) de-DE**, type "Halo Welt"
11. Under **Spanish (Spain) es-ES**, type "Hola Mundo"

Key	English (United States), e...	German (Germany), de-DE	Spanish (Spain), es-ES
Hello World	Hello World	Halo Welt	Hola Mundo

12. Do the same on the second available row for Keys English, German and Spanish:
Key: English
English: English
German: Englisch
Spanish: Inglés **Optional**: Press **Alt** + **130** to enter **é**
13. On the third row:
Key: German
English: German
German: Deutsch
Spanish: Alemán **Optional**: Press **Alt** + **160** to enter **á**
14. On the fourth row:
Key: Spanish
English: Spanish
German: Spanisch
Spanish: Español **Optional**: Press **Alt** + **164** to enter **ñ**

Your Localization Dictionary should look like the following:

Translation Table LocalizationDictionary1 Search... Remove translations Add locale

Key	English (United States), e...	German (Germany), de-DE	Spanish (Spain), es-ES
Hello World	Hello World	Halo Welt	Hola Mundo
English	English	Englisch	Inglés
German	German	Deutsch	Alemán
Spanish	Spanish	Spanisch	Español

Note: A design-time script is available from the FactoryTalk Optix Studio Libraries that imports or exports translations from a CSV file. You may search Libraries for translations.

Create objects using translations from the Localization Dictionary

1. In Project view navigate to **UI**-> **Pages** and rename **Page9** to "**LanguageSwitching**".
2. We will stack a smaller Panel container on top of LanguageSwicthing to ensure a consistent look regardless which resolution you selected in the start of the lab. Right click on **LanguageSwitching** and select **New** > **Containers** > **Panel**.

© 2025 Rockwell Automation, Inc. All rights reserved.

3. **Panel1** container is created under LanguageSwitching. Change the Panel1 Size and layout properties to the following.

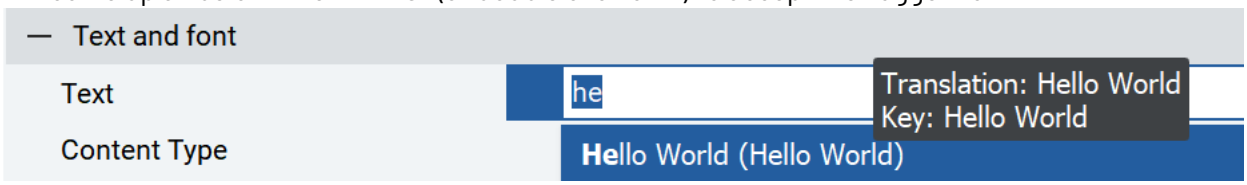
Size and layout	
Horizontal alignment	Center
Vertical alignment	Center
Width	300
Height	300
Left margin	*0*
Top margin	*0*
Right margin	*0*
Bottom margin	*0*

4. Right-click the newly created **Panel1** and select **New** > **Base controls** > **Text box**.

5. Textbox1 is created under Panel1. Set the Size and layout properties for Textbox1 to the following:

Size and layout	
Horizontal alignment	Center
Vertical alignment	*Top*
Width	*Auto*
Height	*Auto*
Left margin	*0*
Top margin	*0*
Right margin	*0*

6. For the **Text** property: type “he” and a suggestion to use the key “Hello World” from the dictionary will come up as below. Press **Enter** (or double click on it) to accept the suggestion.



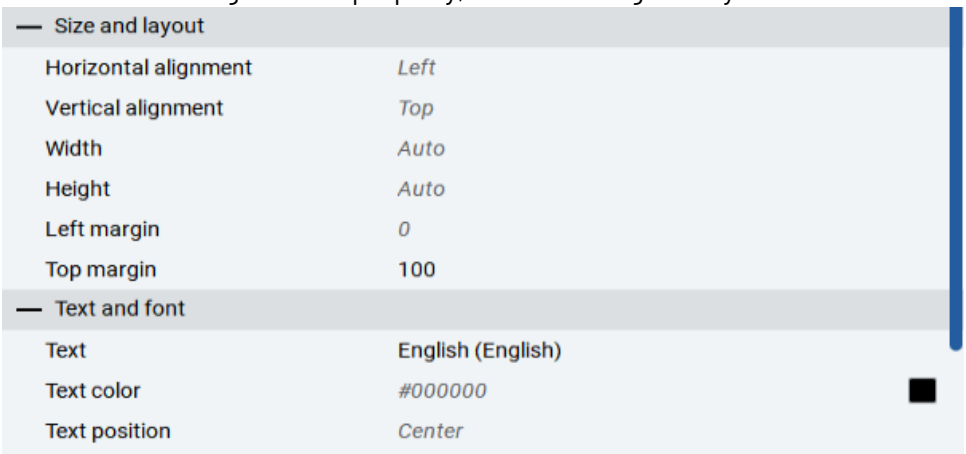
7. Set the *Text horizontal alignment* and *Text vertical alignment* to **Center aligned**.

Text and font	
Text	Hello World (Hello World)
Content Type	*Normal*
Text color	*#000000*
Text horizontal alignment	Center aligned
Text vertical alignment	Center aligned

© 2025 Rockwell Automation, Inc. All rights reserved.

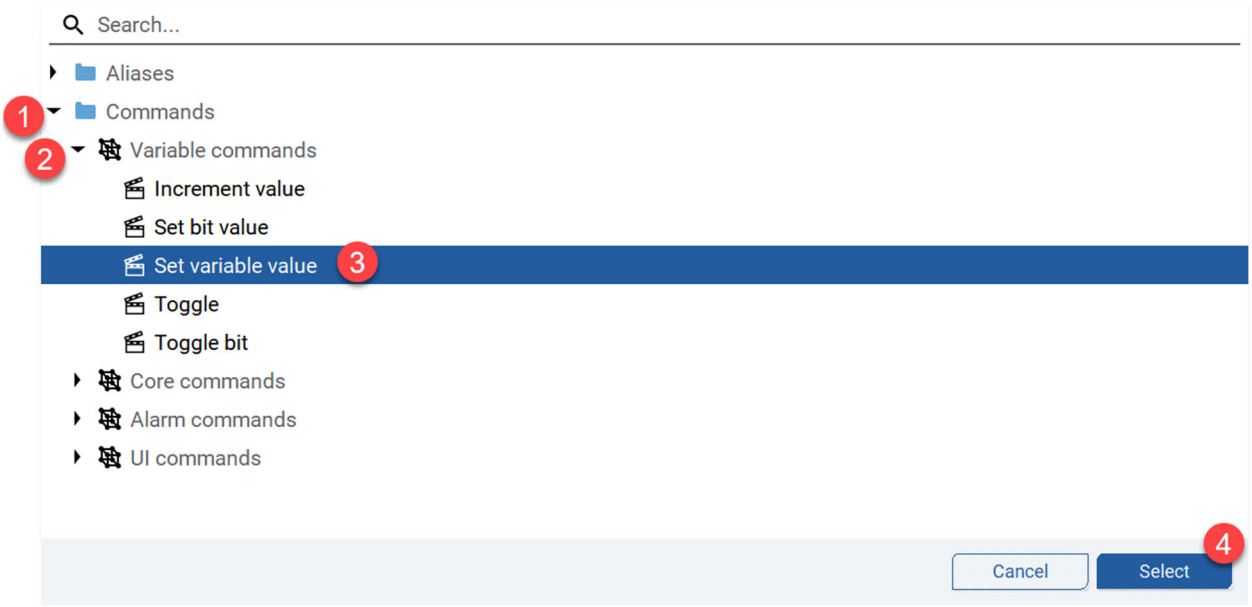
Configure buttons to change the application's language during runtime

1. In Project view, right-click on **Panel1** and select **New** > **Base control** > **Button**
2. Set the following properties for **Button1**.
   Note: When setting the Text property, chose the English key so the final result is English (English).



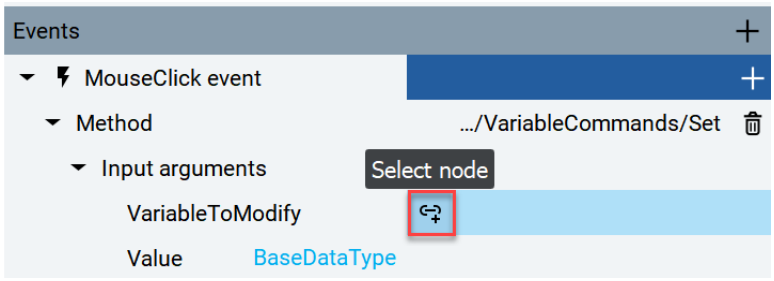
3. Still under the *Button1* properties, under the *Events* section, click + next to MouseClick Event:

4. Expand **Commands**, expand **Variable commands**, select **Set variable value**, and click **Select**

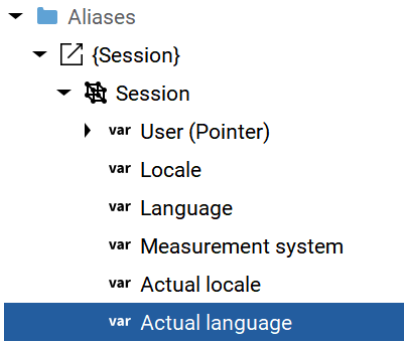


© 2025 Rockwell Automation, Inc. All rights reserved.

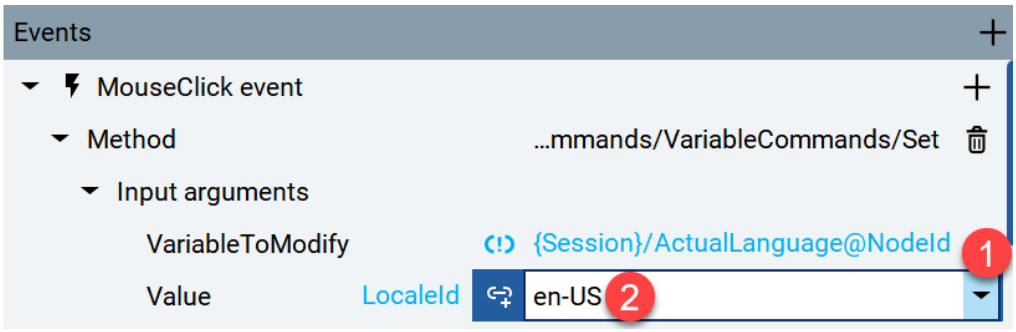
5. A method will appear under *MouseClick event*. Mouse-over the **VariableToModify** argument and click the **Add Dynamic Link** icon .



6. Navigate to language: **Aliases** > **{Session}** > **Session** > **Actual Language** and click **Select**.



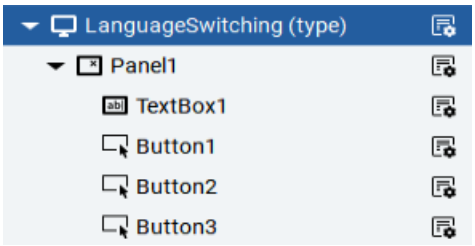
7. For the **Value** property: select the locale we will want to change the application to when pressing the button. In this case **en-US**:



We are ready to create another button to switch the language to German and another one to switch the language to Spanish

8. In Project view, right-click **Button1** we used for English, select **Copy**

9. Still in Project view, right-click **Panel1** (under LanguageSwitching) and select Paste **twice**. One button to use for German and one to use for Spanish.



© 2025 Rockwell Automation, Inc. All rights reserved.

10. For *Button2*, change the **Horizontal alignment** to **Center** and **Text** to **German (German)**:

11. Still for *Button2*, under the *Events* section, change the **Value** property to the locale we want to change the application to when pressing the button. In this case **de-DE**:

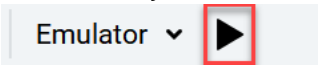
12. For *Botton3*, change the **Horizontal alignment** to **Right** and **Text** to **Spanish (Spanish)**:

13. Still for *Button3*, under the *Events* section, change the **Value** property to the locale we want to change the application to when pressing the button. In this case **de-DE**: **es-ES**

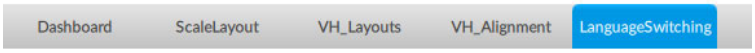
© 2025 Rockwell Automation, Inc. All rights reserved.

Panel1 under LanguageSwitching will look like the following:

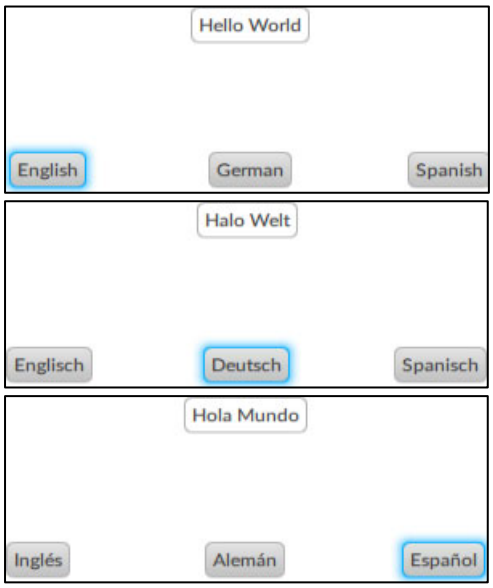
14. We are ready to emulate the project. Click Run next to the emulator



15. Navigate to **LanguageSwitching**



16. Test that the buttons we just created change the language of the text box and the text of the buttons



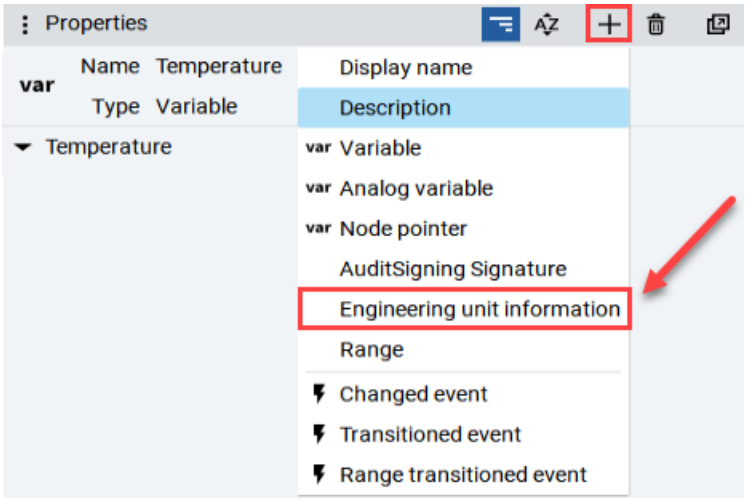
17. Close the **Emulator**.

Display Analog value according to a measurement system

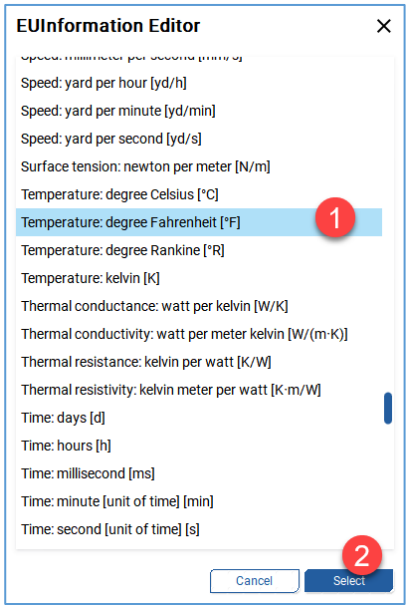
1. In Project view, under the Model folder, we had created earlier a variable called Temperature. Select the **Temperature** variable to access its properties.

© 2025 Rockwell Automation, Inc. All rights reserved.

2. In the **Properties** window for the Temperature variable, click + and select **Engineering unit information**.



3. In the *EUInformation Editor*, scroll down to *Temperature*, select **Temperature: degree Fahrenheit[°F]**, and click **Select**.

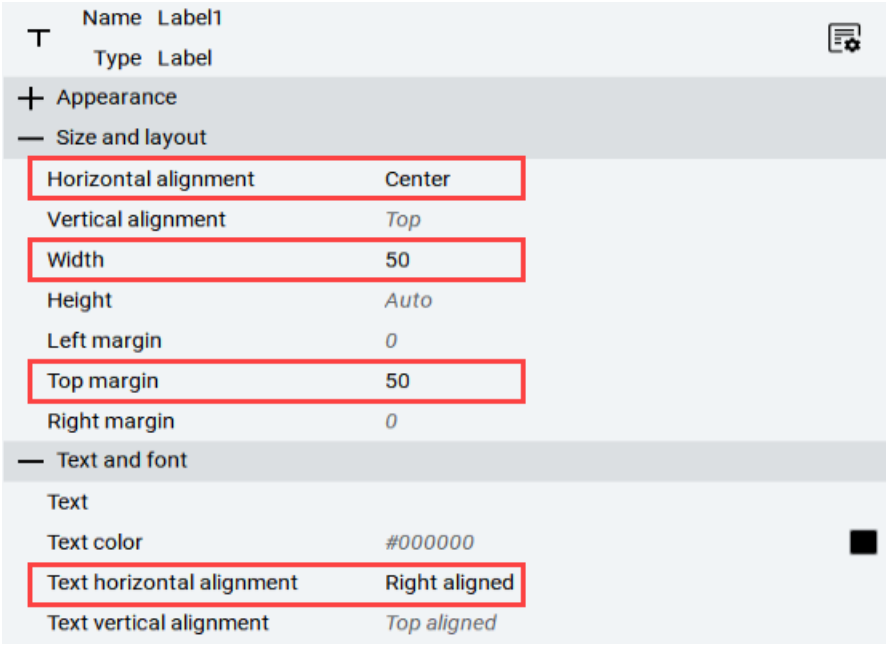


4. Note the **°F** now shown and change the default value from "0" to "***32***".

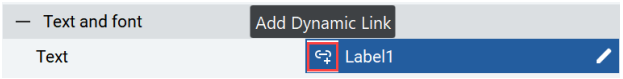
5. In Project view, under *LanguageSwitching*, right-click **Panel1** and select **New** > **Base controls** > **Label**.

© 2025 Rockwell Automation, Inc. All rights reserved.

6. *Label1* is created under *Panel1*. Under the *Size and layout* section of the *Label1* properties, set the **Horizontal alignment** to **Center**, the **Width** to **50**, and the **Top margin** to **50.** Under the *Text and font* section, set the **Text horizontal alignment** to **Right aligned.**

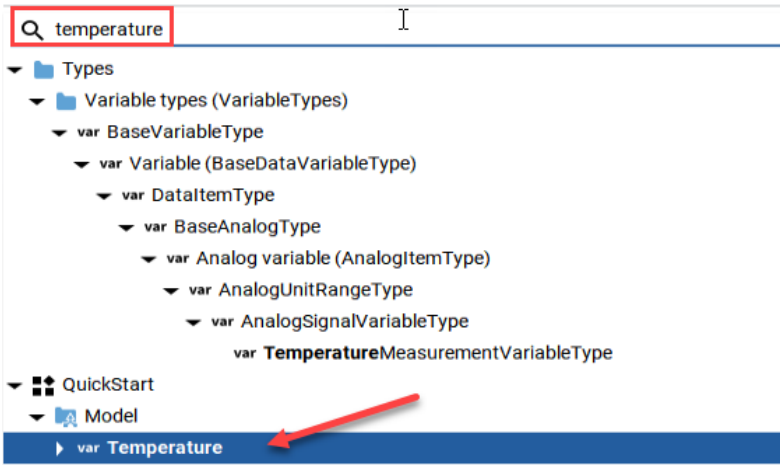


7. Mouse-over the **Text** property and click the **Add Dynamic Link** icon .



8. In the dynamic link browser, select the **Advanced** tab to create a complex dynamic link. Mouse-over the text entry field and click the **Add Dynamic Link** icon .

9. In the browser, enter "***temperature***" in the search, select **Temperature** under the Model folder and click **Select**.

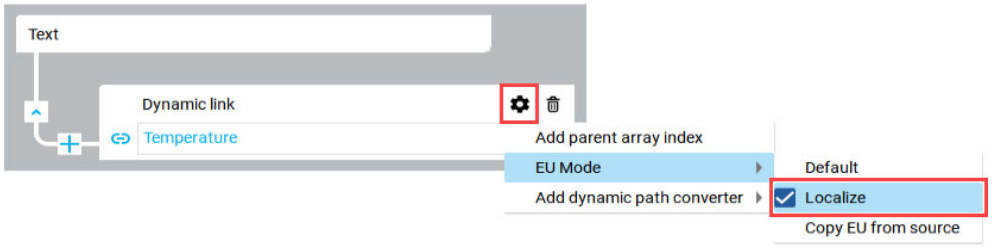


© 2025 Rockwell Automation, Inc. All rights reserved.

10. To configure the value to be converted to different engineering units

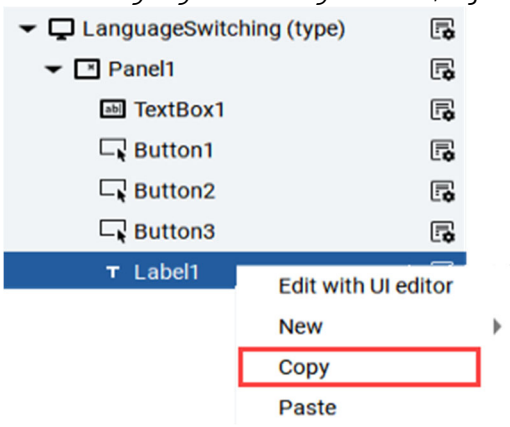
    a. Click the **Configure** icon

    b. Mouse-over **EU Mode** and then select **Localize**.

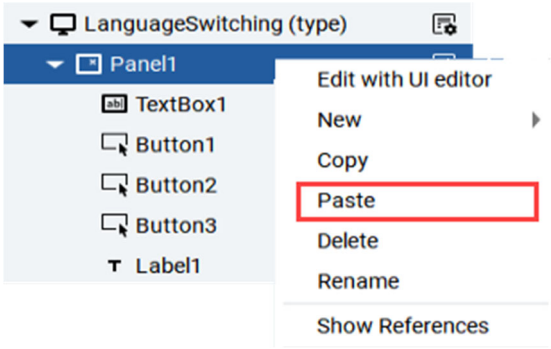


11. **Close** the dynamic link browser.

12. Next, we will configure the engineering units to be displayed next to the value. Let's copy and paste Label1 in the same panel and change its properties. From the Project view pane on the left, under *LanguageSwitching* > *Panel1*, right-click, click **Label1** and select **Copy**.



13. To add Label2 to Panel1, right-click **Panel1** and select **Paste**.



14. In the *Properties* for Label2, set **Left margin** property to **100**.

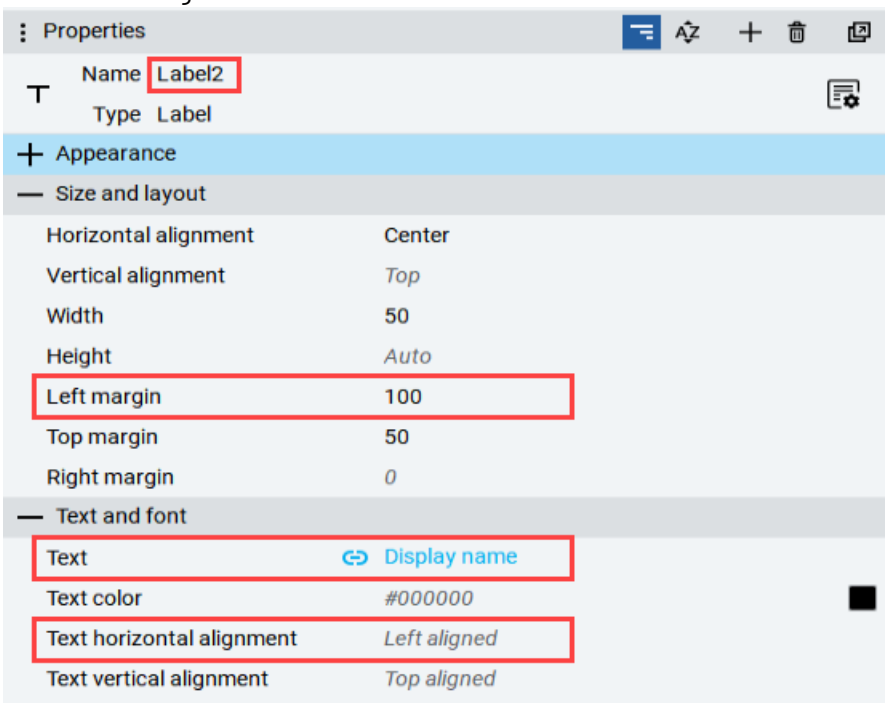
15. The Text property is already set to the Temperature variable value. Let's change it to display the engineering units name instead. For the **Text** property, click the **Change Dynamic Link** icon

© 2025 Rockwell Automation, Inc. All rights reserved.

16. In the dynamic link browser, expand **var Temperature** > **var Engineering units**, select **var Display name** and click **Select**.

17. The last property to change for Label 2 is under the *Text and font* section. Set the **Text horizontal alignment** to **Left aligned**.

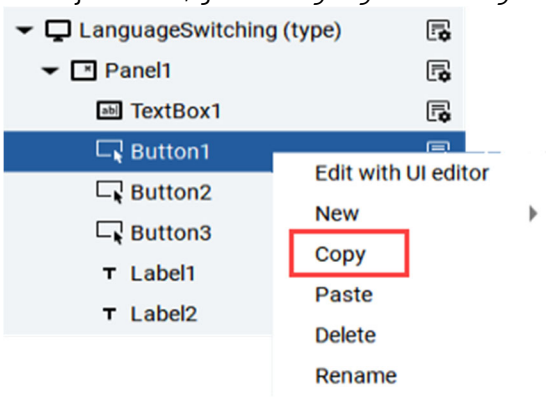
    Since Label2 is a copy of Label1, *EU Mode* is already set to *Localize*. Label2 properties will be set to the following:



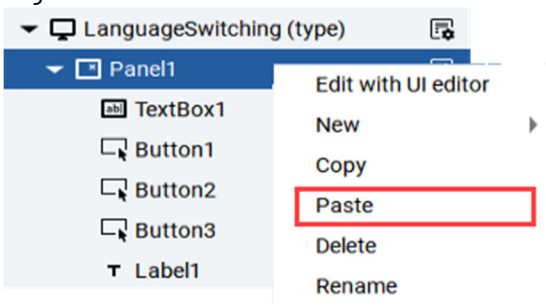
© 2025 Rockwell Automation, Inc. All rights reserved.

Create buttons to change locale during runtime

1. In Project view, go to *LanguageSwitching* > *Panel1*, right-click **Button1** and then select **Copy**.



2. Right-click **Panel1** and select **Paste** to add *Button4* under *Panel1.*

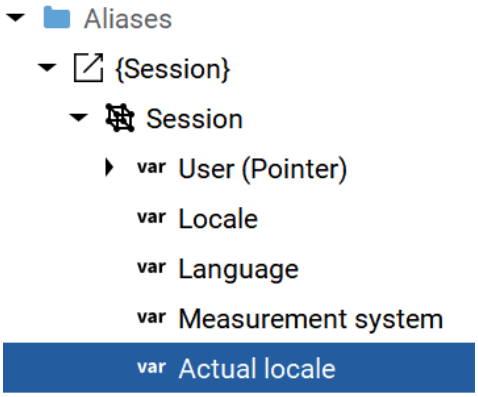


3. In the *Properties* pane on the right for *Button4*, make the following changes

    i. Set the **Top margin** to **150**

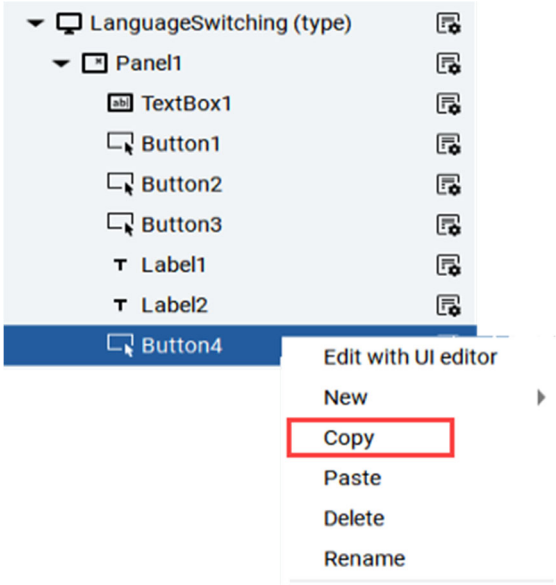
    ii. Set the **Text** property to "**en-US**"

4. Under *Events*, the method is already defined properly. We just need to change the VariableToModify argument. Use the **Change Dynamic Link** icon to browse to **Aliases** > **{Session}** > **Session** > **Actual local.** Click **Select** to close the dynamic link browser.

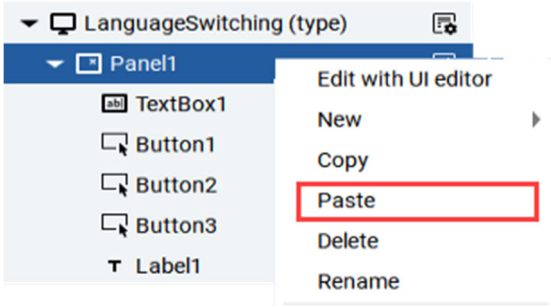


© 2025 Rockwell Automation, Inc. All rights reserved.

5. We are ready to duplicate Button4 twice to be used to change the Locale to de-DE and es-ES. Under **LanguageSwitching** > **Panel1**, right-click **Button4** and select **Copy**.



12. Right-click **Panel1** under *LanguageSwitching* and select **Paste** twice to create Button5 and Button6 under Panel1.



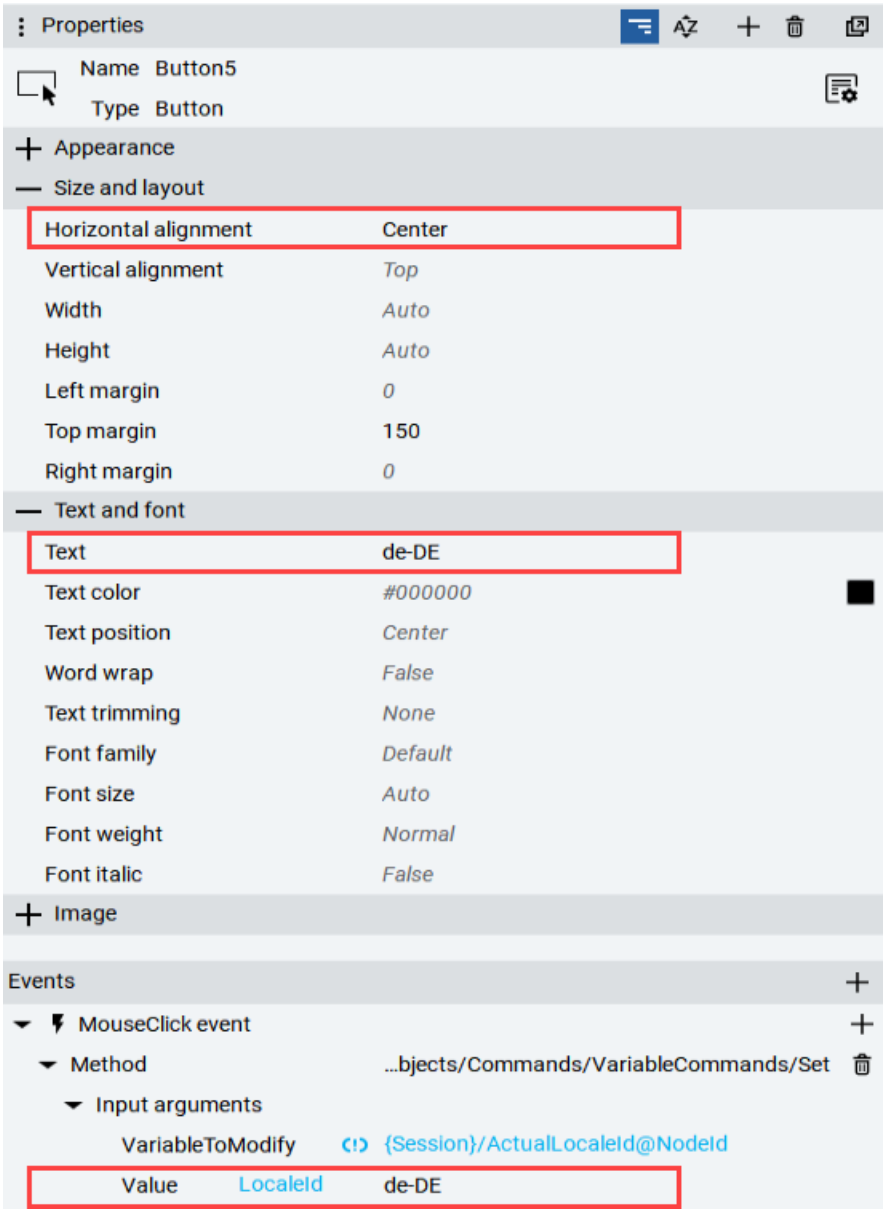
© 2025 Rockwell Automation, Inc. All rights reserved.

13. For **Button5**, change the following properties and event argument

    a. Set the **Horizontal alignment** to **Center**

    b. Set the **Text** property to **de-DE**

    c. Under *Events*, change the **Value** argument to **de-DE**



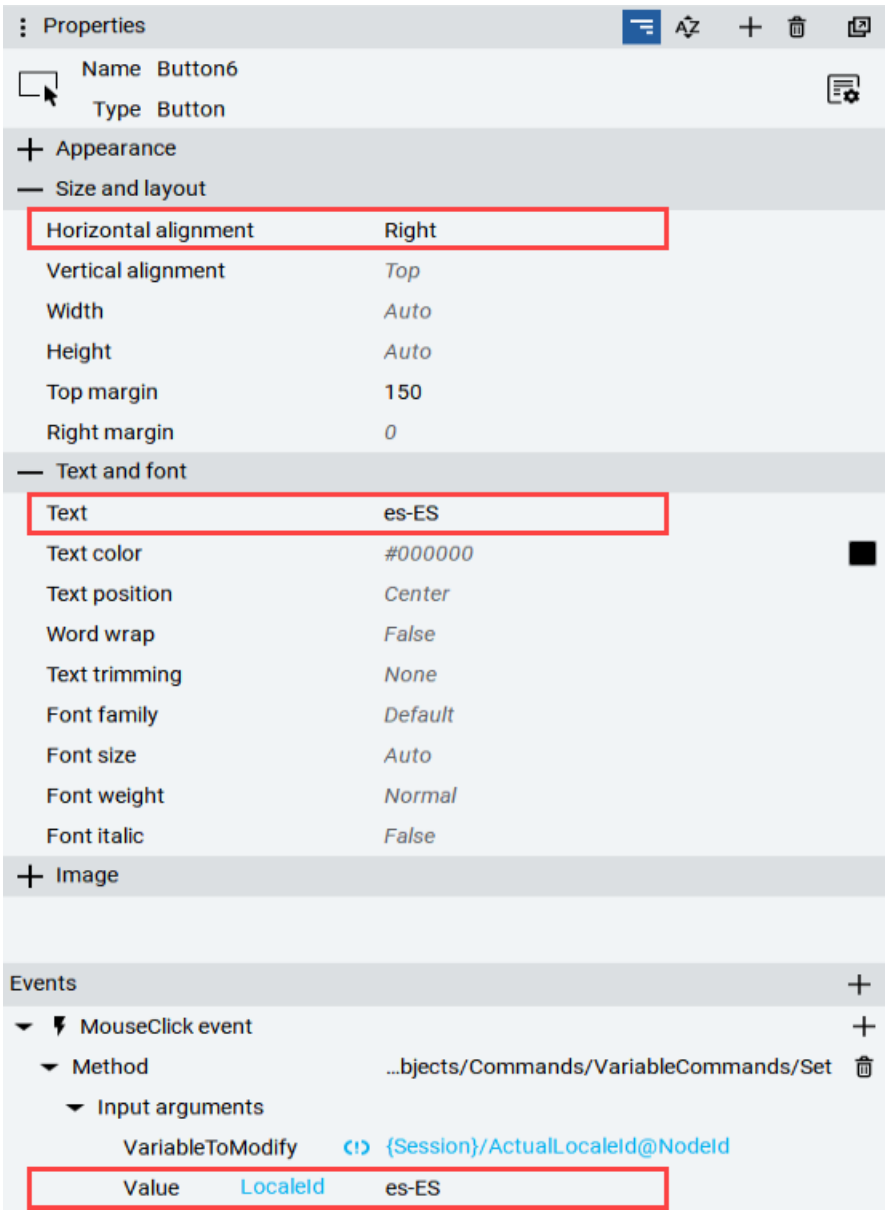
© 2025 Rockwell Automation, Inc. All rights reserved.

14. For **Button6**, change the following properties and event argument

    a. Set the **Horizontal alignment** to **Right**

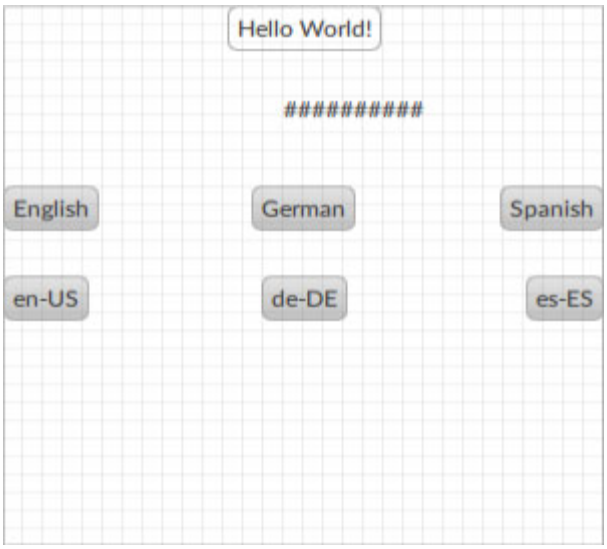
    b. Set the **Text** property to **es-ES**

    c. Under *Events*, change the **Value** argument to **es-ES**



© 2025 Rockwell Automation, Inc. All rights reserved.

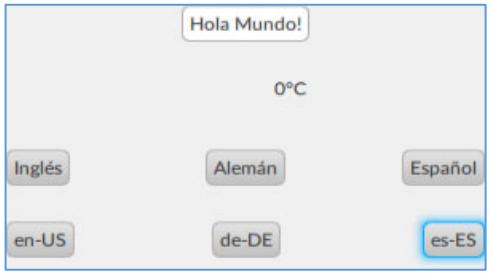
Here is how Panel1 will look like:

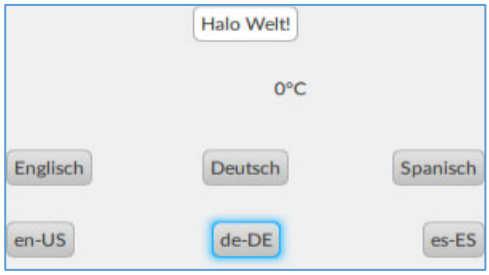


15. From the toolbar, click the **Run on Emulator** icon ▶.

24. Navigate to the **LanguageSwitching** screen.

Note: In addition to be able to change the language, FactoryTalk Optix allows you to change the Locale. The Locale specifies a language and a country, e.g. en-US, en-UK, it-IT, etc. In particular, the second segment determines the date and time format, the date separator and the measurement system (International Measurement System, United States Customary System or British Imperial System).





Note: User Login with Locales is covered in the next section.

16. Close the **Emulator**.

17. (Optional) Can you determine the total feature tokens needed to run this project in production? Which components in this project are impacting the runtime entitlement size?

© 2025 Rockwell Automation, Inc. All rights reserved.

SUMMARY

FactoryTalk Optix provides the ability to create a single application that can be used in various countries via localization. Users can select their specific language via a button and any analog values and associated engineering units will convert to their locale. Additionally, any strings that have been translated will also convert to the user's native language.
Further, rather than using a button, the locale information can be associated with a user in the security configuration for the user.

You have completed the Language Switching lab.

LOCALES

Read about Locale

Introduction

The term locale means the set of display settings of a user interface based on language and country. It is represented by a label called locale ID, made up of language and country (e.g., en-US, en-UK, it-IT etc.).
A project can support multiple locales. The locales supported by the project are set in the Locales property of the project node.
In multilingual projects, the Presentation engine displays the interface based on the session locale, if configured, and the texts displayed based on the translations available in the LocalizationDictionary object.

Locale IDs

The Locale ID specifies a language and a country, e.g. en-US, en-UK, it-IT. In particular, the second segment determines the date and time format, the date separator and the measurement system (International Measurement System, United States Customary System or British Imperial System).

Session locale

The session locale determines the locale of the user interface (i.e. the translation of the texts according to the project settings), the data display format, and the conversion of all values according to the required measurement system. It is set at runtime based on the user or object locale Session UI.

Note

If there are no locale settings at the user or Session UI object level, the session locale is set based on the locales supported by the project: in particular, the first locale in order of writing in the Locales property has priority (see Fallback locale).

Fallback locale

If the user-determined session locale is not configured in a project, or if translations of some interface texts are not available for the session locale, the texts are displayed in the project fallback locales, configured in the Fallback locales for translations property of the project node.

Note

When there are multiple fallback locales, at runtime the system uses the fallback locale based on the order of insertion of the different locales in the Fallback locales for translations property. If the list of fallback locales contains the locale not supported by the project, this fallback locale takes priority over the others.

© 2025 Rockwell Automation, Inc. All rights reserved.

Security (20 Minutes)

Objectives

- Examine the Project Settings regarding security
- Create users and groups during design time
- Create screens to be used during runtime to manage users and log into the application
- Add elements to a screen to explore some security and localization settings during runtime

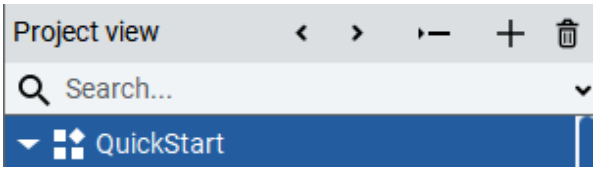
Scenario

FactoryTalk Optix provides the ability to perform project, local, and domain authentications. Users can be created both during design time and runtime – with runtime creation limited to project scoped users. FactoryTalk Optix also provides the ability to assign a locale to a user. Therefore, when a specific user logs into the application, the values and associated engineering units will convert to the locale assigned to that user. Additionally, any strings that have been translated will also convert to the user's native language.

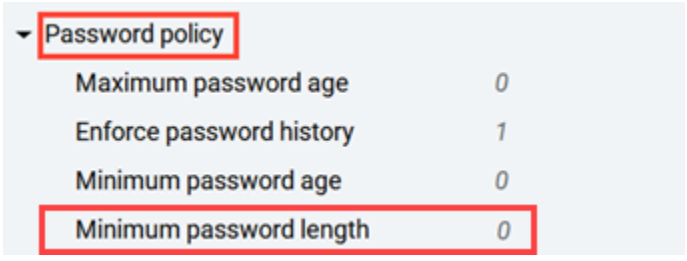
Lab Procedure

First, you will examine the **Localization**, **Authentication**, and **Password policy** properties that are available for configuration within the project.

1. In Project view, click **QuickStart**.

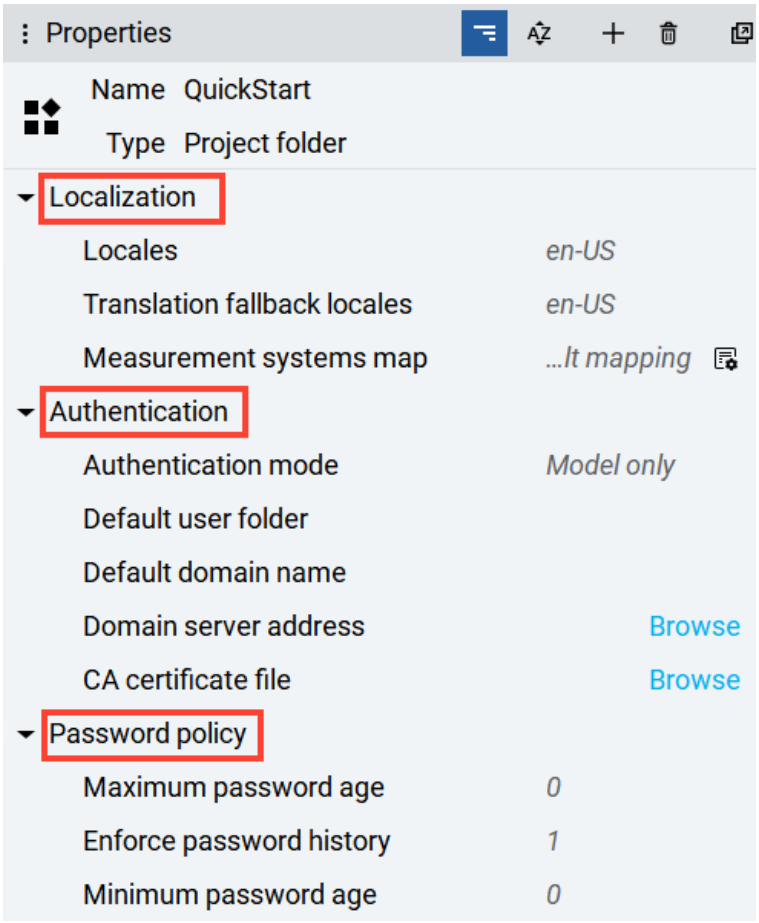


2. Looking at the project Properties on the right, under Password policy, change the **Minimum password length** to **0** (please note that this is not recommended in production, but will help us go faster with this lab)

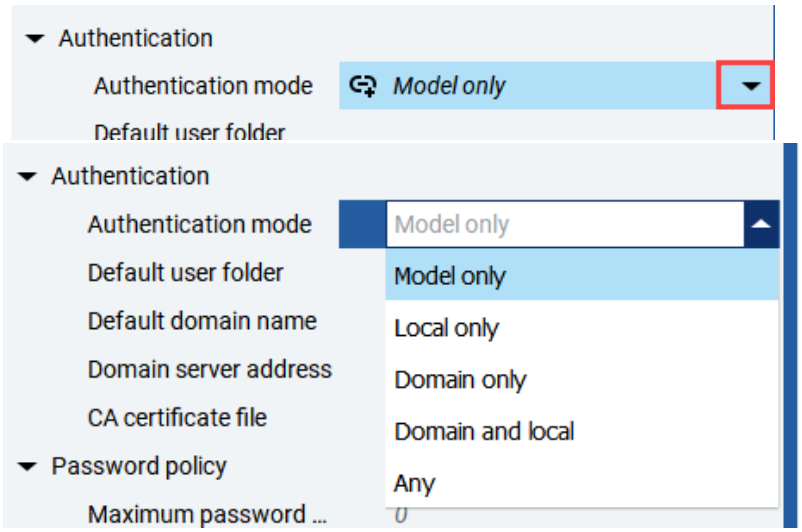


© 2025 Rockwell Automation, Inc. All rights reserved.

Notice the Localization, Authentication, and Password policy properties (Screenshots may differ depending on what optional sections have been completed previously).



3. Click the **Authentication mode** drop-down arrow and observe the available options.



4. Leave the default option of **Model only**

© 2025 Rockwell Automation, Inc. All rights reserved.

AUTHENTICATION MODES

Model uses the accounts created within the project.

Local uses accounts created on the local PC.

Domain uses accounts created in the specified domain.

Domain and local uses accounts created in both the local PC and the specified domain.

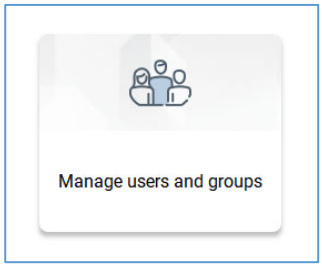
OAuth 2.0 Users authorized with the OAuth 2.0 protocol with PKCE.

Any uses all of the above.

Create Users and Groups

1. From the toolbar, Click the **Open dashboard page** icon

2. Select the **Manage users and groups** tile.



In Optix, you can define, Users, Groups and Roles.

a. **Users** : Each project has a default Anonymous user that authenticates the application session. Additional users can be created when needed. At runtime, the session user is the user who is logged on to the application. The content and UI that appears depends on the visibility set for a group of users or users with a specific role.

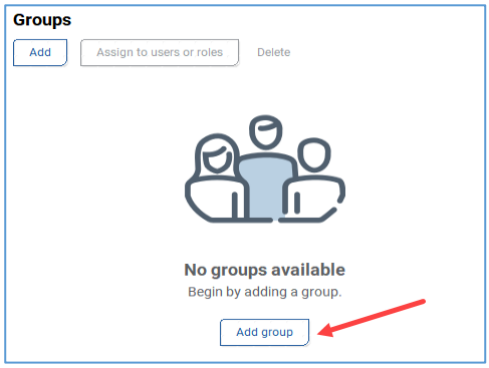
b. **Groups** : Define groups to classify users in different areas of your system. For example, PackagingTeam1, PackagingTeam2, and PackagingTeam3. Groups create system organization for users. You can show or hide screens and graphic objects for specific groups at runtime

c. **Roles** : **New!** Starting with FactoryTalk Optix version 1.3, define a set of roles for users and groups that have different responsibilities. For example, create an Administrator role, an Operator role, and a Guest role, and then assign users and groups to the applicable role. You can show or hide screens and graphics objects for specific roles at runtime. Roles identify the responsibilities of users and groups. Roles typically create security profiles.

© 2025 Rockwell Automation, Inc. All rights reserved.

In this lab, we will create 2 users and 2 groups.

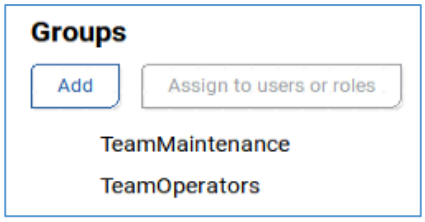
3. Let's start with groups. Click **Add group**.



4. For the name, enter ***TeamOperators*** and click **Add**.

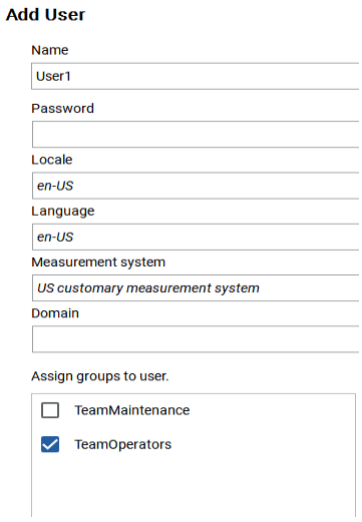
5. In the Groups section, click **Add** to add another group. For the name, enter ***TeamMaintenance*** and click **Add**.

We have just created 2 groups:



6. In the Users section, click **Add** to add a new user.

7. For User1, do the following



   a. Keep the Name as User1

   b. Leave the Password blank

   c. Set the **Locale** to **en-US**

   d. Set the **Language** to **en-US**

   e. Set the **Measurement system** to **US customary measurement system**

   f. Leave the **Domain** field blank.

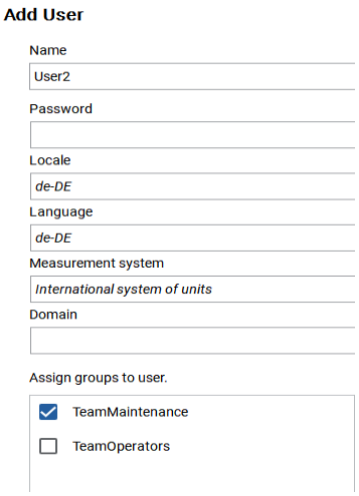
   g. Check **TeamOperators** to assign this group to User1

   h. and click **Add**.

8. In the Users section, click **Add** to add one more user.

© 2025 Rockwell Automation, Inc. All rights reserved.

9. For User2, do the following



   a. Keep the Name as User2

   b. Leave the Password blank

   c. Set the **Locale** to **de-DE**
   (Leave blank or set to en-US if only one local exists in the project)

   d. Set the **Language** to **de-DE**
   (Leave blank or set to en-US if language translation is not configured in the project)

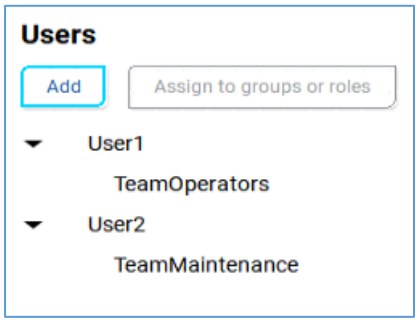
   e. Set the **Measurement system** to **International system of units**

   f. Leave the **Domain** field blank.

   g. Check **TeamMaintenance** to assign this group to User2

   h. and click **Add**.

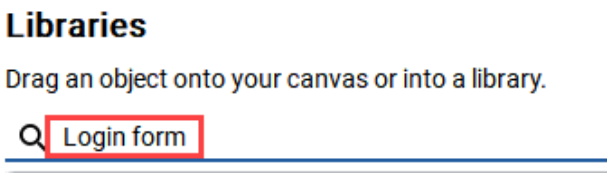
We have just created 2 users where each is assigned to a different group



Add the LoginForm to the project

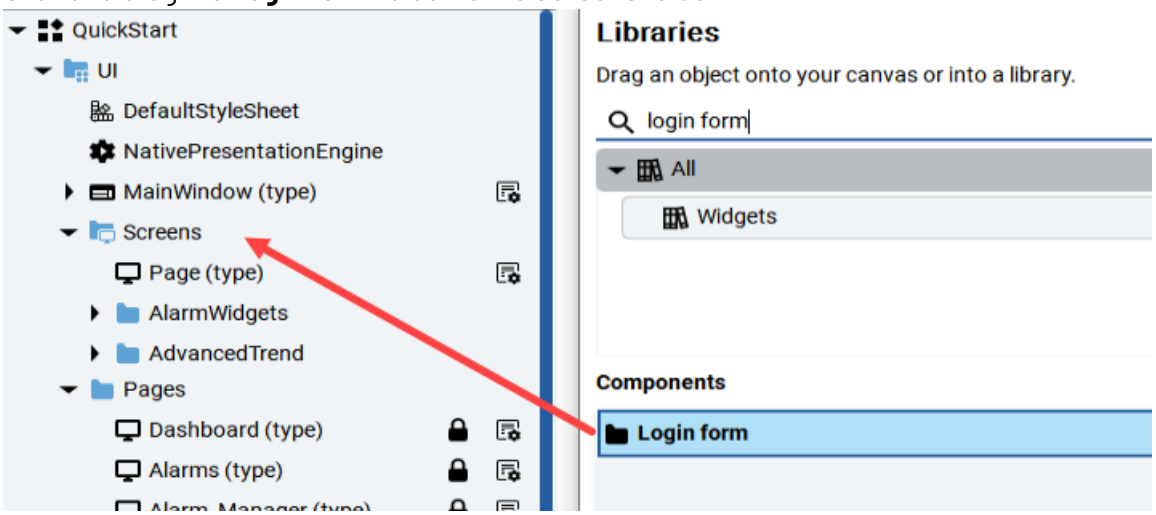
1. Click the **Template Libraries** icon.

2. Type “Login form” in the **Libraries** search field.



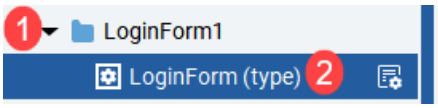
© 2025 Rockwell Automation, Inc. All rights reserved.

3. Click and drag the **Login form** folder to the **Screens** folder.

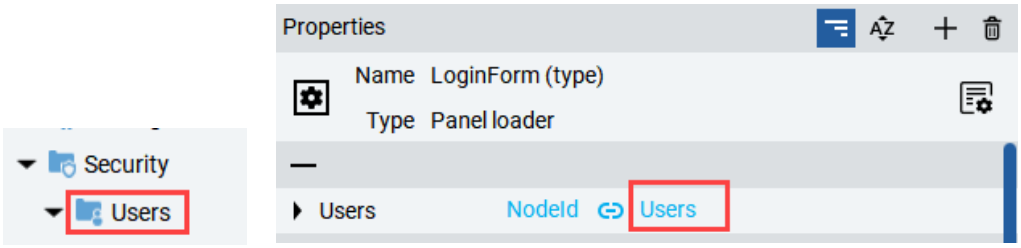


4. Select **Close** to close the **Libraries** window.

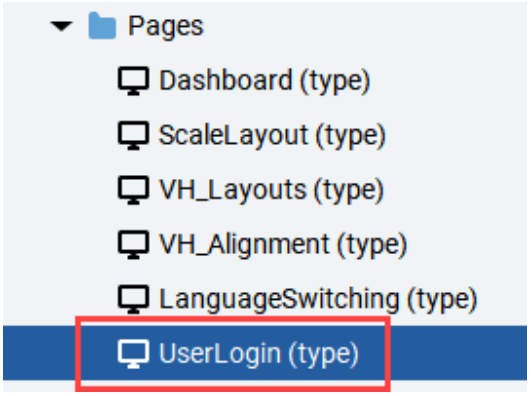
5. Expand **LoginForm** under the Screens folder and click **LoginForm (type)** to access its properties.



6. Drag and drop the **Users** folder from the **Security** node to the **Users NodeId** property.



7. In Project view navigate to **UI** > **Pages** and rename **Page10** to "**UserLogin**".



8. Right-click **UserLogin** and select **New** > **Screens** > **LoginForm** > **LoginForm.**

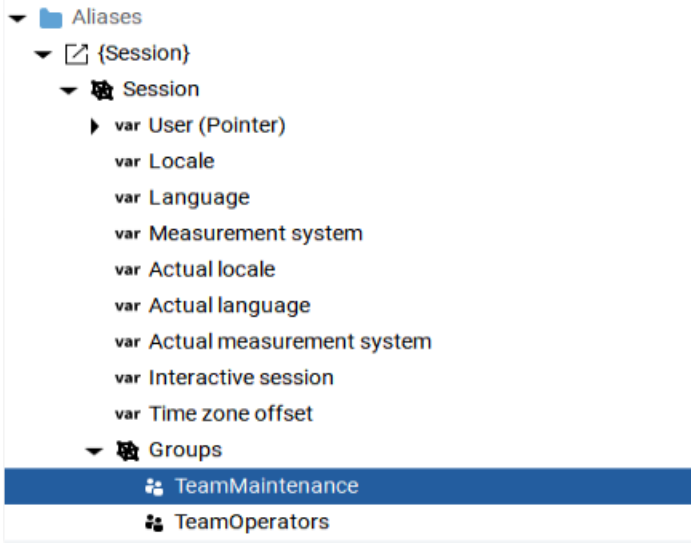
© 2025 Rockwell Automation, Inc. All rights reserved.

Working with Users and Groups

Let's secure a screen making not accessible to TeamOperators and make couple of objects only enabled for TeamMaintenance

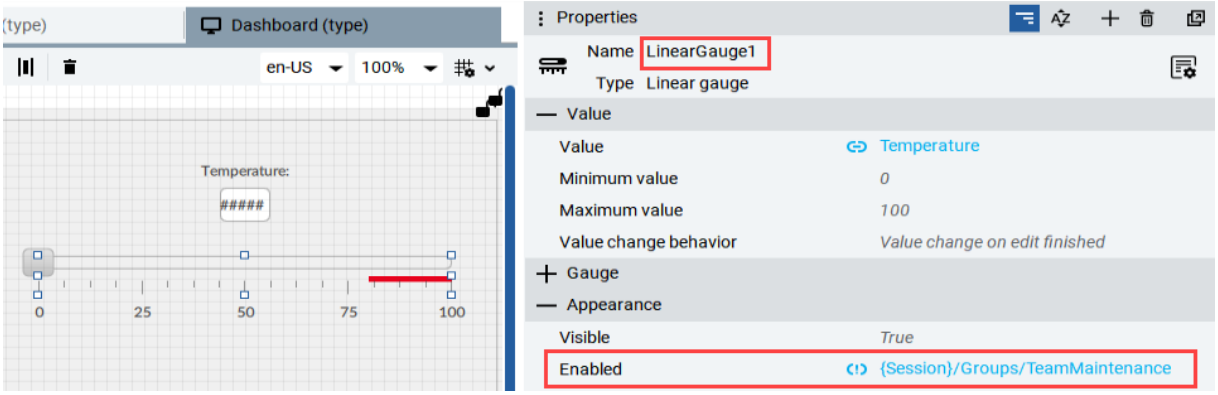
1. In Project view navigate to **UI** > **Pages** > **Dashboard**.
2. Select the Textbox that we used in Section 1 to display the temperature variable value. in this lab, it is called Textbox1.
3. For Textbox1 properties, mouse-over the **Enabled** property and click the **Add Dynamic Link** icon .

4. In the dynamic link browser, navigate to **Aliases** > **{Session}** > **Session** > **Groups** > **TeamMaintenance**.



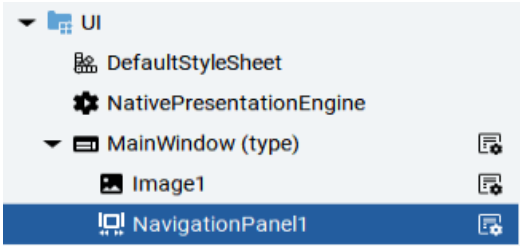
5. Click **Select** to close the dynamic link browser. The Enabled property will look as follows

6. Still on the Dashboard screen, repeat the previous steps for the Linear Gauge that we used to also change the temperature variable value in Section 1. in this lab, it is called LinearGauge1.



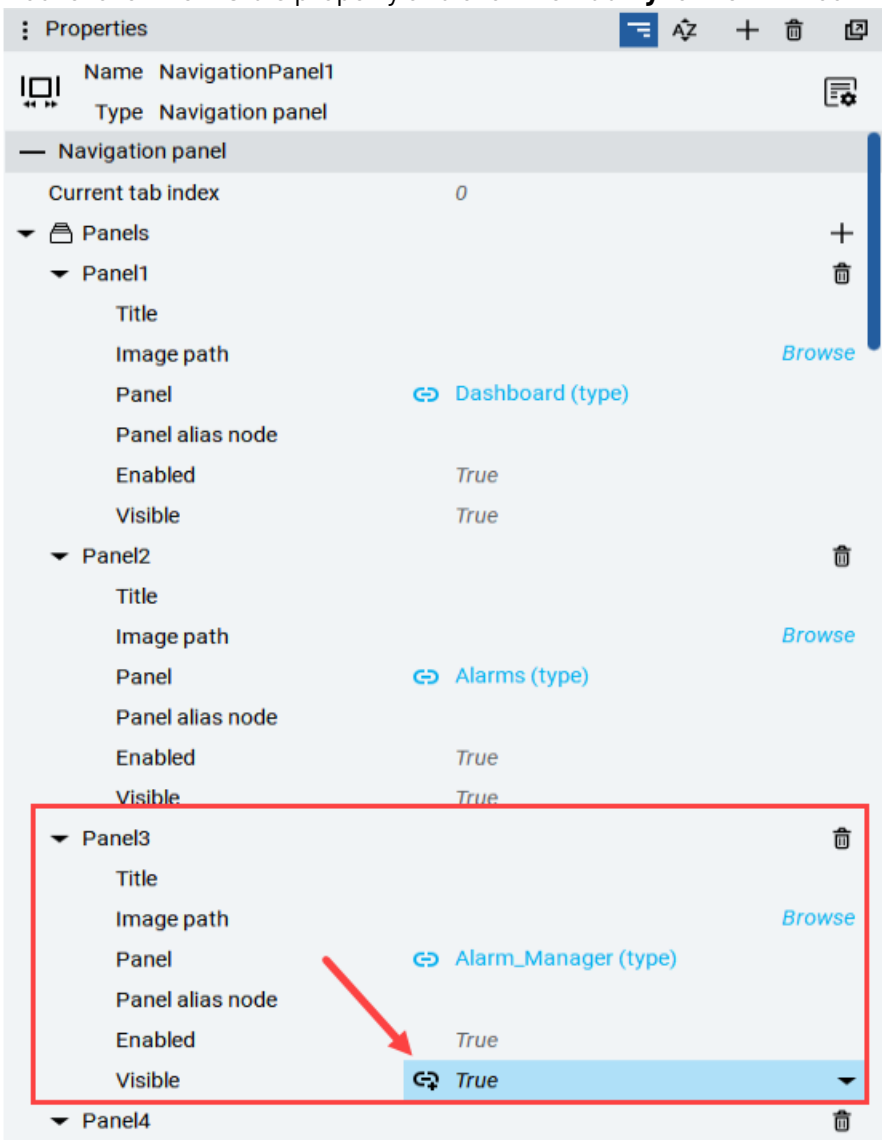
© 2025 Rockwell Automation, Inc. All rights reserved.

7. We can also prevent a screen from being displayed to TeamOperators. In Project view navigate to **UI** > **MainWindow (type)** > **NavigationPanel1**.



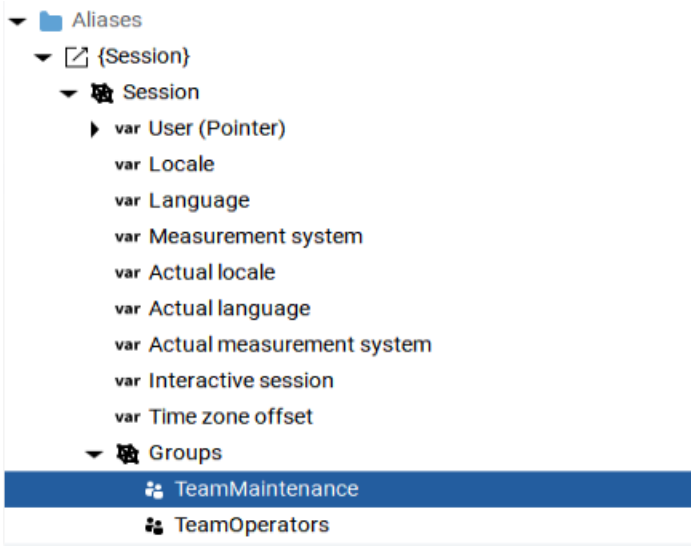
8. Looking at the properties for NavigationPanel1 on the right and more specifically the list of panels that you can navigate to, we will link the visibility of the Alarm_Manager panel to a specific group. If you have not completed the Alarming lab, Do Not Use the Dashboard or the UserLogin panels for this step. Select any other panel listed in the NavigationPanel1 properties.

9. Mouse-over the **Visible** property and click the **Add Dynamic Link** icon .



© 2025 Rockwell Automation, Inc. All rights reserved.

10. In the dynamic link browser, navigate to **Aliases** > **{Session}** > **Session** > **Groups** > **TeamMaintenance**.



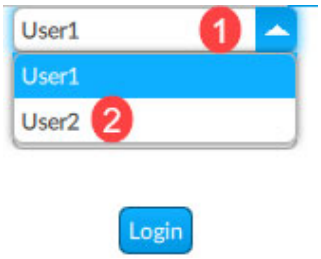
11. Click **Select** to close the dynamic link browser.

12. From the toolbar, click the **Run on Emulator** icon ▶.

13. Click the **UserLogin** tab

14. Confirm the following: When logged in as Anonymous or User1, you cannot change the temperature variable value from the Dashboard screen and the Alarm_Manager screen is not visible to you.

15. Click the **User** drop-down arrow button and click **User2** and login.



16. Confirm the following: When logged in as User2, you can change the temperature variable value from the Dashboard screen as well as having the Alarm_Manager screen visible so you can navigate to it.

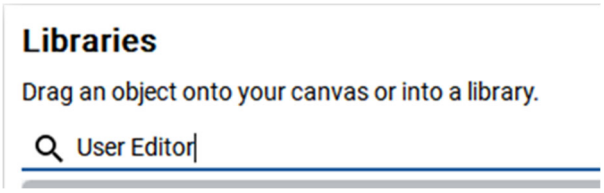
© 2025 Rockwell Automation, Inc. All rights reserved.

Runtime user management

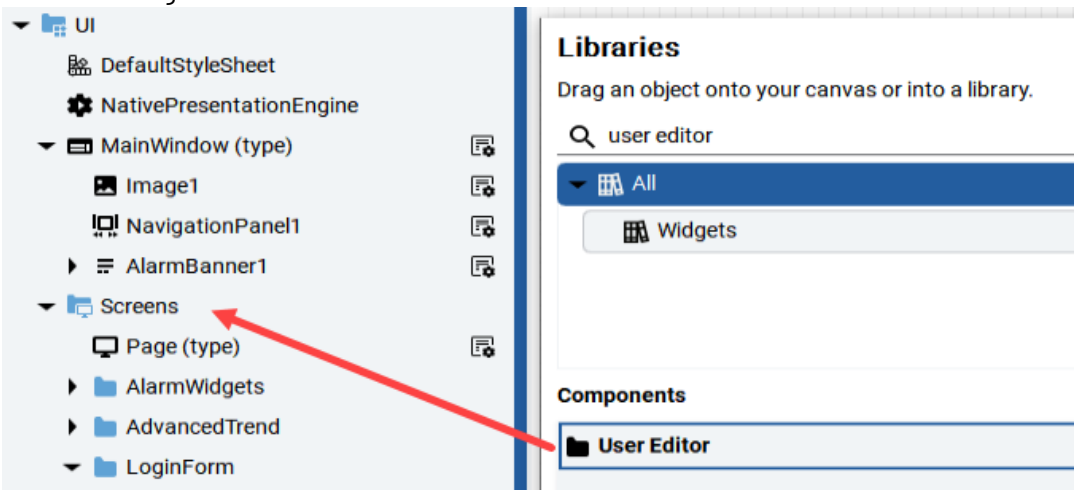
With Optix, user management like adding a new user, even when not using a centralized domain controller, is also supported at runtime.

1. Click the **Template Libraries** icon.

2. Type “User Editor” in the **Libraries** search field.

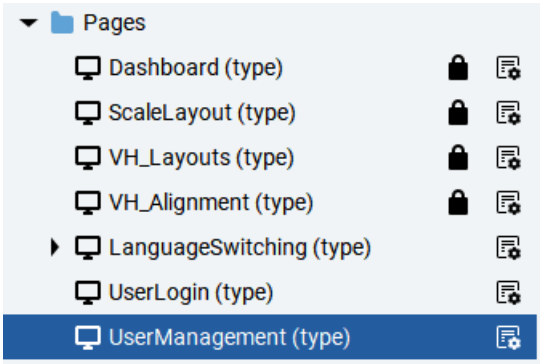


3. Click and drag the **User Editor** folder to the **Screens** folder.



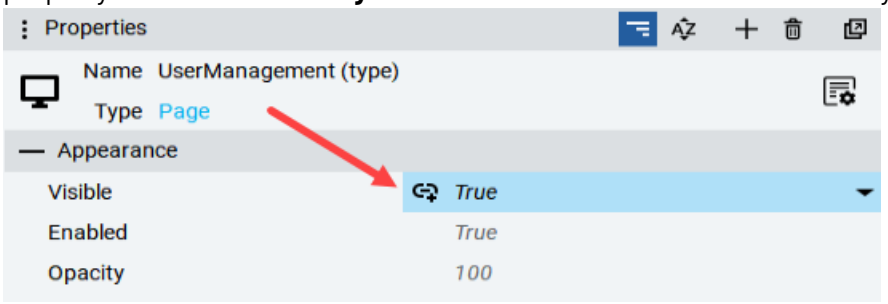
4. Select **Close** to close the **Libraries** window.

5. In Project view navigate to **UI** > **Pages** and rename **Page11** to “**UserManagement**”.

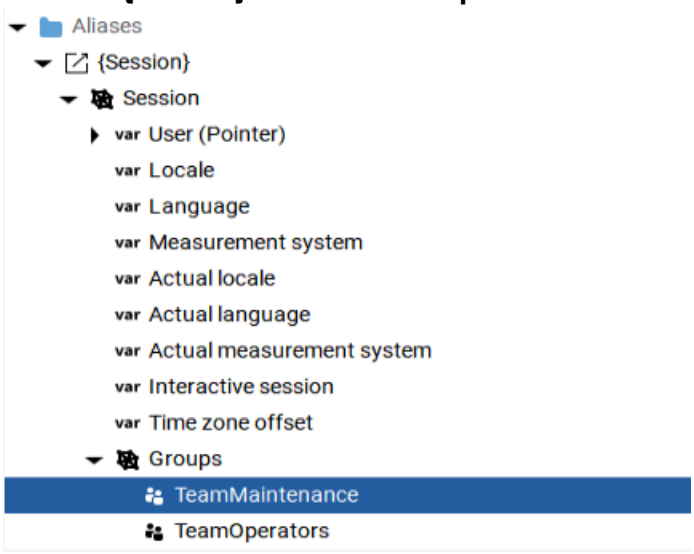


© 2025 Rockwell Automation, Inc. All rights reserved.

6. Looking at the Properties for the **UserManagement** page on the right, mouse-over the **Visible** property and click the **Add Dynamic Link** icon to secure its visibility.



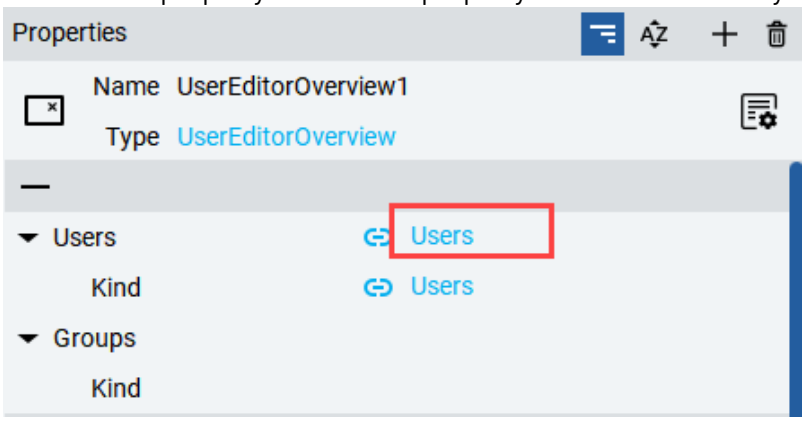
17. In the dynamic link browser, navigate to **Aliases** > **{Session}** > **Session** > **Groups** > **TeamMaintenance**.



18. Click **Select** to close the dynamic link browser.

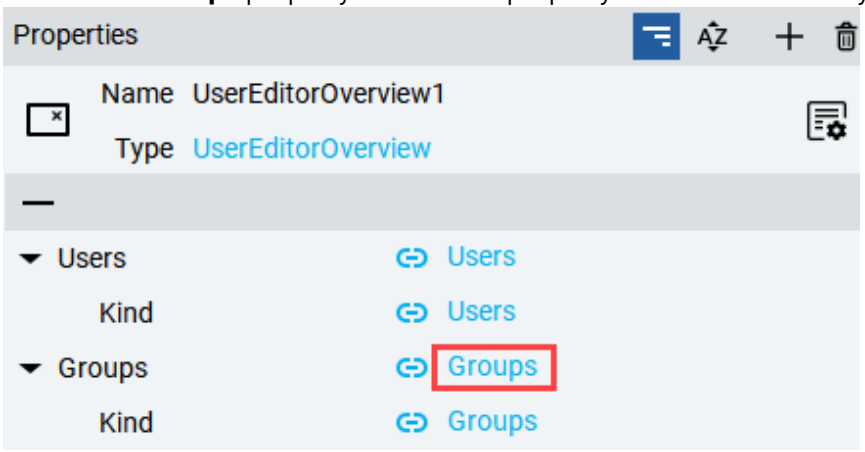
7. We are ready to add the template runtime user management widget we added to the project earlier. Right-click **UserManagement** and select **New** > **Screens** > **UserEditor** > **UserEditorOverview.**

19. In the Properties for **UserEditorOverview1**, drag and drop the **Users** folder in the **Security** node to the **Users** property. The **Kind** property will be automatically assigned.



© 2025 Rockwell Automation, Inc. All rights reserved.

20. Still in the Properties for **UserEditorOverview1**, drag and drop the **Groups** folder in the **Security** node to the **Groups** property. The **Kind** property will be automatically assigned.



Any user that logs in that is a member of **TeamMaintenance** will have access to the **UserManagement** panel.

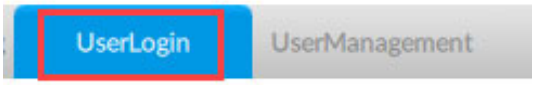
Note: If performing navigation by using a button, this same dynamic link can be applied to the Enabled or Visible property of the button.

21. Set the UserEditorOverview1 Horizontal and Vertical alignment properties to Stretch to fit the entire page.

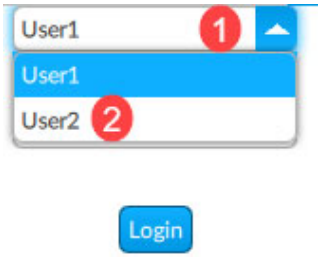
Emulate and Explore

1. From the toolbar, click the **Run on Emulator** icon ▶.

2. Click the **UserManagement** tab on the Navigation panel.
3. Notice that the **UserManagement** panel is blank. This is because only users of TeamMaintenance can interact with this screen.
22. Click the **UserLogin** tab.



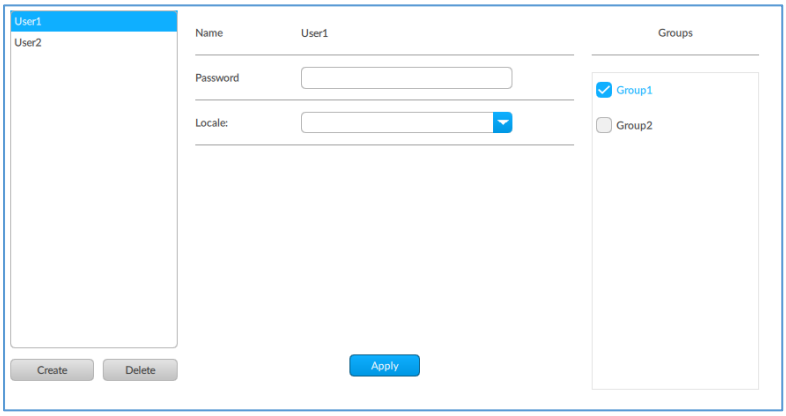
4. Click the **User** drop-down arrow button and click **User2** and login.



The **UserManagement** tab is available for User2.

© 2025 Rockwell Automation, Inc. All rights reserved.

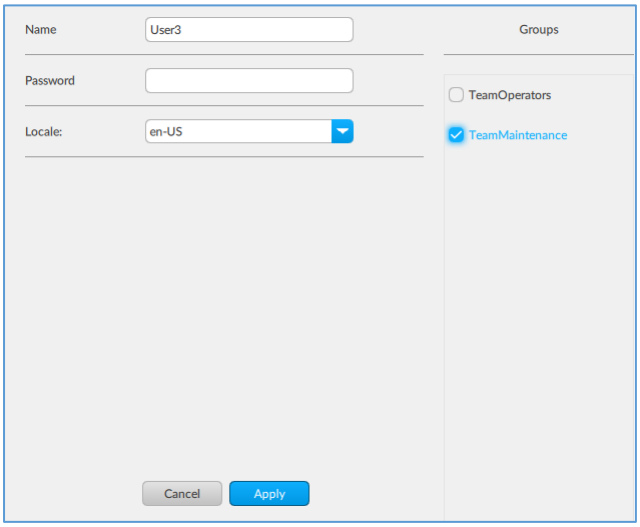
23. Click the **UserManagement** tab on the Navigation panel. If the language is not English, simply navigate to the LanguageSwicthing tab, select the English button and go back to UserManagement.



5. Click the **Create** button.

6. Assign the following to create a new user – press **Enter** after any text entry – and click **Apply** when finished.

   - Name: ***User3*** (leave password blank)
   - Locale: **en-US**
   - Groups: **TeamMaintenance**



   User3 has now been created.

7. Click on the **UserLogin** tab in the Navigation panel.

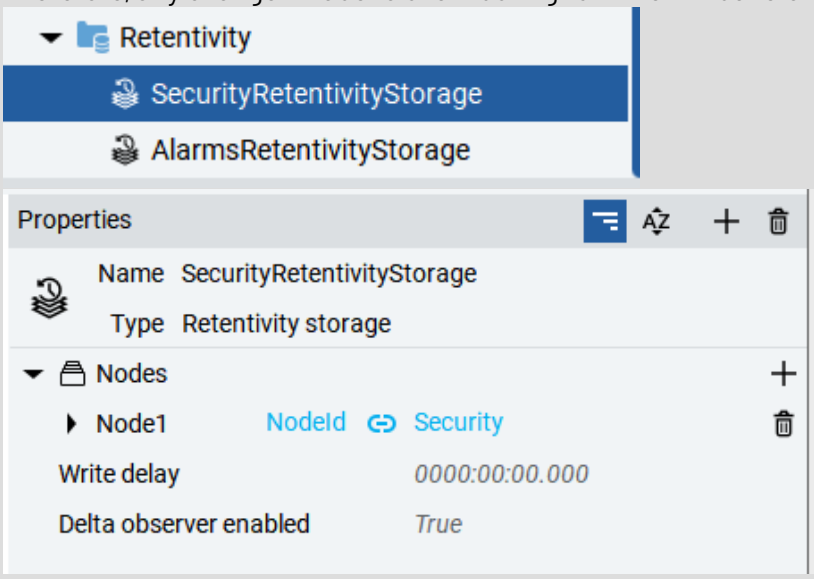
8. Click the **Logout** button.

© 2025 Rockwell Automation, Inc. All rights reserved.

9. Click the **User** drop-down arrow button and click **User3**.
10. Click the **Login** button.
11. Observe the following:
    - User3 is the Current User.
    - The UserManagement tab is available for User3.
    - User 3 can access anything that TeamMaintenance can access
12. Close the **Emulator**.
13. (Optional) Can you determine the total feature tokens needed to run this project in production? Which components in this project are impacting the runtime entitlement size?

RETENTIVITY

There is a Retentivity folder that is part of any project by default. Also, by default, within this folder, there is a database called SecurityRetentivityStorage which is assigned to the Security node. Therefore, any changes made to users during runtime will be retained.



SUMMARY

FactoryTalk Optix provides the capability of creating local users and groups and/or using Windows domain accounts. When developing the project, designers can use the **Template Libraries** to quickly and easily add pre-built login and user management functionality. Finally, FactoryTalk Optix provides the ability to add locale information to individual users. Therefore, when a user logs into the application, the text, values, and engineering units will switch to the user's native language and measurement system.

You have completed the Security lab.

© 2025 Rockwell Automation, Inc. All rights reserved.

Aliases (15 Minutes)

Objectives

- Create a Model object as a type
- Create a motor faceplate
- Use aliasing to display data from multiple motors dynamically

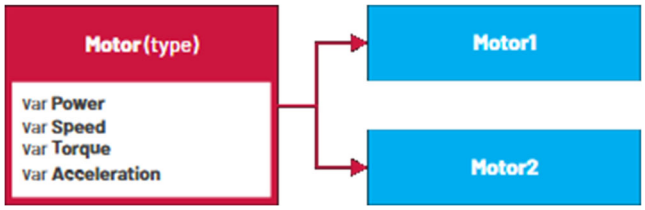
Scenario

Aliases allow you to have instances of an object display different values. For example, by using an alias, an instance of a single motor faceplate can show data for multiple motors dynamically.

Lab procedure

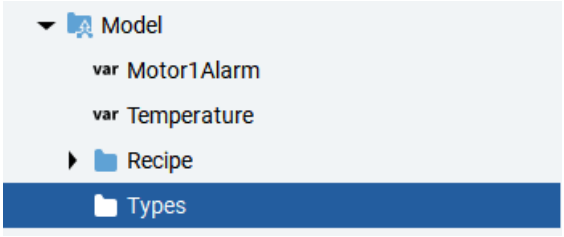
Create a Model Object as a type

Our goal in this lab is to create the following but we will use only the speed property:



1. In Project view, right-click the **Model** folder and select **New** > **Folder**. Folder1 appears under Screens.

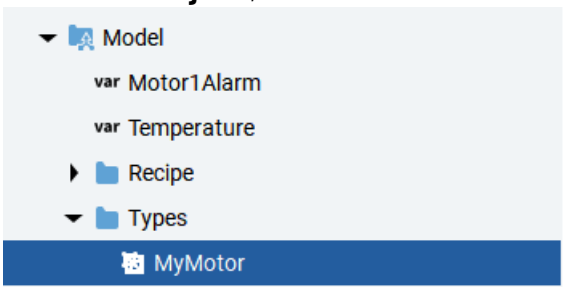
2. Mouse-over **Folder1**, select the edit icon , and enter "**Types**" as the new name.



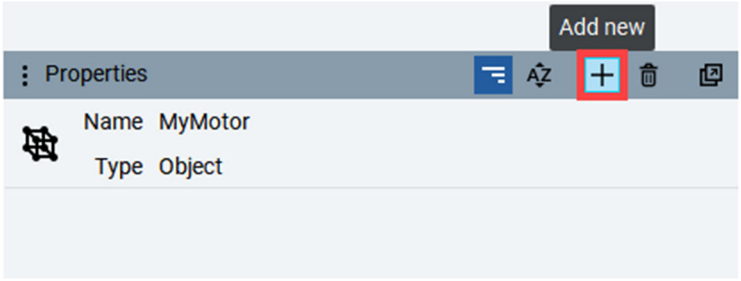
© 2025 Rockwell Automation, Inc. All rights reserved.

3. Right click on **Types** and select **New** > **Object**

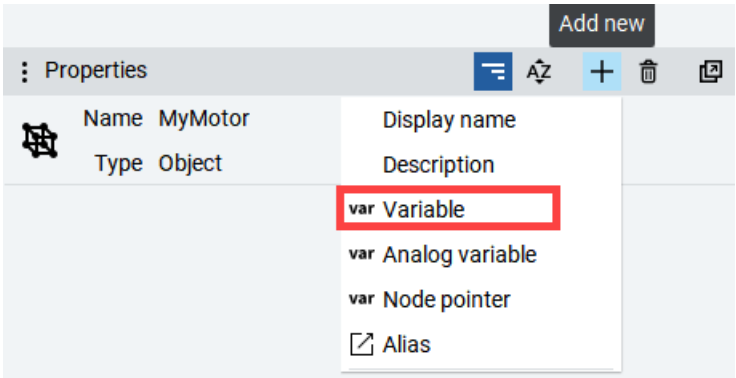
4. Mouse-over **Object1**, select the edit icon , and enter "**MyMotor**" as the new name.



5. In the Properties pane on the right for *MyMotor*, click the **Add new** + icon.



6. Add a **Variable**.

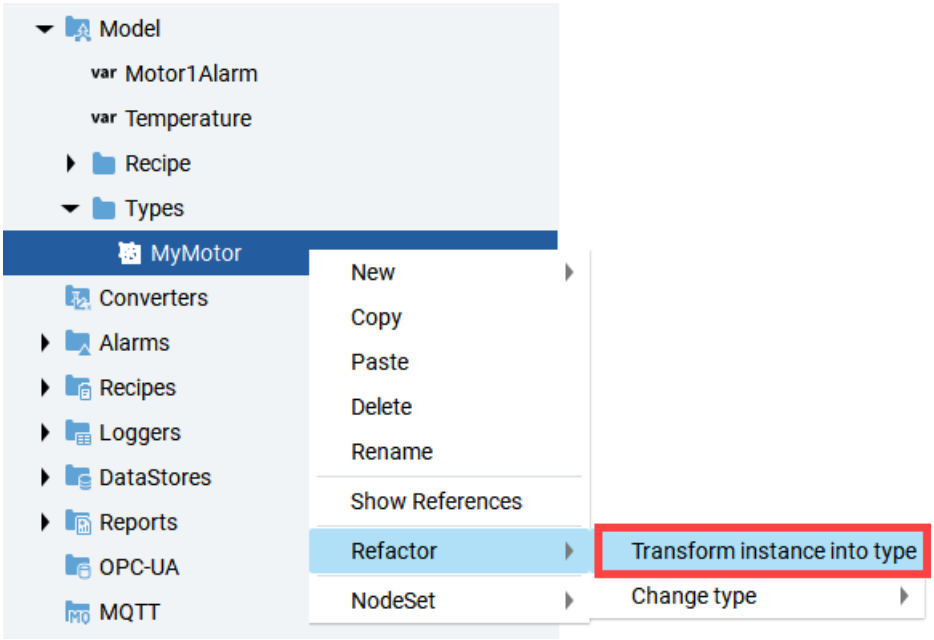


© 2025 Rockwell Automation, Inc. All rights reserved.

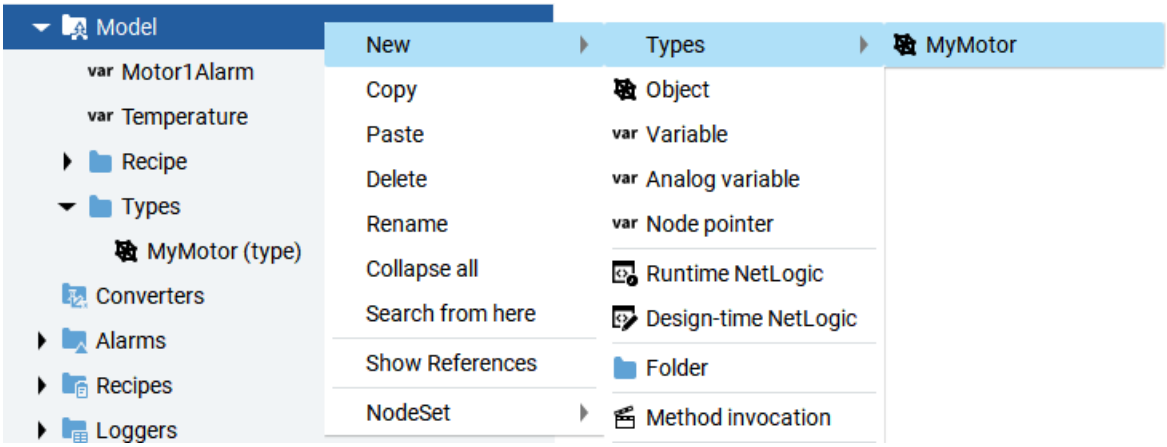
7. Mouse-over **Variable1**, select the edit icon , and enter "**Speed**" as the new name.

Note: More complex objects can be created. For this lab, we will get the idea across using this very simple object.

8. Back in Project view, right-click the **MyMotor** object and select **Refactor** > **Transform instance into type**.

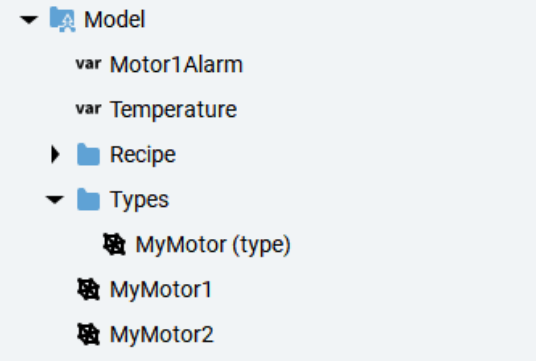


9. We are ready to create two instances of MyMotor. in Project view, right-click the **MyMotor** object and select **New** > **Types** > **MyMotor.**



© 2025 Rockwell Automation, Inc. All rights reserved.

10. Repeat the previous step to create the two instances.
MyMotor1 and MyMotor2 are now created as instances of MyMotor type.



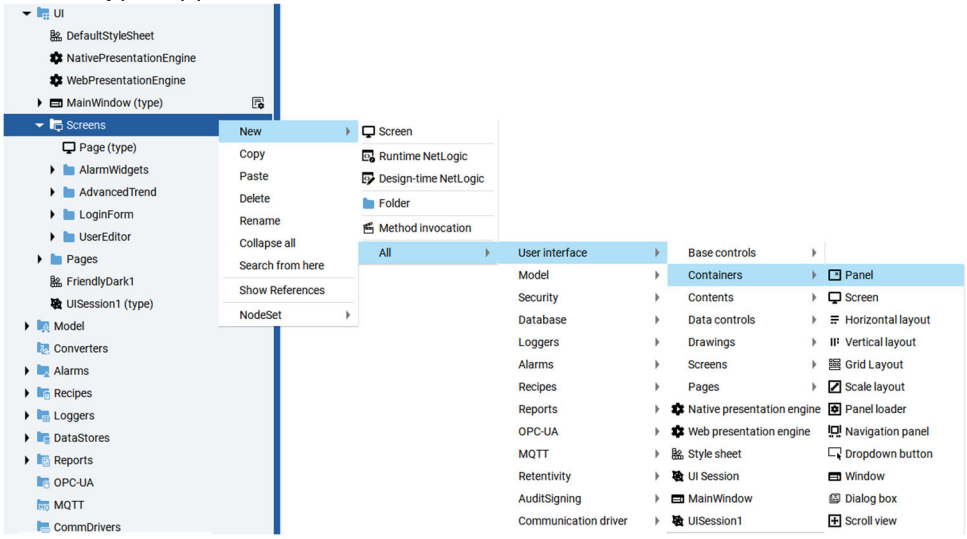
11. Select **MyMotor1** and change its initial value for **Speed** to **10**.

12. Select **MyMotor2** and change its initial value for **Speed** to **20**.

13. **Save** the project.

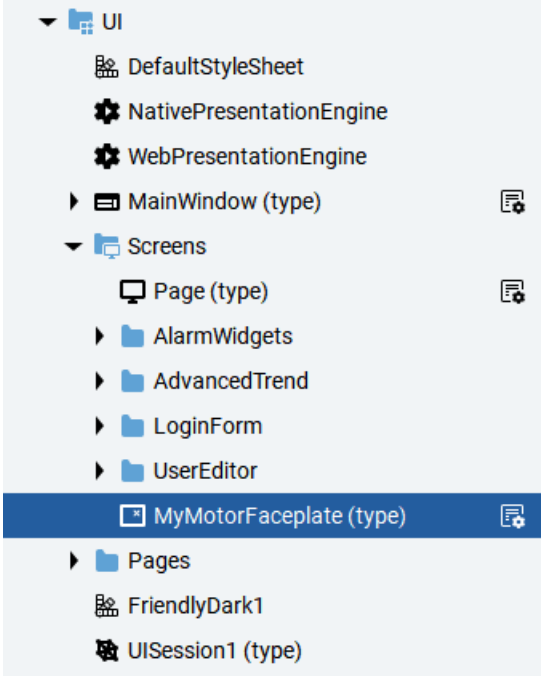
Create the Motor widget (or faceplate)

1. In Project view, right-click **Screens** and select **New** > **All** > **User Interface** > **Containers** > **Panel**.
Panel1 (type) appears under Screens.

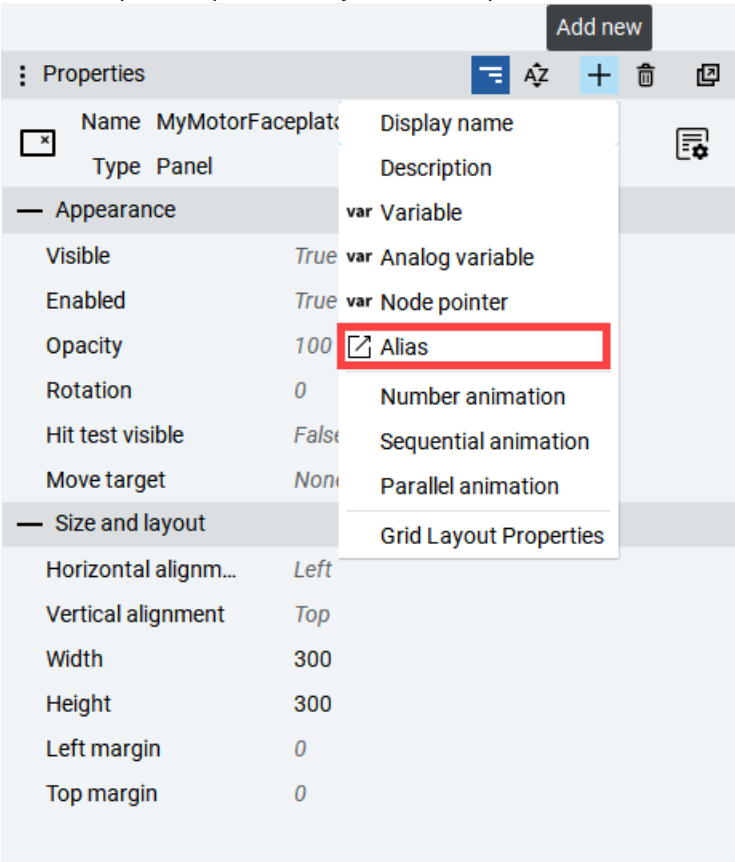


© 2025 Rockwell Automation, Inc. All rights reserved.

2. Mouse-over **Panel1 (type)**, select the edit icon , and enter "**MyMotorFaceplate**" as the new name.

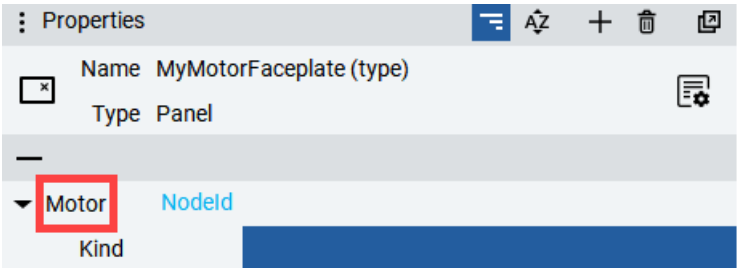


3. In the Properties pane for MyMotorFaceplate, click the **Add new** icon and select **Alias**.



© 2025 Rockwell Automation, Inc. All rights reserved.

4. Mouse-over **Alias1**, select the edit icon , and enter "**Motor**" as the new name.

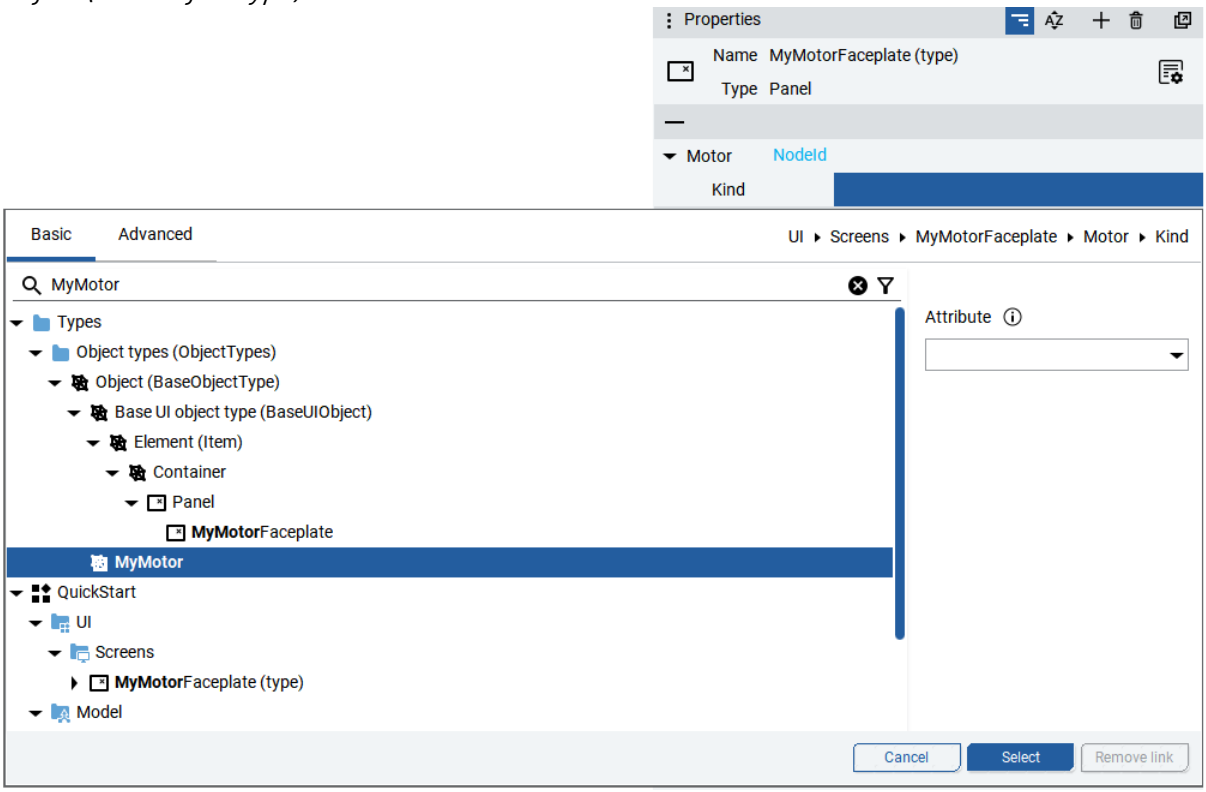


Note: **New!** starting with FactoryTalk Optix version 1.3, multiple Aliases can be configured.

An alias contains a Kind property with a value that is a reference node. This reference is to the object or variable type from the source node. For example, to design a widget that displays the properties of a Motor type, create a Motor type with two properties -- Speed and Acceleration. Create a Motor Panel type and add the Motor alias. Set the Kind property of the alias to the Motor type node. You can then add two labels or other UI control and dynamically link the Speed and Acceleration properties through the Motor alias.

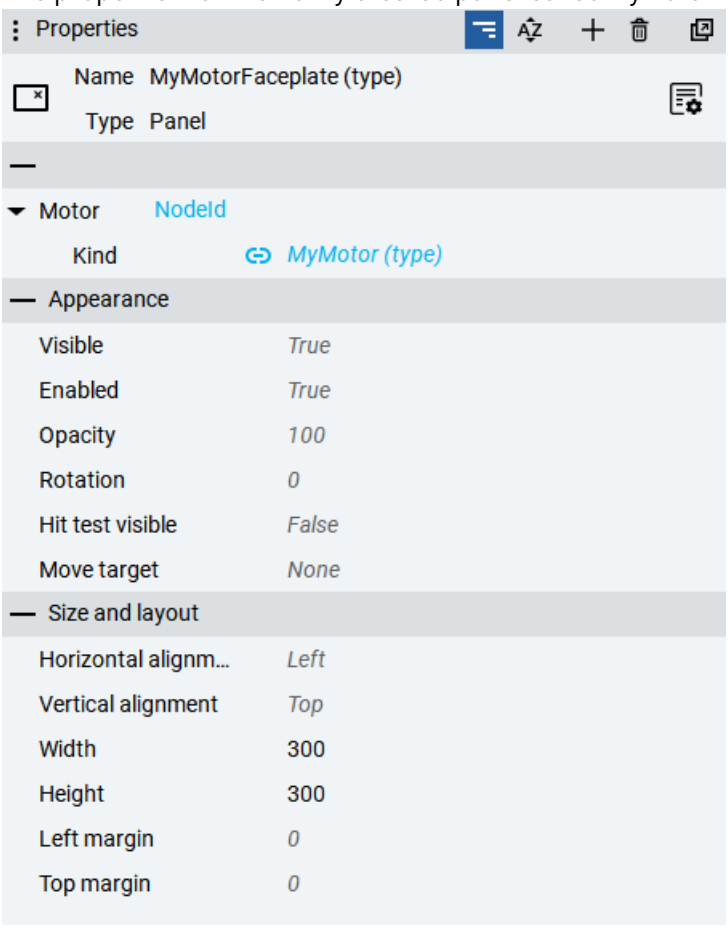
5. Mouse-over the **Kind** property and click the **Set kind** icon .

6. In the browse window, use the **search** to look for *MyMotor*, select the **MyMotor** type object under *Object (BaseObjectType)* as shown below and click **Select**.



© 2025 Rockwell Automation, Inc. All rights reserved.

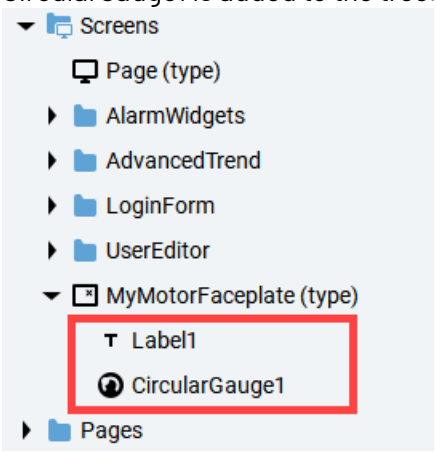
The properties for the newly created panel called MyMotorFaceplate will look like the following:



7. **Save** the project.

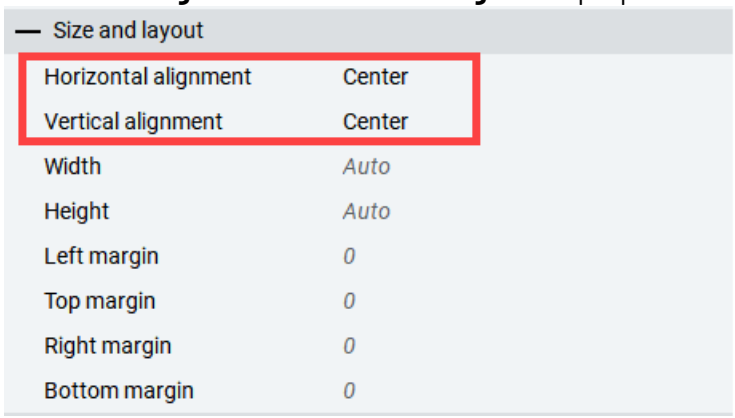
Add graphic elements to the widget

1. In Project view, right-click the newly added **MyMotorFaceplate** panel, and select **New** > **Base controls** > **Label.** Label1 is added to the tree.
2. Right-click **MyMotorFaceplate** again and select **New** > **Base controls** > **Circular gauge.** CircularGauge1 is added to the tree.



© 2025 Rockwell Automation, Inc. All rights reserved.

3. In Project view, select **Label1** to access its properties in the Properties pane on the right and set the **Horizontal alignment** and **Vertical alignment** properties to **Center**.



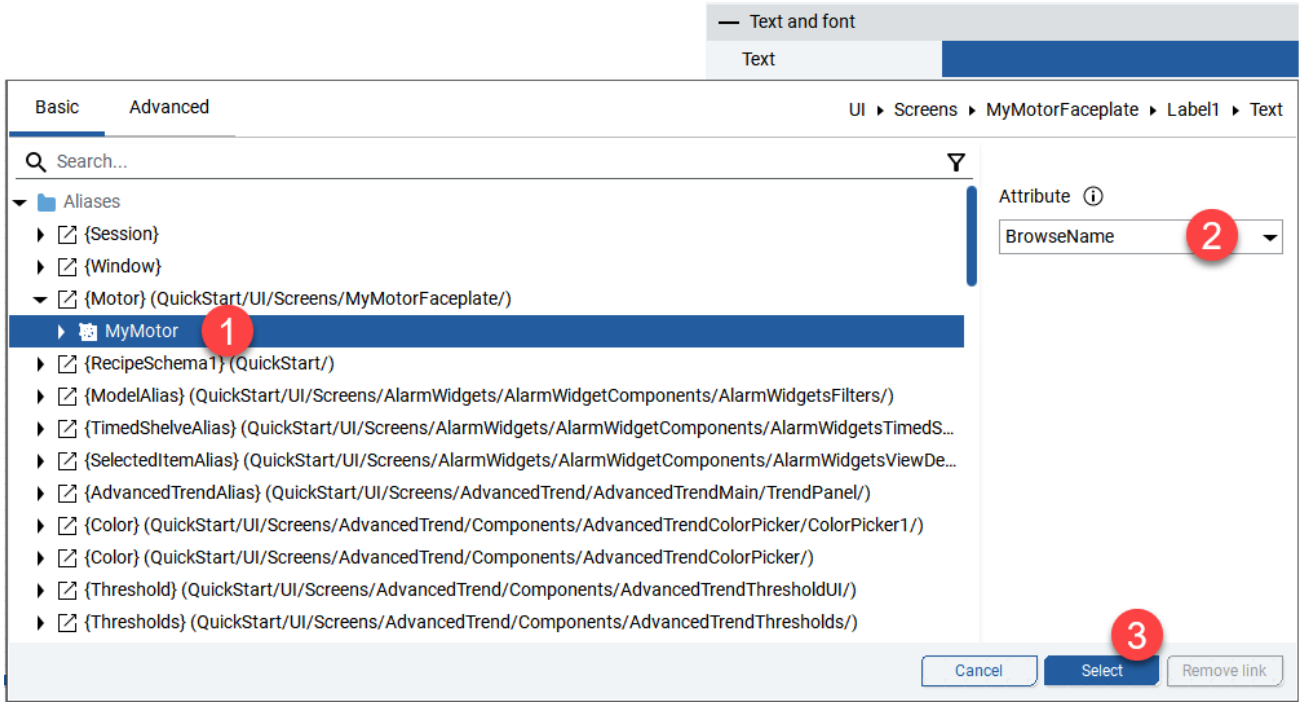
4. Still in the Label1 Properties pane, mouse-over the **Text** property and click the **Add Dynamic Link** icon .

5. Perform the following 3 steps to select the correct dynamic link

   i. You may have to scroll all the way to the top and then select **Aliases** > **{Motor} (QuickStart/UI/Screens/MyMotorFaceplate** and select **MyMotor**.

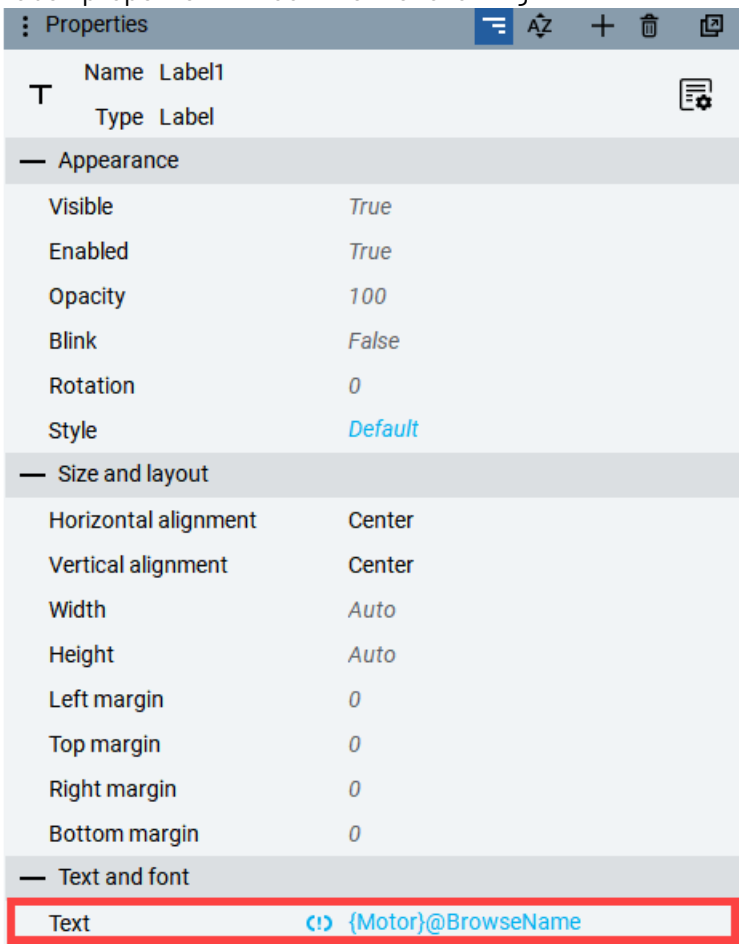
   ii. Set the *Attribute* to **BrowseName**

   iii. Click **Select** to close.

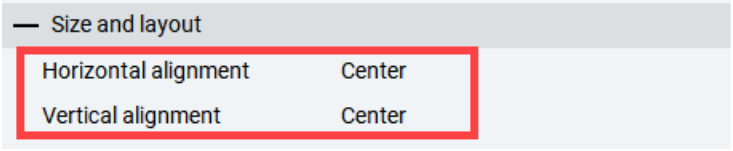


© 2025 Rockwell Automation, Inc. All rights reserved.

Label1 properties will look like the following:



6. We are ready to change the properties for the circular gauge. In Project view, select the newly added **CircularGauge1** to access its properties in the Properties pane on the right and set the **Horizontal alignment** and **Vertical alignment** properties to **Center**.



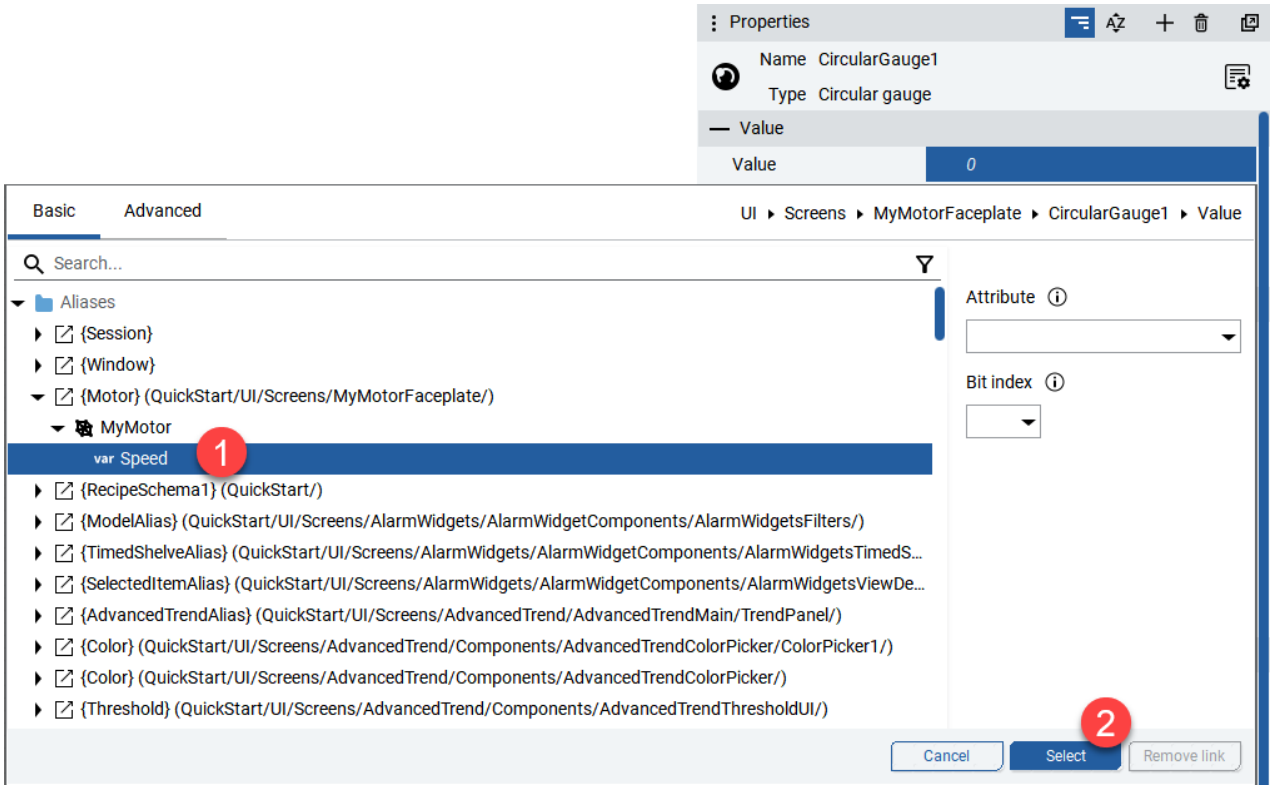
7. Still in the CircularGauge1 Properties pane, mouse-over the **Value** property and click the **Add Dynamic Link** icon .

© 2025 Rockwell Automation, Inc. All rights reserved.

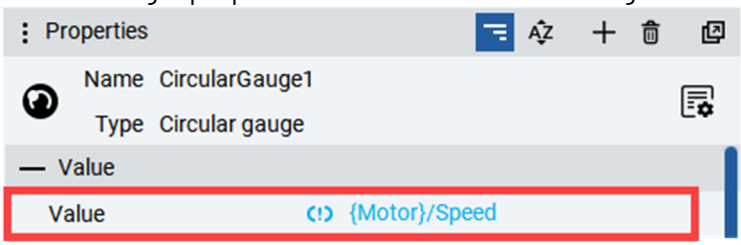
8. Perform the following 2 steps to select the correct dynamic link

    i. Select **Aliases** > **{Motor} (QuickStart/UI/Screens/MyMotorFaceplate** and select **MyMotor** > **var Speed**

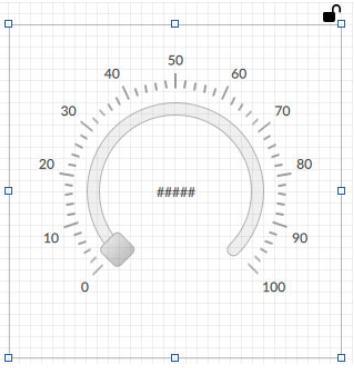
    ii. Click **Select** to close.



CircularGauge1 properties will look like the following:



9. Our final step here is to double-click the **MyMotorFaceplate** panel so it opens in the Editor and make sure it matches the screen capture below.

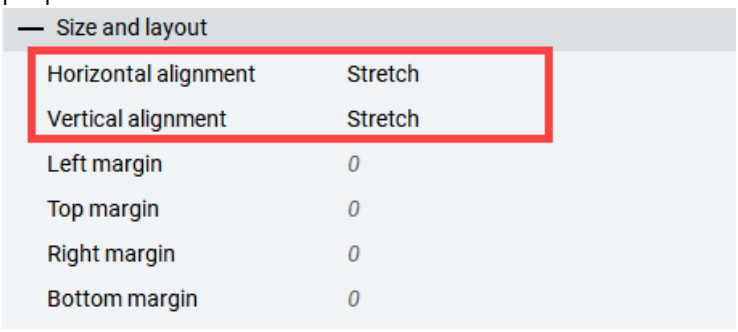


© 2025 Rockwell Automation, Inc. All rights reserved.

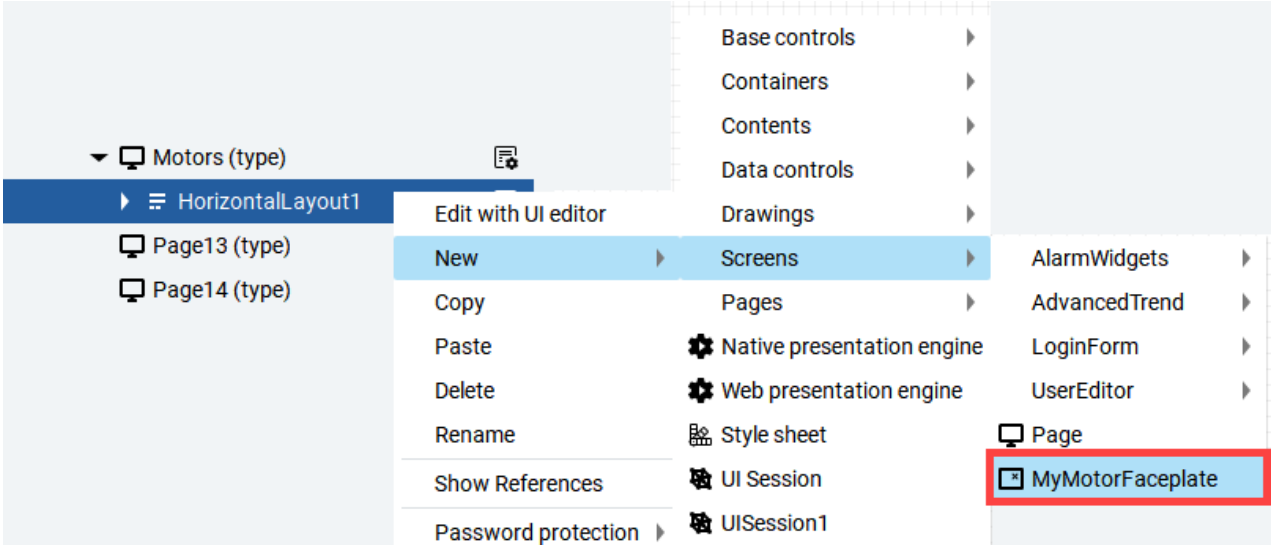
10. **Save** the project.

Create instances of our newly created widget

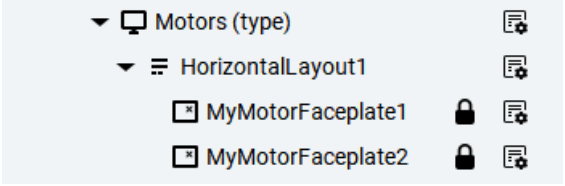
1. In Project view navigate to **UI**-> **Pages** and rename **Page12** to "**Motors**".
2. We will add a horizontal layout container on top of Motors to automatically arranges graphic objects horizontally next to each other. Right click on **Motors** and select **New** > **Containers** > **Horizontal layout**. HorizontalLayout1 is added to the tree.
3. For the newly added HorizontalLayout1, set the **Horizontal alignment** and **Vertical alignment** properties to **Stretch**.



4. We are ready to create two instances of our widget called MyMotorFaceplate. In Project view, right-click **HorizontalLayout1**, select **New** > **Screens** > **MyMotorFaceplate.** MyMotorFaceplate1 is added to the tree under Motors.



5. Repeat the previous step to add one more instance. Right-click **HorizontalLayout1** again, select **New** > **Screens** > **MyMotorFaceplate.** MyMotorFaceplate2 is added to the tree under Motors.

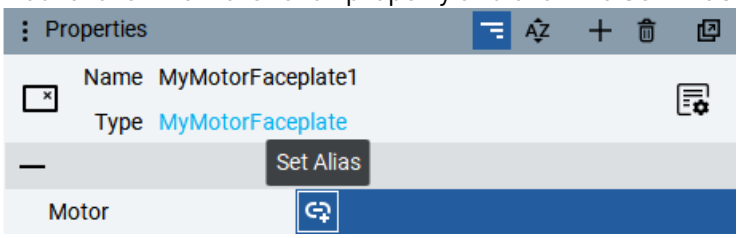


© 2025 Rockwell Automation, Inc. All rights reserved.

Let's link MyMotorFaceplate1 to MyMotor1 and MyMotorFaceplate2 to MyMotor2.

6. Select the newly created **MyMotorFaceplate1** to access its properties in the Properties pane on the right.

7. Mouse-over the **Motor** alias property and click the **Set Alias** icon .

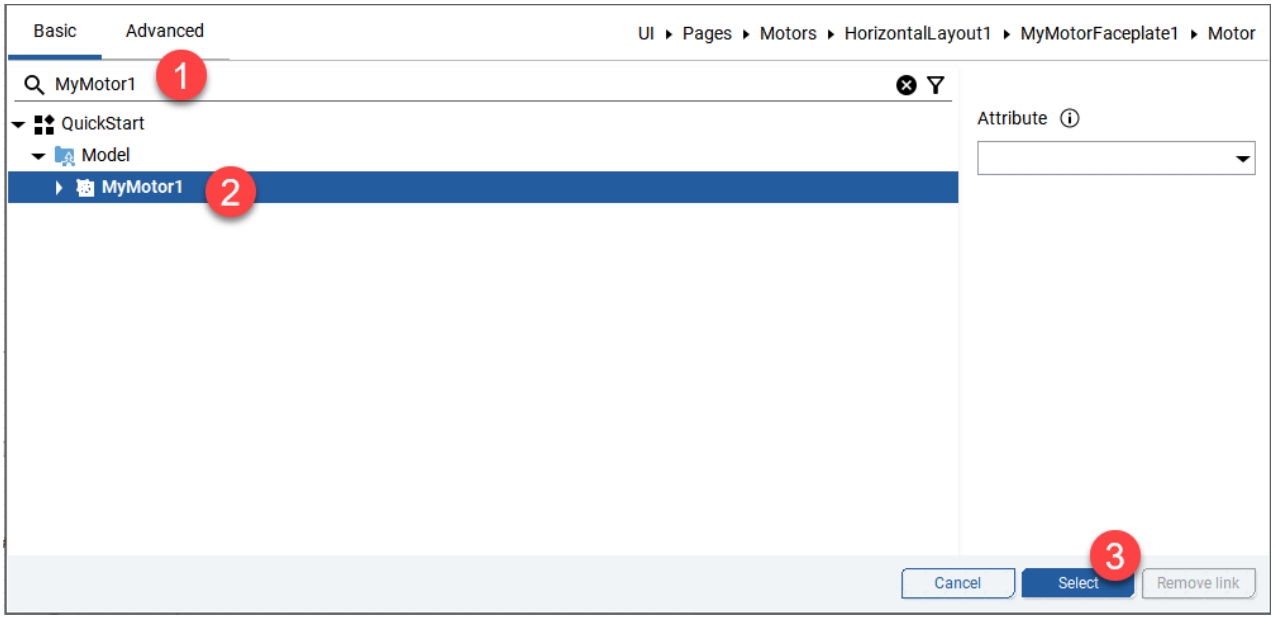


8. In the Set Alias browser window,

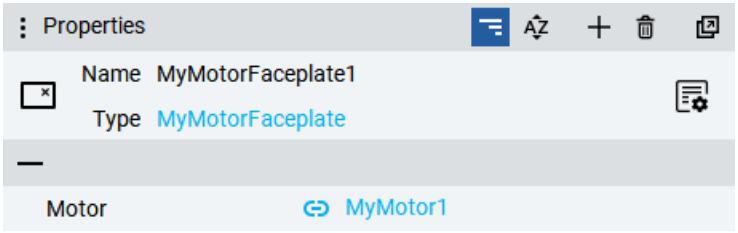
   i. Enter **MyMotor1** in the search field.

   ii. Click **MyMotor1** under the Model folder.

   iii. Click **Select** to close.



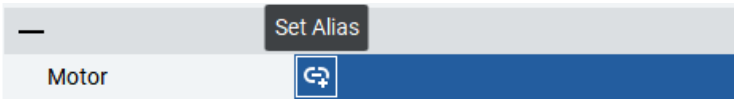
MyMotorFaceplate1 properties will look like the following:



9. Link MyMotorFaceplate2 to MyMotor2. Select the newly created **MyMotorFaceplate2** to access its properties in the Properties pane on the right.

© 2025 Rockwell Automation, Inc. All rights reserved.

10. Mouse-over the **Motor** alias property and click the **Set Alias** icon .

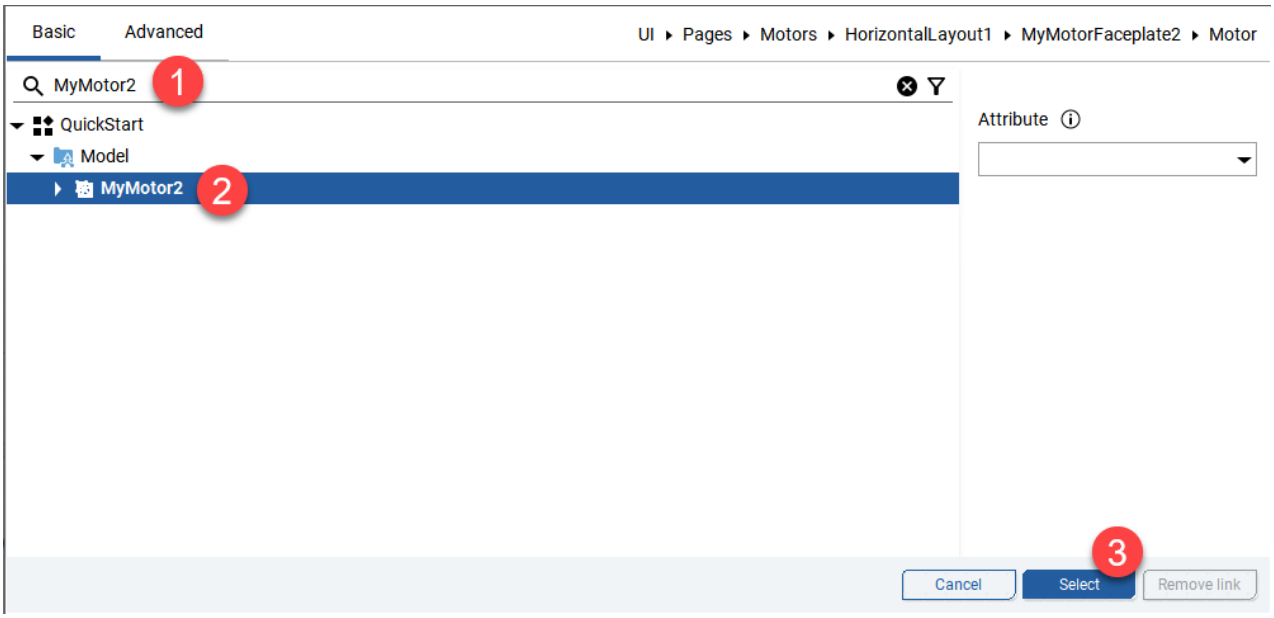


11. In the Set Alias browser window,

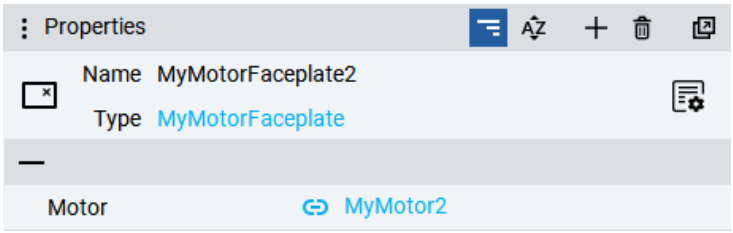
    iv. Enter **MyMotor2** in the search field.

    v. Click **MyMotor2** under the Model folder.

    vi. Click **Select** to close.



MyMotorFaceplate2 properties will look like the following:

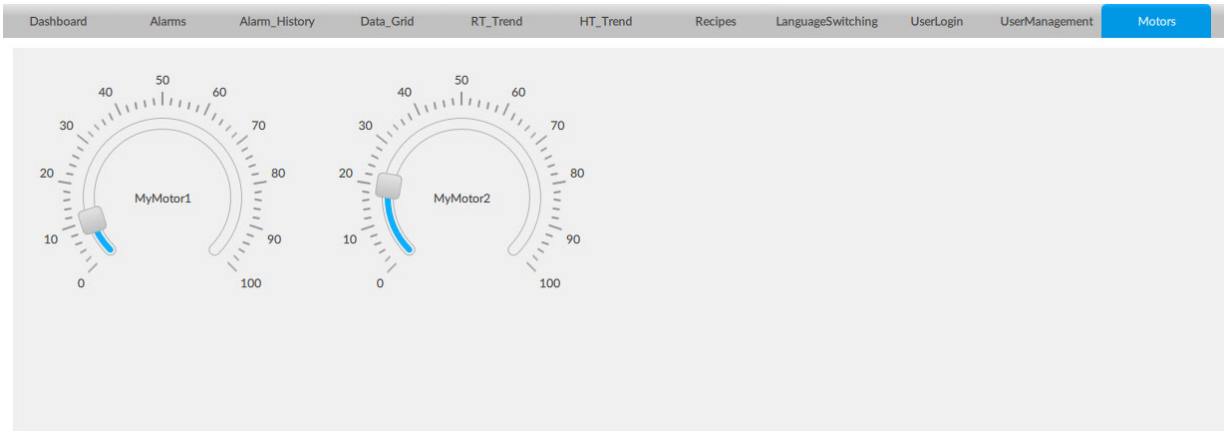


12. **Save** the project.

13. From the toolbar, click the **Run on Emulator** icon .

© 2025 Rockwell Automation, Inc. All rights reserved.

14. Wait for the Optix Runtime application to launch and navigate to the **Motors** tab. Note the two instances properly labeled based on the BrowseName of the MyMotor object and using the initial values for the Speed variables we set up earlier: MyMotor1 Speed has an initial value of 10 and MyMotor2 Speed has an initial value of 20.



15. Close the **Emulator**.

16. (Optional) Can you determine the total feature tokens needed to run this project in production? Which components in this project are impacting the runtime entitlement size?

Note: **New!** Starting with FactoryTalk Optix version 1.4, you can protect a container or a drawing with a password to prevent other FactoryTalk Optix Studio users from accidentally editing the protected object children. Check the help for more information.

You have completed the Aliases lab.

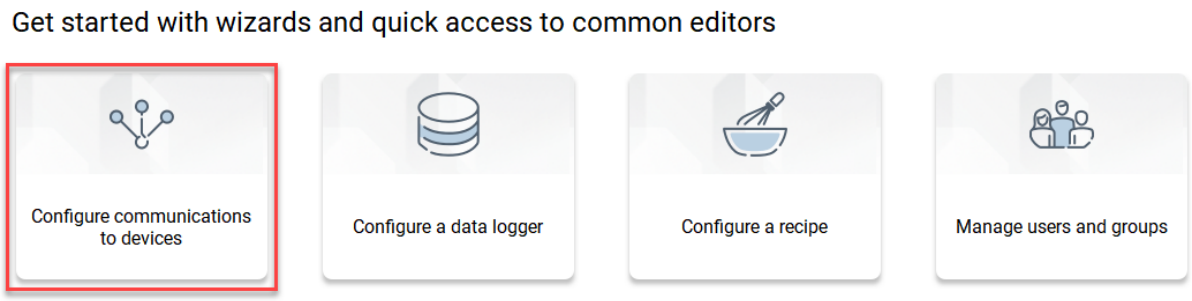
© 2025 Rockwell Automation, Inc. All rights reserved.

Appendix A – Configure Communications: Logix Station

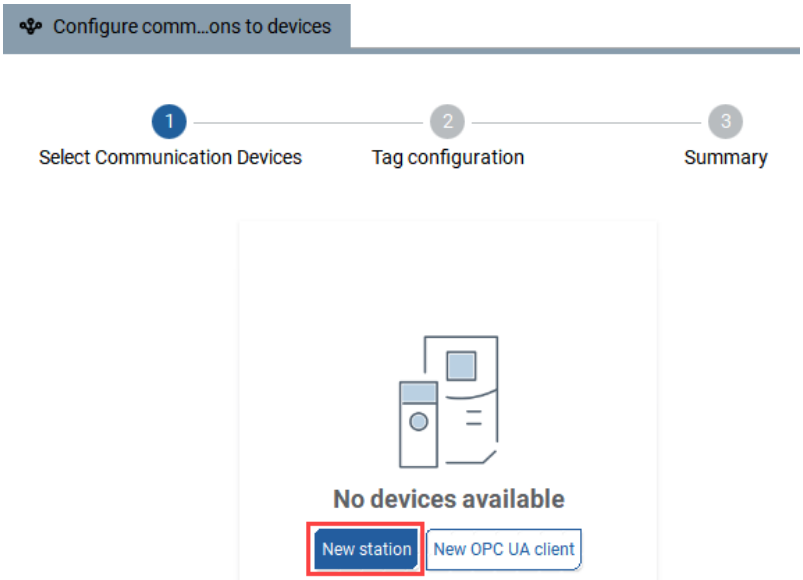
The following drivers are supported with FactoryTalk Optix

- RA EtherNet/IP
- RA Micro Controller (**New!** Starting with FactoryTalk Optix version 1.4)
- CODESYS
- MELSEC FX3U
- MELSEC Q
- Modbus
- OMRON EtherNet/IP
- OMRON Fins
- S7TCP
- S7 TIAPROFINET
- TwinCAT

1. Let's configure a RA EtherNet/AP station to communicate with Logix. For this section, you will select the **Configure communications to devices** wizard.

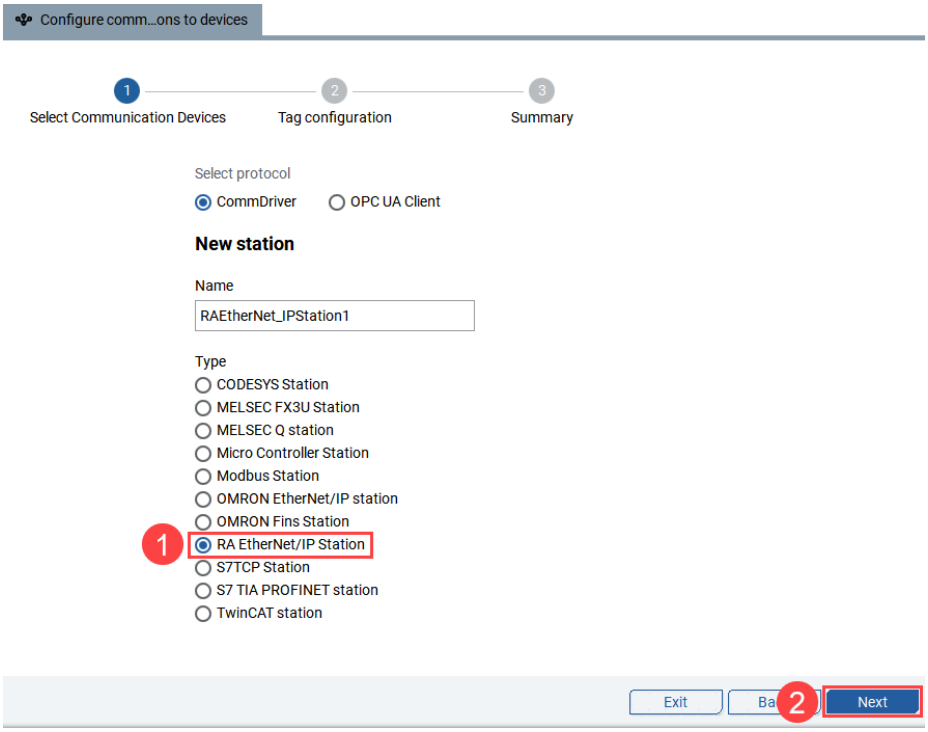


2. Select **New Station**.



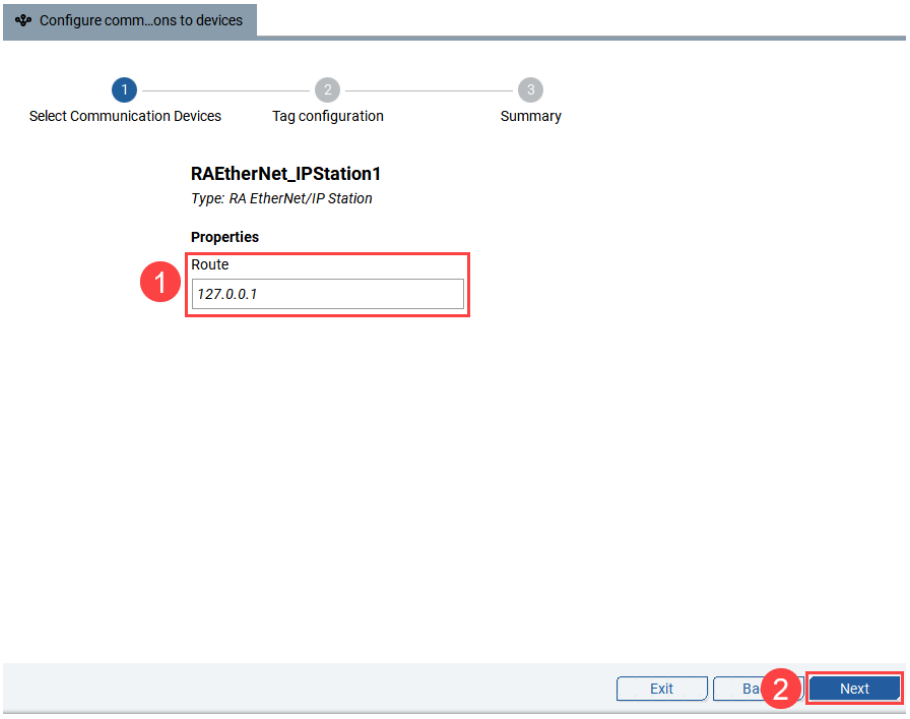
© 2025 Rockwell Automation, Inc. All rights reserved.

3. Now pick the radio button for **RA EtherNet/IP station**. Leave the **Name** at default and click on **Next**.



Note: Notice the list of native drivers that can be added to FactoryTalk Optix as well as communicating via OPC UA.

4. Enter the IP address or path in the **Route** field, and then click **Next**.

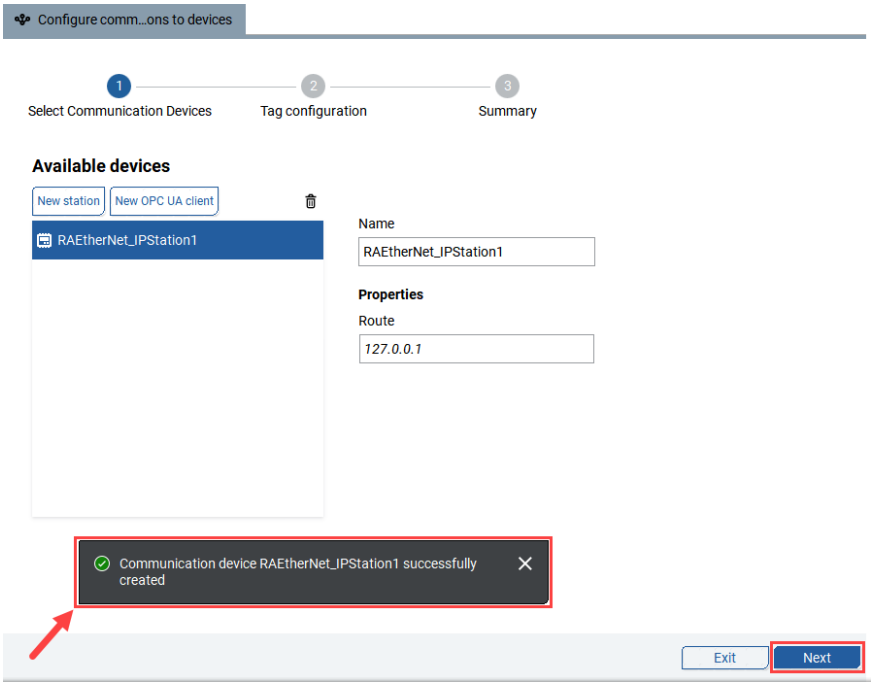


© 2025 Rockwell Automation, Inc. All rights reserved.

ROUTE SYNTAX

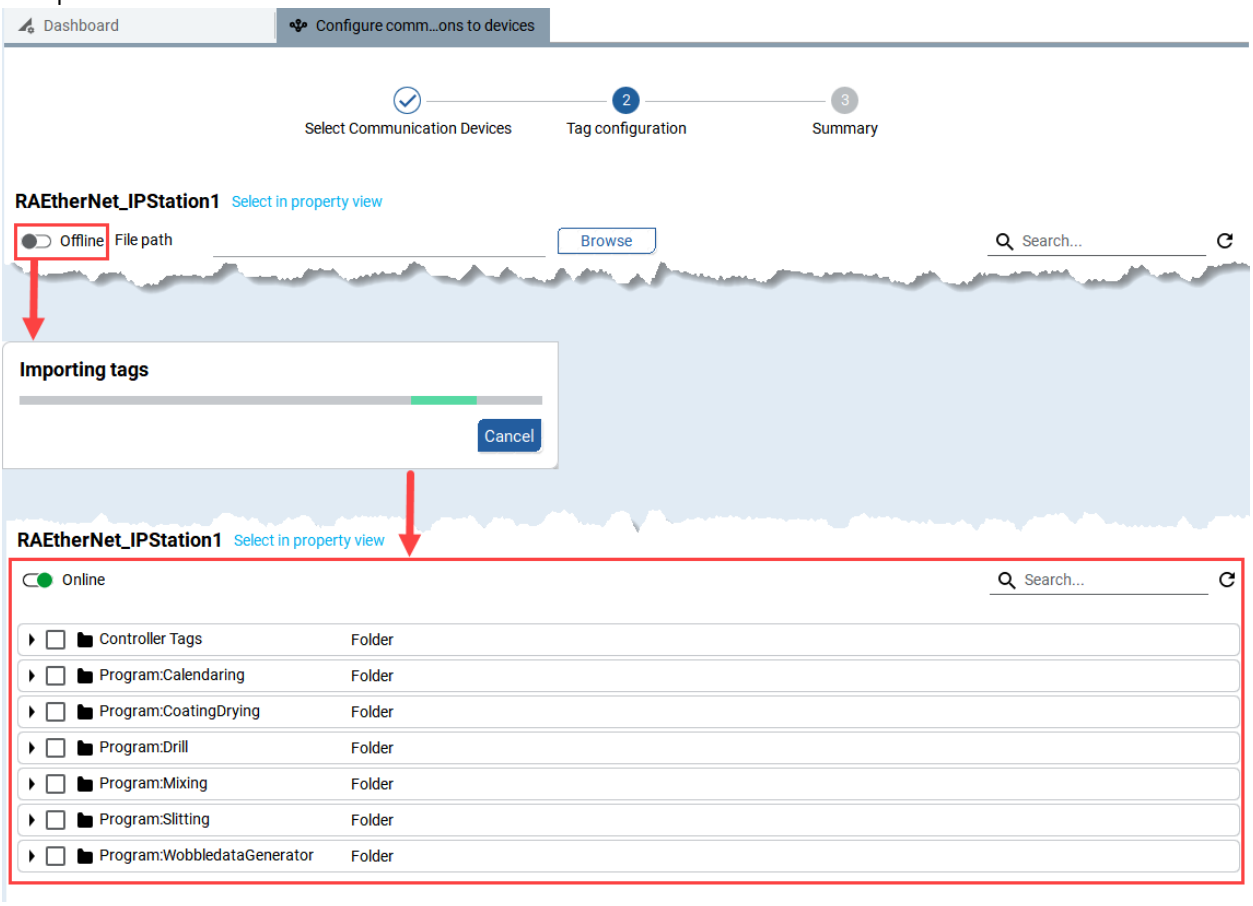
Using the IP address, the driver is looking at the controller in slot 0.
If you want to use a controller in a different slot you need the following syntax:
IP address\Backplane\slot, for example:
192.168.1.20\Backplane\2 would be the controller in slot 2

5. The RA Ethernet/IP station has been successfully created. Choose **Next** to continue to **Tag configuration**.



© 2025 Rockwell Automation, Inc. All rights reserved.

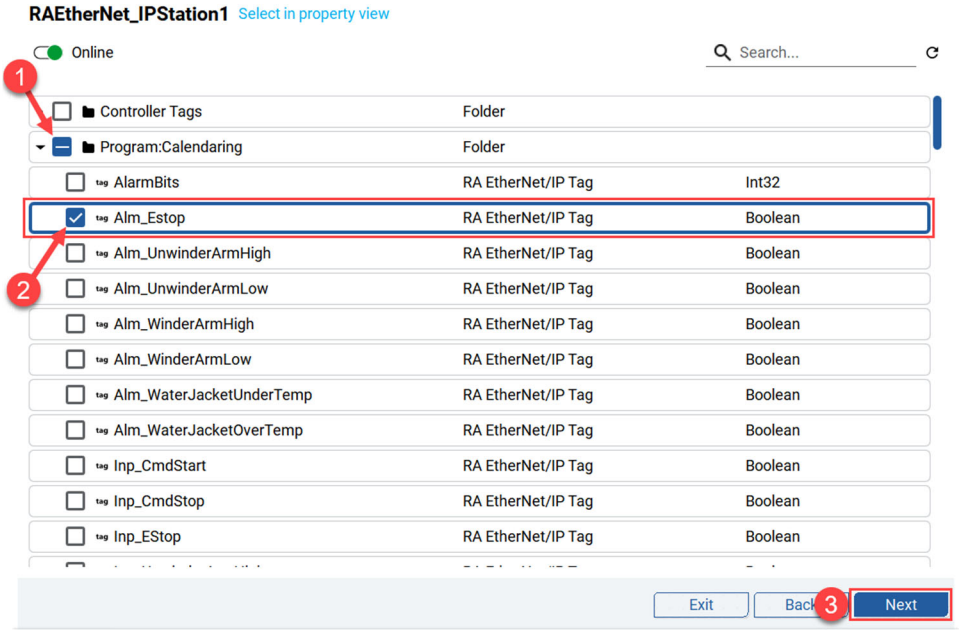
6. Tags can be imported offline (from an ACD file) or Online (directly from the controller). Toggle the switch to the desired setting to execute an **Online** or **Offline** tag import. Wait for tags to import to complete.



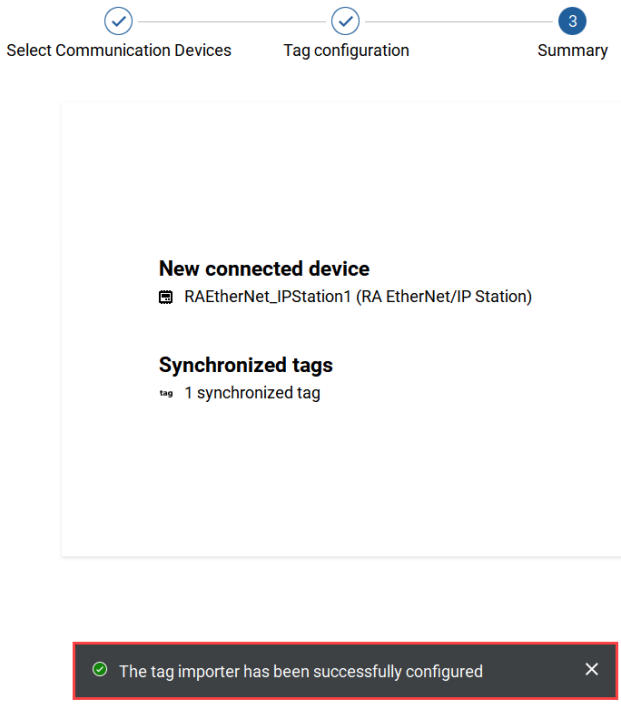
Note: If you don't have a controller, you can develop in Offline mode through an L5K, L5X or ACD file.

© 2025 Rockwell Automation, Inc. All rights reserved.

7. Now, expand **Program:Calendaring** (Your program could have a different name) and check the box next to **Alm_Estop**. Then choose **Next** to add the tag to the application.

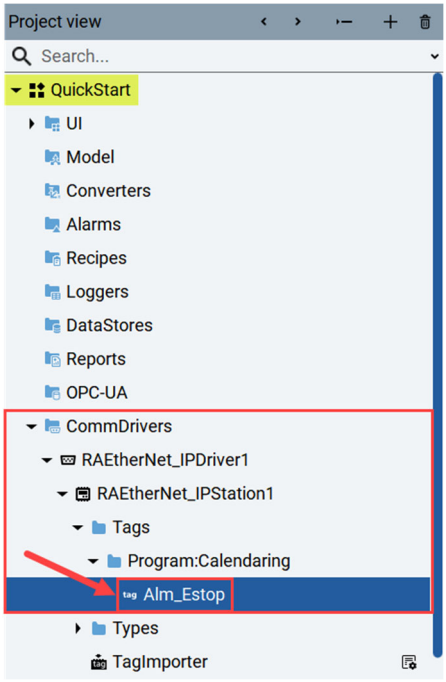


8. After seeing the Summary tab, click on **Exit** to close the communications wizard.

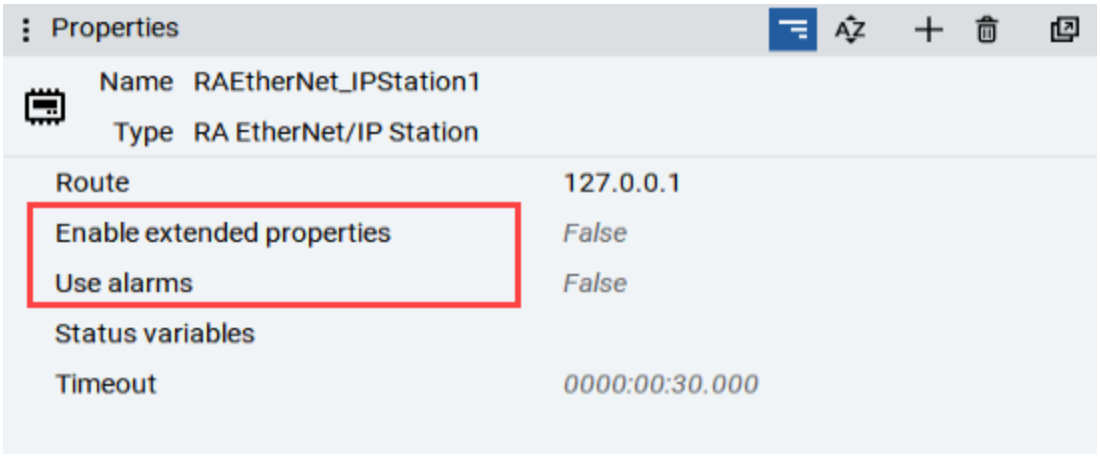


© 2025 Rockwell Automation, Inc. All rights reserved.

9. Now, in the Project view under **QuickStart** > **CommDrivers** > **RAEtherNet_IPDriver1** > **RAEtherNet_IPStation1** > **Tags** > **Program:Calendaring** you will find the communication driver and station created plus the tag you imported from the Logix controller into the project, **Alm_Estop**.



Note: When enabled, FactoryTalk Optix preferred connectivity to Logix-based controllers include access to Logix tag extended properties as well as Logix-tag based and instruction-based alarms. You access these settings from the Station Properties.



© 2025 Rockwell Automation, Inc. All rights reserved.

Appendix B – Container Objects

CONTAINER OBJECTS

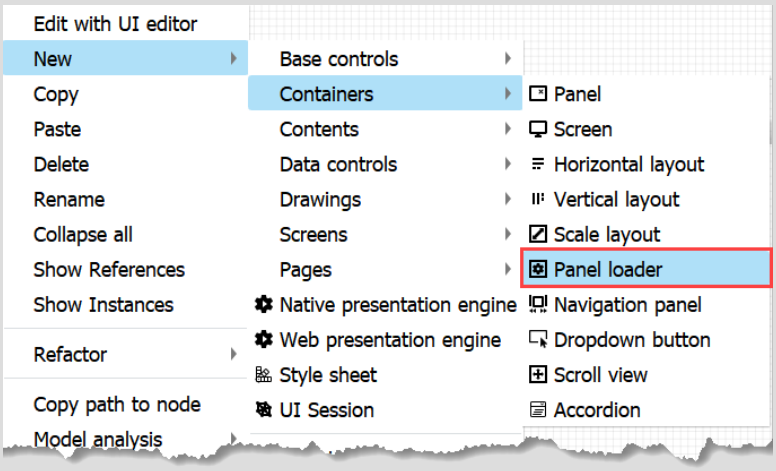
Container objects include the following:

- **Panel** - Simple container of graphic objects within a selectable area that constitutes a specific widget. By default, each new panel has a fixed width and height and is aligned with the top-left corner of the parent object.
- **Screen** – A Panel that constitute a specific interface. By default, each new screen stretches to cover the entire parent object and has a background color.
- **Horizontal layout** – A container that automatically arranges graphic objects horizontally next to each other.
- **Vertical layout** - A container that automatically arranges graphic objects vertically next to each other.
- **Scale layout** - Container that automatically scales its children graphic objects when the container aspect ratio changes.
- **Grid layout** - Container that arranges objects in a grid.
- **Panel loader** - Container that uses logic to display different alternative panels at runtime.
- **Navigation panel** - Panel that contains other panels and automatically organizes panels into tabs that you can navigate.
- **Dropdown button** - Button that, when selected, opens and closes a pop-up window.
- **Scroll view** - Panel that can scroll to display content larger than the panel size.
- **Accordion** - Container that contains expandable content blocks.
- **Dialog box** - A modal window that does not allow interaction with other interface components until it is closed.
- **Window** - Root container. In a Presentation Engine, you can set Window to specify the initial user interface at runtime.

© 2025 Rockwell Automation, Inc. All rights reserved.

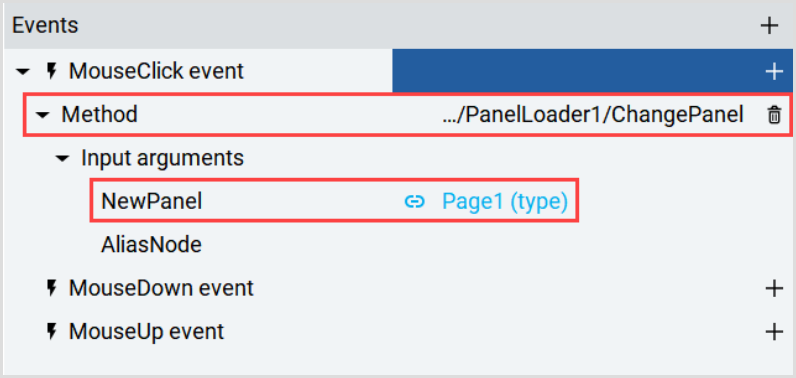
THE PANEL LOADER OBJECT

An alternative way to add navigation to get to the various panels (aka. pages, screens, etc.) is to use the **Panel loader** object. The object is found under **New** > **Containers**.



The **Panel loader** Object includes a method, **ChangePanel**, to load a panel from within your project into the **Panel loader** container at runtime.

For example, a button can trigger a **MouseClick event** to call the method. **Page1** will be loaded in this instance.



© 2025 Rockwell Automation, Inc. All rights reserved.

Appendix C – OPC UA

Please note that FactoryTalk Optix includes the ability to natively support OPC UA in addition to the following:

- MQTT Broker
- MQTT Subscriber
- MQTT Publisher
- **New!** Staring with FactoryTalk Optix version 1.5, MQTT can be supported without requiring Netlogic C# script.

This appendix will focus on OPC UA

OPC UA Connectivity

READ ABOUT OPC UA CONNECTIVITY

Use these components to develop your application logic:

- OPC UA nodes
- OPC UA server object
- OPC UA client objects

OPC UA nodes

- FactoryTalk Optix Studio enables importing nodes from the OPC UA server.

OPC UA server

- Publishes the nodes of the project information model at runtime. The server publishes all project nodes by default. You can select which nodes to publish to the assigned users.
- Important: You can configure a single OPC UA server object in a project.
- The server object enables an OPC UA client to:
  - Read from the server (listen to events)
  - Write to the server (invoke methods)

OPC UA client

- The OPC UA client object enables communication with the OPC UA server.
- Tip: You can configure unlimited OPC UA client objects in a project.
- The client accesses nodes published by the server to:
  - Read from the server (listen to events)
  - Write to the server (invoke methods)
- You can specify the nodes to the server to import at design time or at runtime.
  - For example, an OPC UA client can read specific object variables or listen to events generated by alarms.

© 2025 Rockwell Automation, Inc. All rights reserved.

OPC UA Endpoints

OPC UA ENDPOINT URL

In FactoryTalk Optix, the predefined endpoint URL is **opc.tcp://localhost:59100**
The URL of an endpoint is composed of these elements:

- Communications protocol: opc.tcp
- Network address, expressed in one of the following modes:
  - Localhost, a keyword that expresses local address.
  - Hostname, the name of the network device on which the server is hosted.
  - IP address, the IP address of the network device on which the server is hosted.

Tip: Unless there are specific requirements, Rockwell Automation recommends using the localhost keyword.

- Port
  - In this lab we are using **opc.tcp://localhost:48010**, which is the default endpoint URL for the Unified Automation CPP OPC UA Demo Server being utilized in this lab.

OPC UA Information Model

READ ABOUT THE INFORMATION MODEL OBJECT

In FactoryTalk Optix, the information model appears in the form of Objects containing members of various data types and functions. Objects are similar to User-defined Data Types (UDTs) found in Studio 5000 Logix Designer.

Using an OPC UA information model enables data to be contextualized according to OPC UA Companion Specifications such as PackML M2M communications and ISA-95 Common Object Model, or a custom data model. The OPC UA standard is the backbone of FactoryTalk Optix software, which is built entirely upon OPC UA specifications.

Create custom types linking data accessed from Rockwell Automation controllers, third-party devices, and OPC UA servers, with different names and layouts, to reorganize and unify the data to present a consistent form for an OPC UA Client.

The information model feature enables standard or custom structures and items:

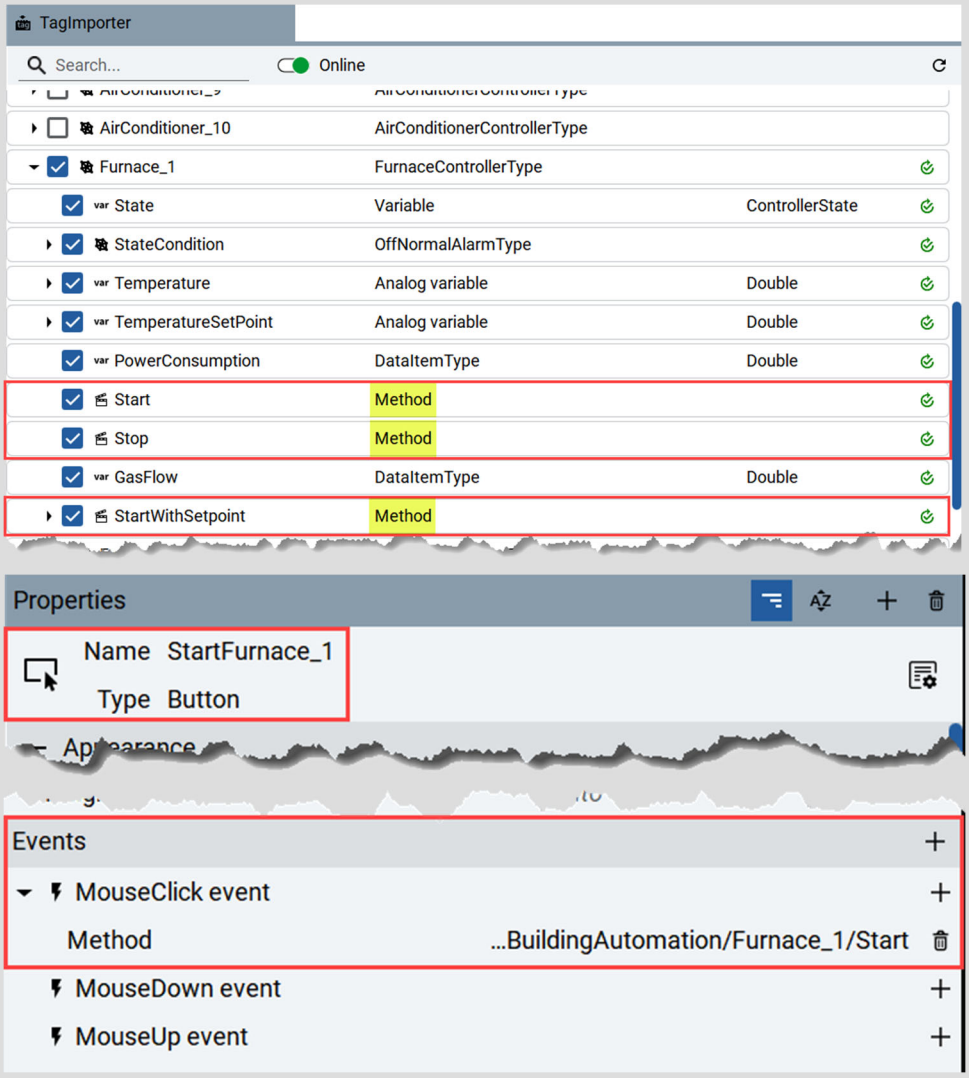
- Organize scalar data into structures.
- Unify/standardize data from multiple sources or programmable controllers into a structure.
- A machine built for one industry may be used in a different one by re-arranging the data model to support a different convention or standard. Therefore, an industry specific standard may be met without having to redefine and program the programmable controllers.
- Improve the organization of the automation system data. For example, the layout of structures coming from programmable controllers may be changed by mapping Logix data with different names.
- Limit or secure the automation system data that is available to clients.
- Contain OPC UA methods, which are the function calls of an Object.

© 2025 Rockwell Automation, Inc. All rights reserved.

OPC UA Methods

CALL AN OPC UA METHOD

- OPC UA methods are the function calls of an Object.
  - Methods may be called, and they will return their status either successful or unsuccessful.
  - The time required to execute a function will vary depending on complexity.
  - For instance, buttons could be created to toggle the **Start** and **Stop** methods of an instance of a furnace in an OPC UA Server.



© 2025 Rockwell Automation, Inc. All rights reserved.

CREATE A CUSTOM OPC UA METHOD

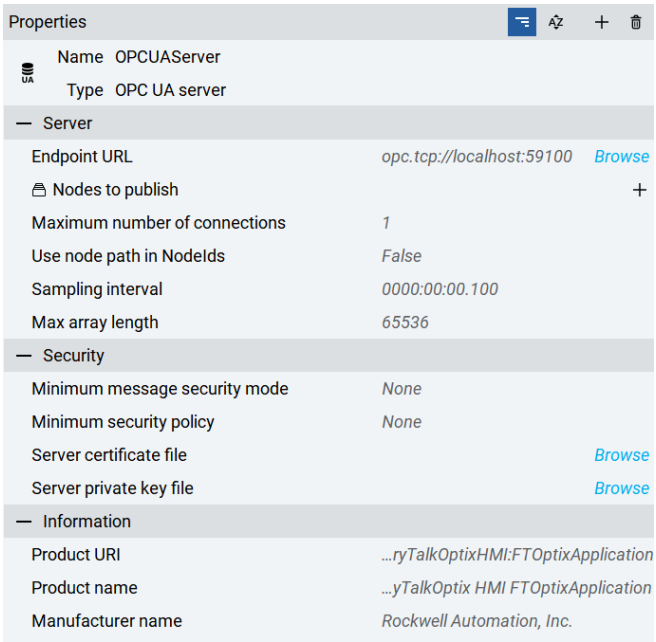
- In FactoryTalk Optix, you can create an OPC UA method callable by an external OPC UA client and then edit its code in an external code editor.
- Example of an OPC UA script structure:

```
[CustomBehavior]
public class testTypeBehavior : BaseNetBehavior
{
    public override void Start()
    {
        // Insert code to be executed when the user-defined
behavior is started
    }
    public override void Stop()
    {
        // Insert code to be executed when the user-defined
behavior is stopped
    }
#region Auto-generated code, do not edit!
    protected new testType Node => (testType)base.Node;
#endregion
}
```

Configure Communications: OPC UA Server

1. In Project view, right-click OPC UA and select **New** > **OPC UA Server**.

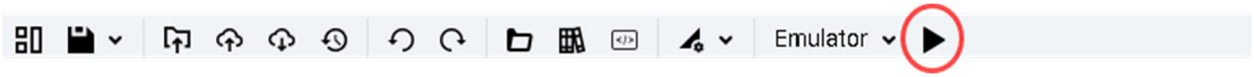
24. Looking in Properties, the default URL is opc.tcp://localhost:59100.



25. Save the project.

© 2025 Rockwell Automation, Inc. All rights reserved.

26. From the main toolbar, with **Emulator** selected, click the Run Emulator icon ▶.



Note: Running the Emulator starts the Optix OPC UA server that the OPC UA test client will connect to.

Minimize the emulator (Do not close)

Using the UaExpert OPC UA test client

1. From the **Start Menu**, select the **UaExpert** OPC UA test client.

27. In **UaExpert**

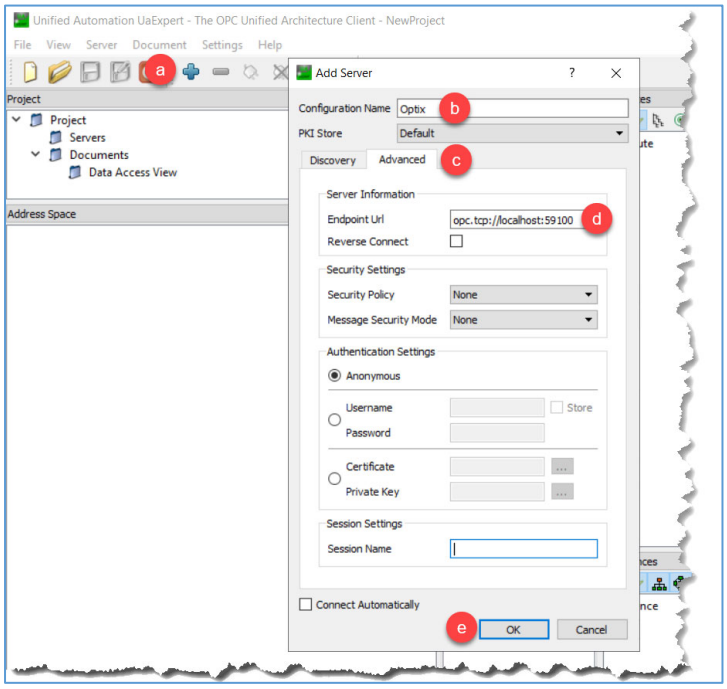
   a. Click the plus icon ➕ to add a new server

   b. Enter **Optix** for the **Configuration Name**

   c. Select the **Advanced** tab

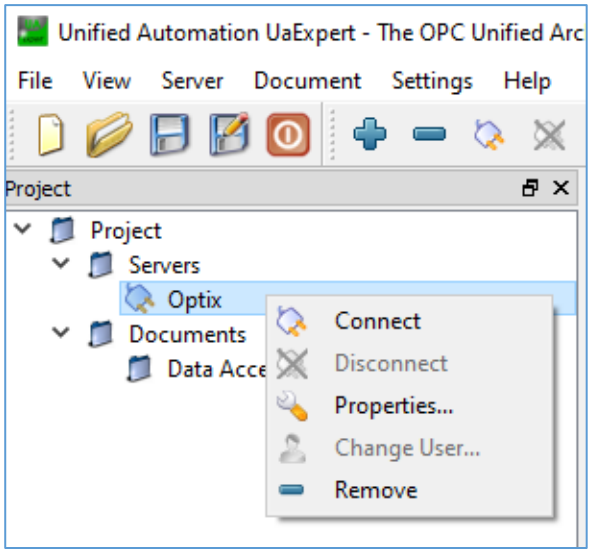
   d. Enter the following **Endpoint Url**: "opc.tcp://localhost:59100"

e. Select **OK**.

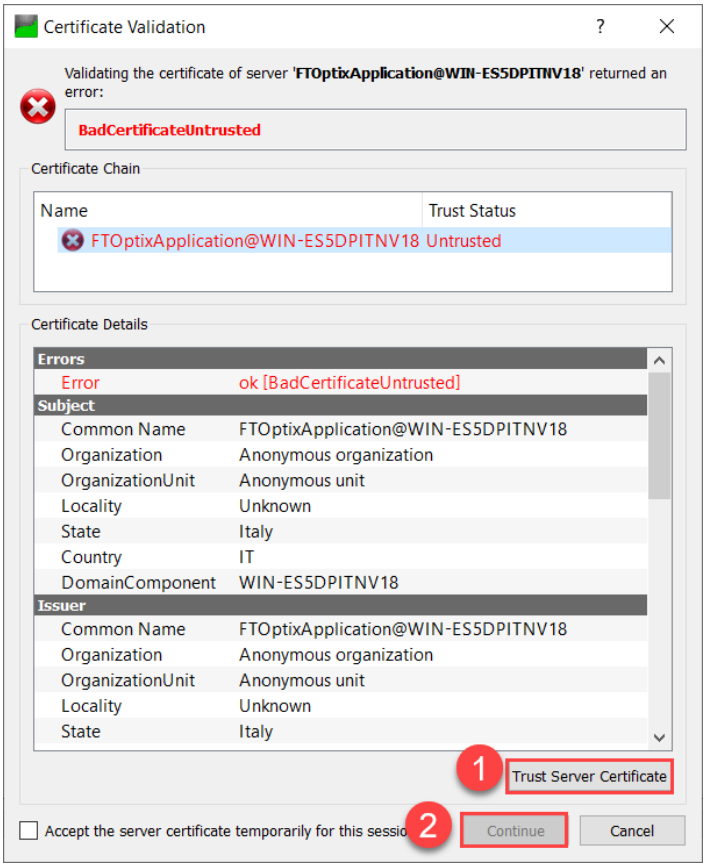


© 2025 Rockwell Automation, Inc. All rights reserved.

28. In the **Project** pane, right-click on **Optix** and select **Connect**.

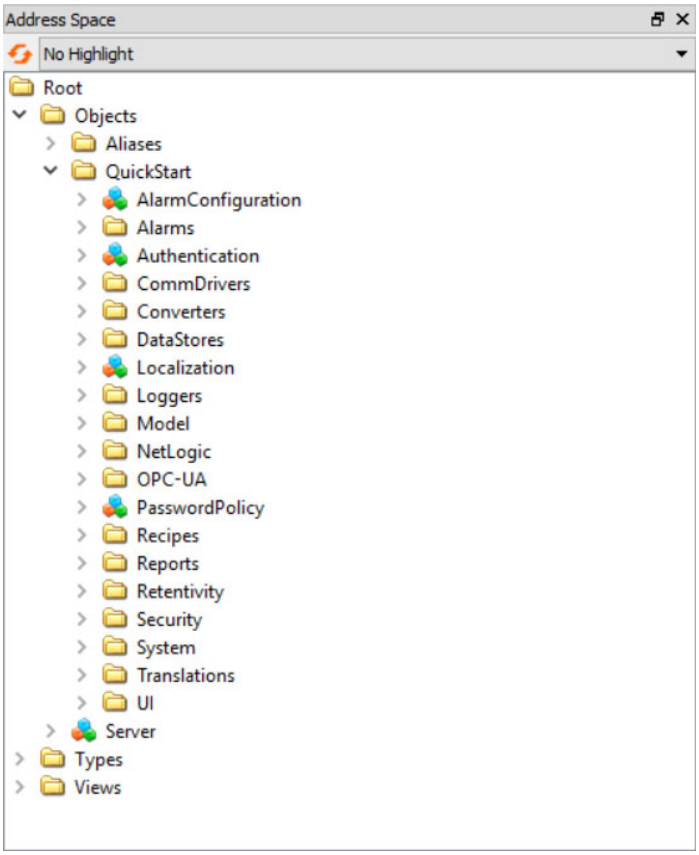


If you are prompted about a certificate issue, for the purpose of this lab, choose **Trust Server Certificate** then **Continue** to connect.

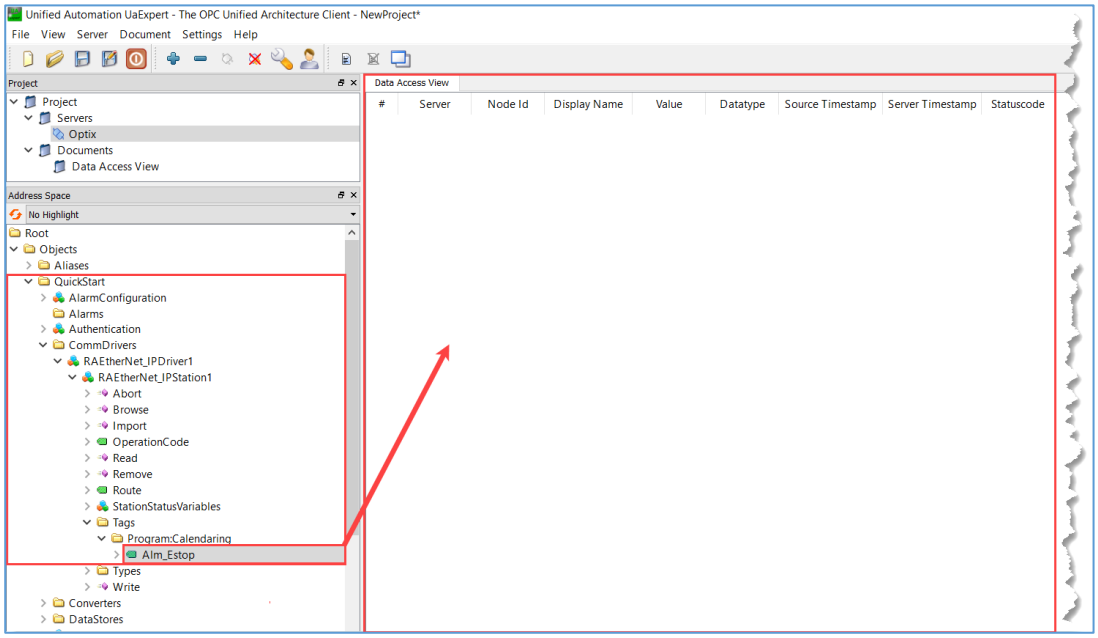


© 2025 Rockwell Automation, Inc. All rights reserved.

29. Using the **Address Space** pane, expand the FactoryTalk Optix application name we are running, **QuickStart**, and note how the application is functioning as a full OPC UA Server.

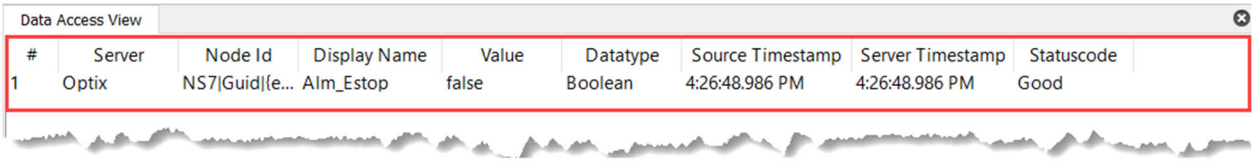


30. Expand **QuickStart** > **CommDrivers** > **RAEtherNet_IPDriver1** > **RAEtherNet_IPStation1** > **Tags** > **Program:Calendaring**, drag and drop the **Alm_Estop** to the center pane.



© 2025 Rockwell Automation, Inc. All rights reserved.

31. The Logix controller variable, **Alm_Estop**, is now being monitored in the center pane **Data Access View** tab.



32. Feel free to experiment by adding other variables
33. Switch back to the FactoryTalk Optix **Emulator** and change the values displayed and make sure you can see those values update in the **UaExpert** client.
34. Close the **UaExpert** client
35. Close the **Emulator**.

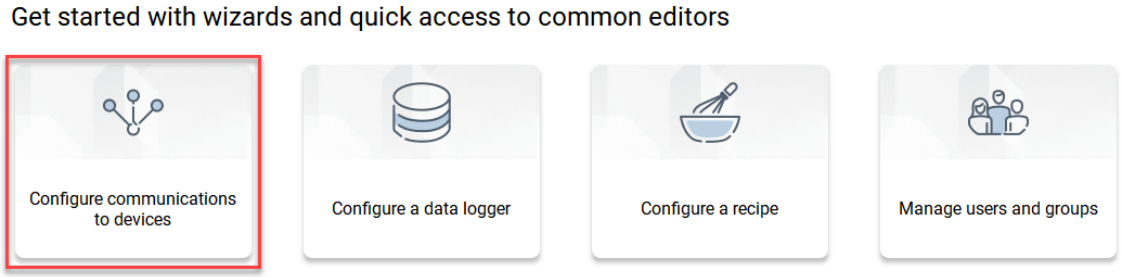
Configure Communications: OPC UA

Besides the communication stations/drivers that can be added to FactoryTalk Optix, the software has full OPC UA connectivity. FactoryTalk Optix is compliant with the OPC UA (OPC Unified Architecture) standard and can communicate with any OPC UA client or server.

Note: Devices running properly configured FactoryTalk Optix applications can operate as an OPC UA client or server. For more information on OPC UA Connectivity, see Appendix D.

OPC UA Client Configuration

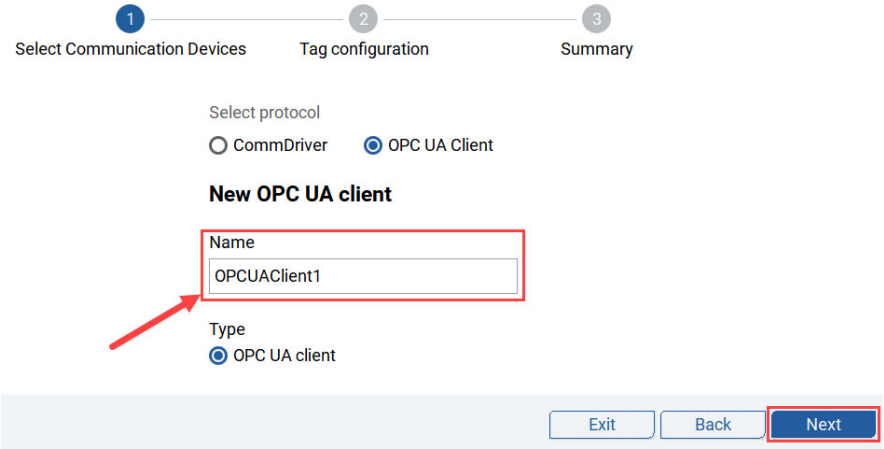
1. Switch to dashboard and click on the **Configure communications to devices** icon.



© 2025 Rockwell Automation, Inc. All rights reserved.

2. Under **Available Devices** choose **New OPC UA client**.

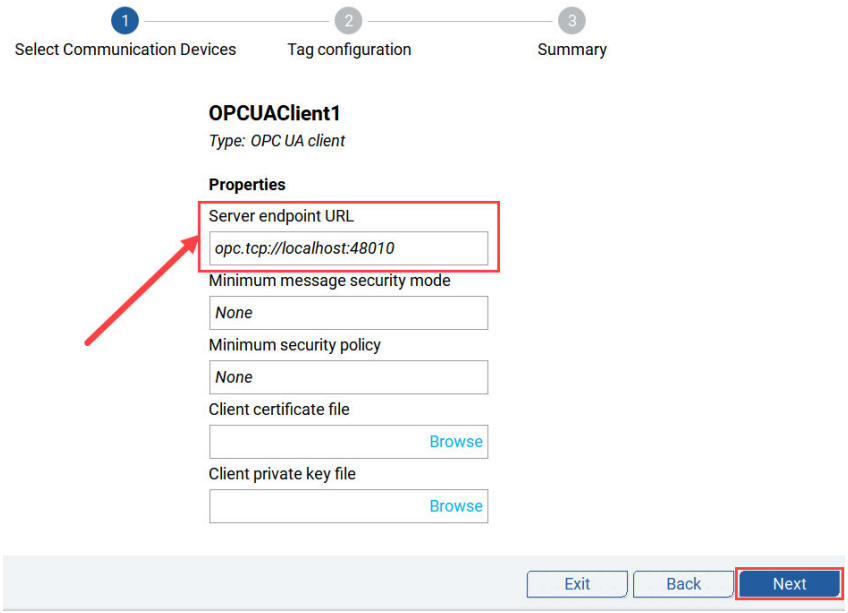
3. Keep the **Name** default as **OPCUAClient1** and click on **Next.**



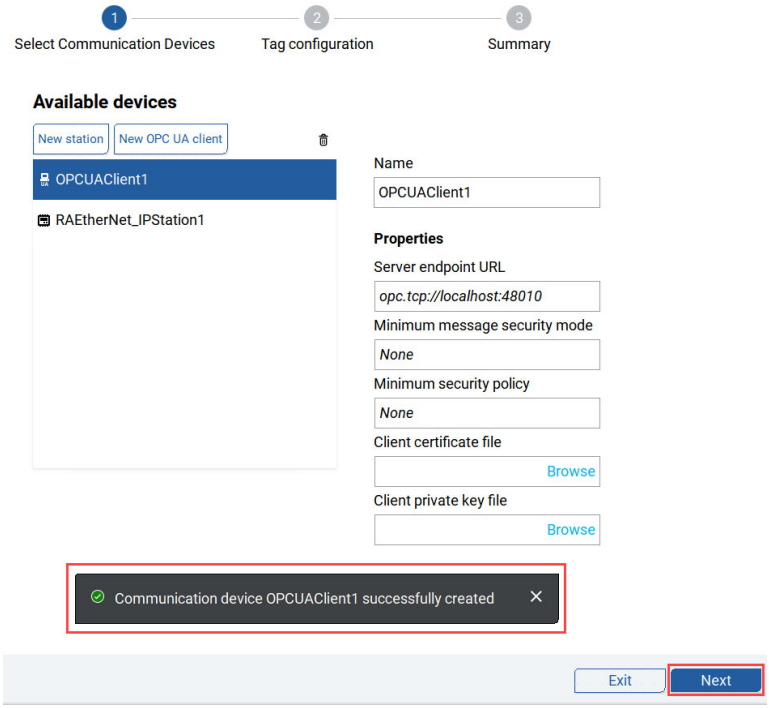
4. Under the **Server Endpoint URL** you need to fill in "opc.tcp://localhost:48010". Leave the rest of settings at default. Then click on **Next**.

Note: In this case, we will not configure security to avoid additional steps required to manage certificates.

© 2025 Rockwell Automation, Inc. All rights reserved.

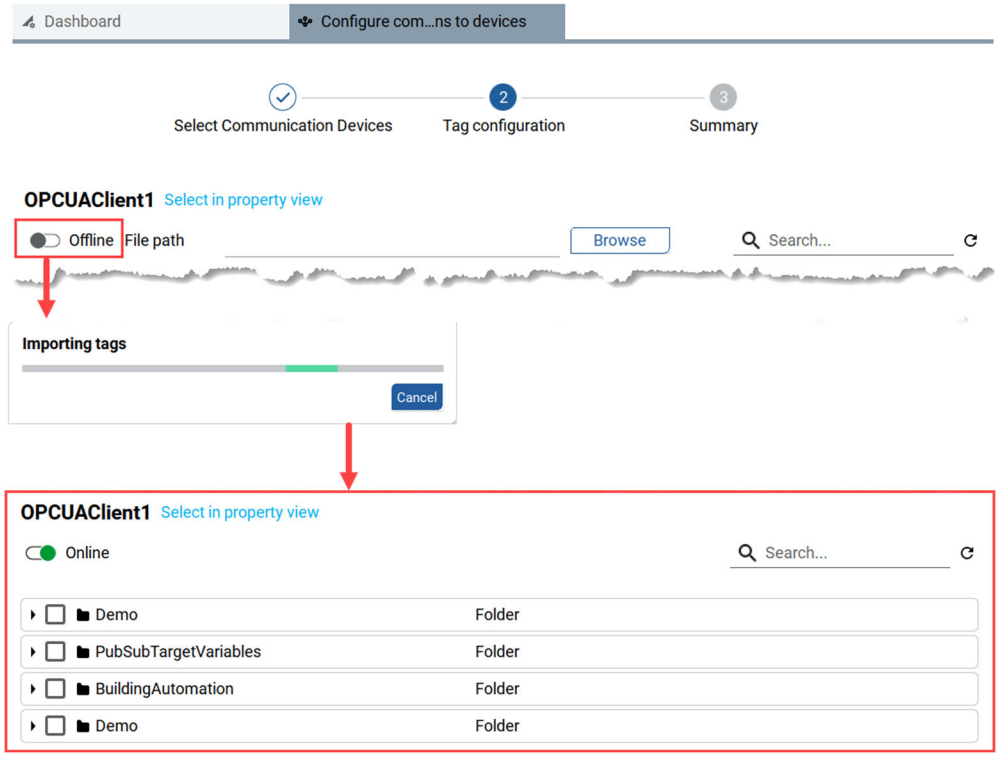


5. You may see the message "**Communication device OPCUAClient1 successfully created**". Optional: take a moment to explore the properties of the two **Available Devices** configured, **OPCUAClient1** and **RAEtherNet_IPStation1.**



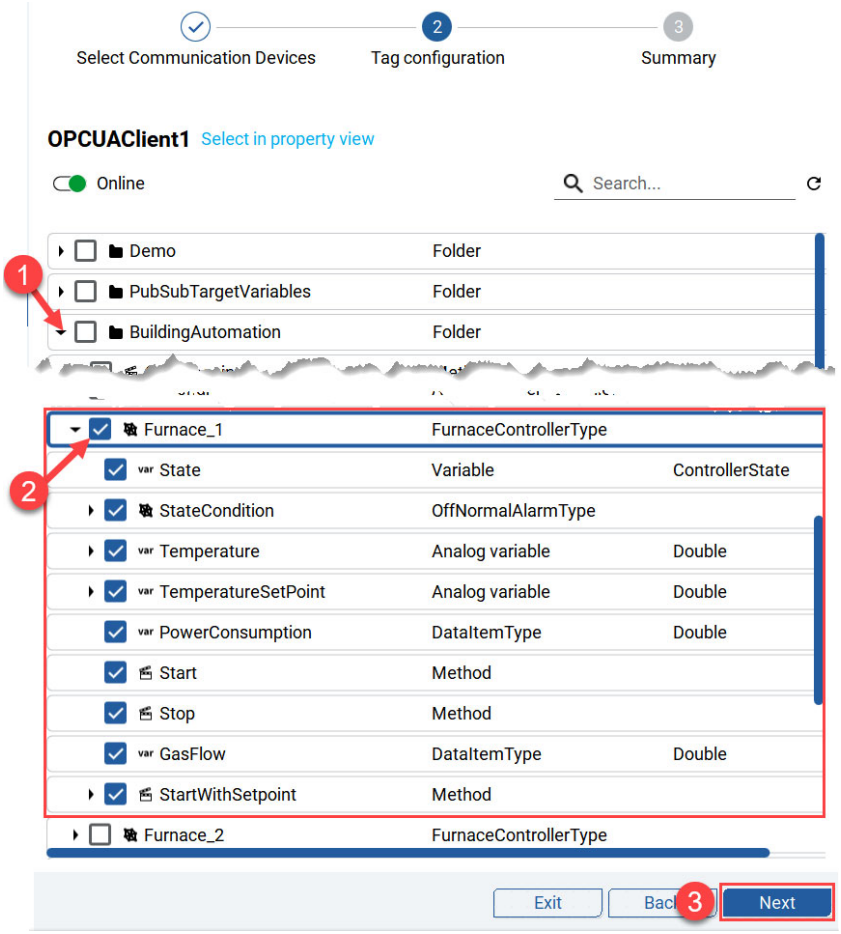
© 2025 Rockwell Automation, Inc. All rights reserved.

6. Toggle the switch next to **Offline** to execute an **Online** tag import. Wait for tags to import from the OPC UA Server.



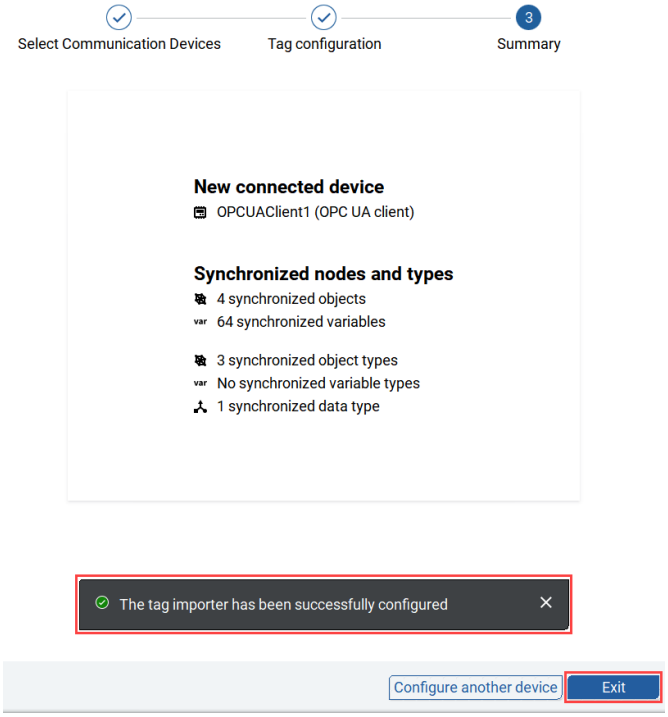
© 2025 Rockwell Automation, Inc. All rights reserved.

7. Now, expand **Building Automation** and check the box next to **Furnace_1** to import the OPC UA information model Object and all its members. Then choose **Next** to add the object to the application.



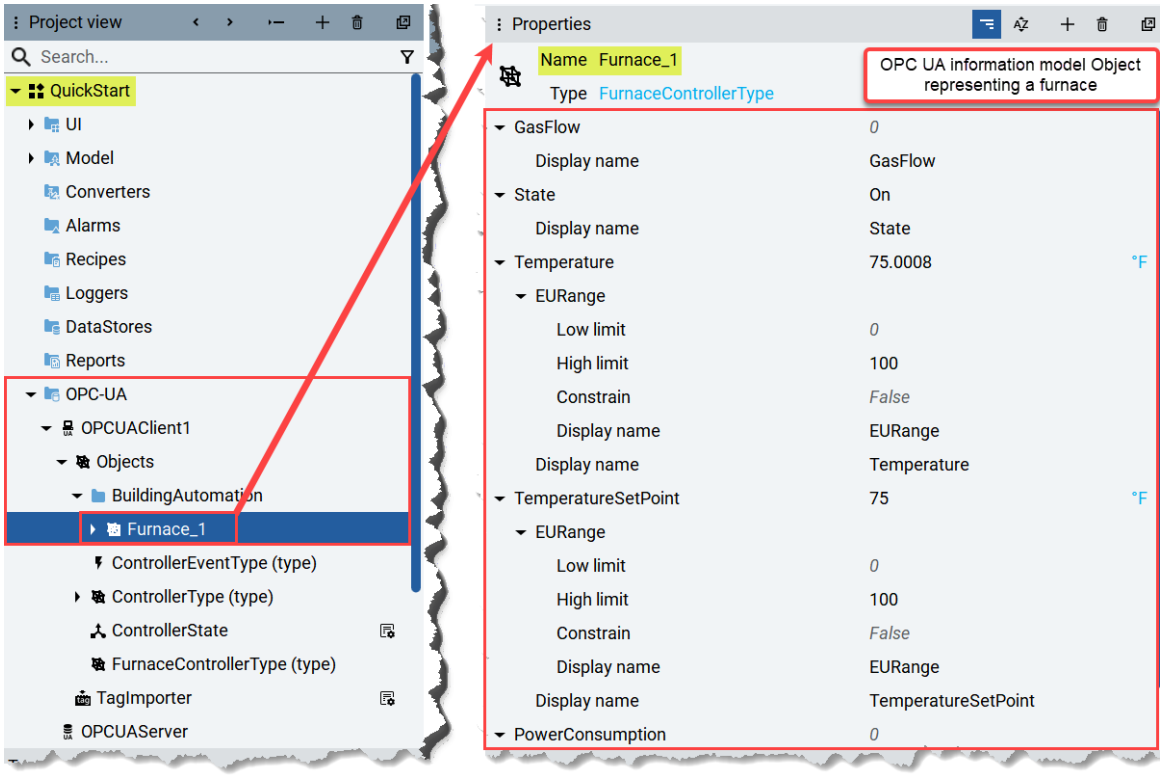
© 2025 Rockwell Automation, Inc. All rights reserved.

8. Click on **Exit** to close the communications wizard.



© 2025 Rockwell Automation, Inc. All rights reserved.

9. Expand **OPC-UA** > **OPCUAClient1** > **Objects** > **Building Automation** and select **Furnace_1** which is the Object you imported from the OPC UA Server into the project. The variable members of the Object will be found under the **Properties** of **Furnace_1**.



Note: Some of the members of Furnace_1 will be used in the next section.

SUMMARY

So far, we have created a new project, established communications to both a Logix controller and an OPC UA Server and saw how to import tags into the FactoryTalk Optix Studio project.

Key takeaways:

- Using the built-in wizards, it is easy to set up communications quickly
- Tags/variables can be imported either online or offline
- Besides Rockwell Automation controller support there are many native driver options for connectivity to third-party devices
- FactoryTalk Optix Studio is built on the OPC UA specification
    - An application can be configured as an OPC UA Client, Server, or both.
    - Companion specifications and custom information models are supported
    - OPC UA information models are:
        - In the form of Objects containing members of various data types and function calls called methods
        - Representative of an object-oriented development methodology, which can be applied to create reusable types for a variety of applications.
        - Applied to provide consistent context for data from various sources

© 2025 Rockwell Automation, Inc. All rights reserved.

Appendix D – Responsive Graphics

Responsiveness - Scale Layout

Develop a UI with responsive graphics using the **Scale layout, Horizontal layout,** and **Vertical layout** objects.

- **Scale layout** - container that automatically scales its children graphic objects when the container aspect ratio changes.
- **Horizontal layout** - container that automatically arranges graphic objects horizontally next to each other
- **Vertical layout** - container that automatically arranges graphic objects vertically next to each other
- **Grid layout** - Container that arranges objects in a grid (**New!** starting with FactoryTalk Optix version 1.4)

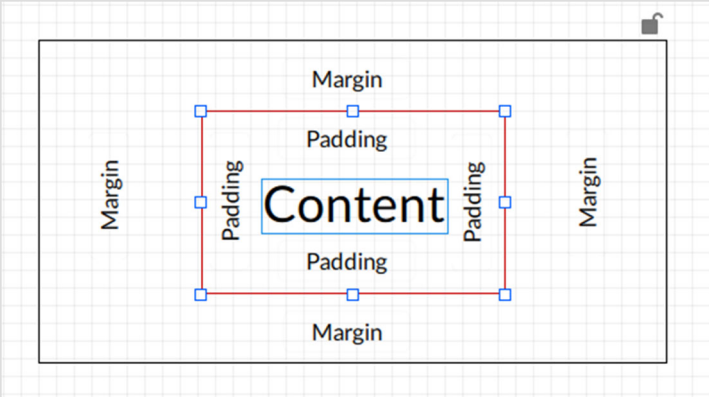
© 2025 Rockwell Automation, Inc. All rights reserved.

CONCEPTS OF LAYOUT

The layout of a user interface in a FactoryTalk Optix application depends on the following elements:

- Organization of the graphical objects in the information model
- Value of the positioning (alignment and margins) and sizing (width and height) properties of the graphical objects
- Order of the graphical objects on the z axis
- Size and aspect ratio of the screen on which the interface is displayed.
- Margins, padding and borders

Below is an example of **Text box** object to present the meaning of the terms content, margin, padding and border:



- **Content** - content text
- **Padding** - space between the content and the edge of the object. Both horizontal and vertical padding are values relative to the height of the text contained (for example, if the padding is 60%, a label with text 20 pixels high has padding of 12 pixels).
  - The padding is present only in some objects such as the **Text box**, **Button**, **Spin box**, **Data grid**, **Drop-down** and **List selection** objects.
- **Edge** - edge of the object, in the example in red
- **Margin** - space between the edge of the object and the edge of the parent object (container)
  - The padding, edge thickness and edge corner radius, if any, are global style properties defined in a **Style sheet**.

© 2025 Rockwell Automation, Inc. All rights reserved.

CONCEPTS OF LAYOUT (CONT.)

Positioning of the objects

- The positioning of a graphical object is determined by its alignment on the x and y axes and by its margins with respect to the sides of the container object. The margins to be set depend on the alignment set, or the value of the **Horizontal alignment** and **Vertical alignment** properties.
- The position of objects on the x and y axes in the layout is always relative to the position of the container. If a container is moved, then all the child objects are also moved.
- The position of objects on the z axis compared to other child objects in the same container is determined by their position in the information model. Objects further down in the information model are higher up on the z axis.
- For a description of all properties related to the positioning, refer to the properties of graphical objects in the Reference on objects and variables section of this manual.

Size of the objects

- The size of a graphical object is defined via the **Width** and **Height** properties, only if the alignment on the x and/or y axes is left, right or center.
- If the value(s) of the **Width** and/or **Height** properties is/are deleted, the two dimensions take on the **Auto** value. In this way, for some objects the size is determined by the intrinsic content (for example the text in a **Text box** object) or by other contained nodes (for example graphical objects in a **Panel** object). For example, the height of a **Text box** object with **Height** = **Auto** equals the height of the text, plus any padding set in the style sheet.
- If the alignment on the x and/or y axes is set to **Adjust**, the width and/or height assume the dimensions of the container, minus any margins. This setting makes the size of the object dynamic relative to the size of the container.
- It is not possible to scale in percentage the size of an object inside a container with respect to the size of the container itself.

Create Template Object to Scale

1. In Project view, right-click **UI** and select **New** > **Folder**.
2. Mouse-over **Folder1**, select Edit , and enter “Templates” as the new name.
3. Right-click **Templates** and select **New** > **Containers** > **Scale Layout**.

   ScaleLayout1 (type) appears under Templates.

© 2025 Rockwell Automation, Inc. All rights reserved.

4. Mouse-over **ScaleLayout1 (type)**, select the edit icon , and enter "MyObjectScaled" as the new name and hit enter.

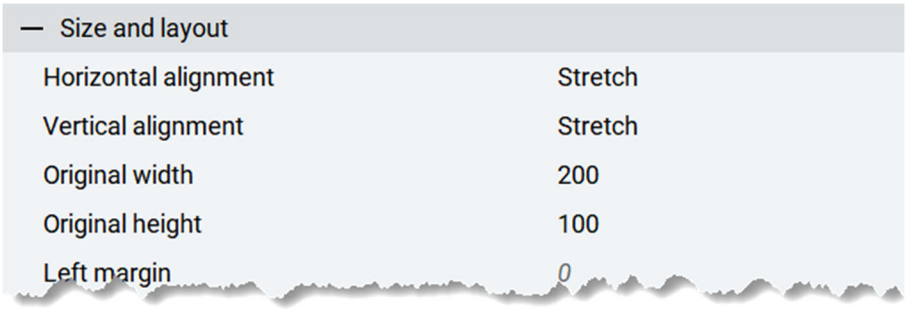
5. In **Properties**,

   a. Set **Horizontal alignment** to **Stretch.**

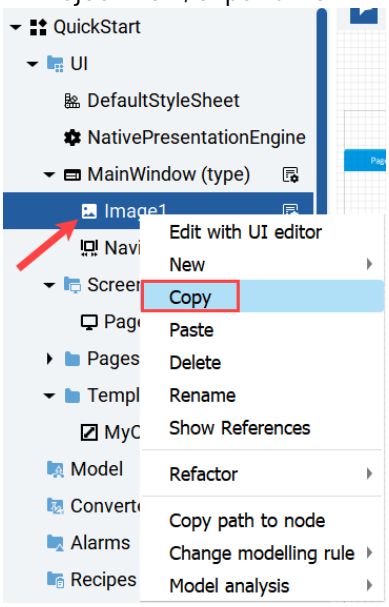
   b. Set **Vertical alignment** to **Stretch.**

   c. Set **Original width** to "200".

   d. Set **Original height** to "100".



6. In Project view, expand **MainWindow (type)**, right-click **Image1** and select **Copy**.



7. Still In Project view, right-click **MyObjectScaled** within the **Templates** folder and select **Paste.** This will add **Image1** to the **MyObjectScaled** container.

Add Template Object to Panel

1. In the Project view pane, right-click **Page12 (type)** and select **New** > **Containers** > **Vertical layout**.

   VerticalLayout1 is created under **Page12 (type).**

© 2025 Rockwell Automation, Inc. All rights reserved.

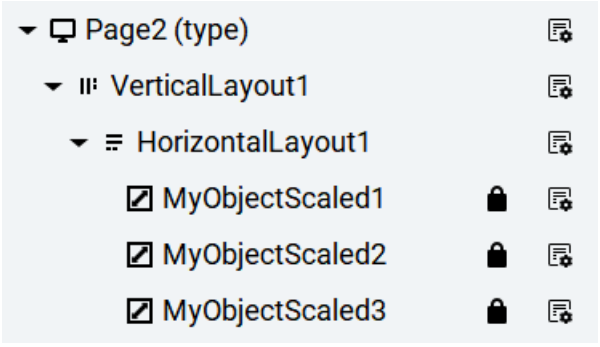
2. Select **VerticalLayout1** to change its properties.
3. In **Properties**,
   a. Set **Horizontal alignment** to **Stretch.**
   b. Set **Vertical alignment** to **Stretch.**

4. Right-click **VerticalLayout1** and select **New** > **Containers** > **Horizontal layout**.

   HorizontalLayout1 is created under **VerticalLayout1.**
5. Select **HorizontalLayout1** to change its properties.
6. In **Properties**,
   a. Set **Horizontal alignment** to **Stretch.**
   b. Set **Vertical alignment** to **Stretch.**

7. Right-click **HorizontalLayout1** and select **New** > **Templates** > **MyObjectScaled**.
8. Repeat the previous step two more times.

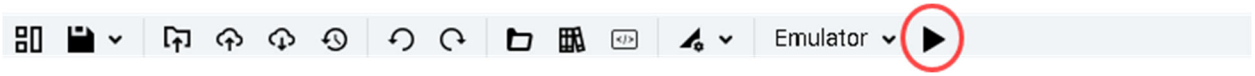
   You will have the following in the Project view:



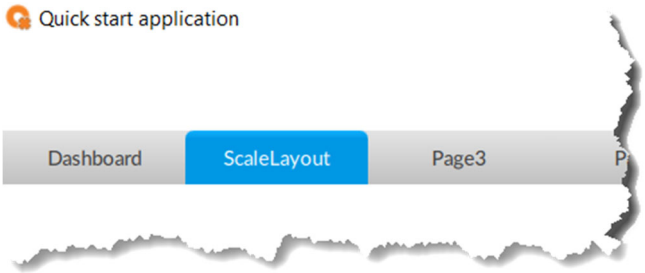
9. Mouse-over **Page12 (type)**, select the edit icon , and enter "**ScaleLayout**" as the new name and hit enter.

© 2025 Rockwell Automation, Inc. All rights reserved.

10. Save.
11. From the toolbar, with **Emulator** still displayed. Click the Run Emulator icon ▶.

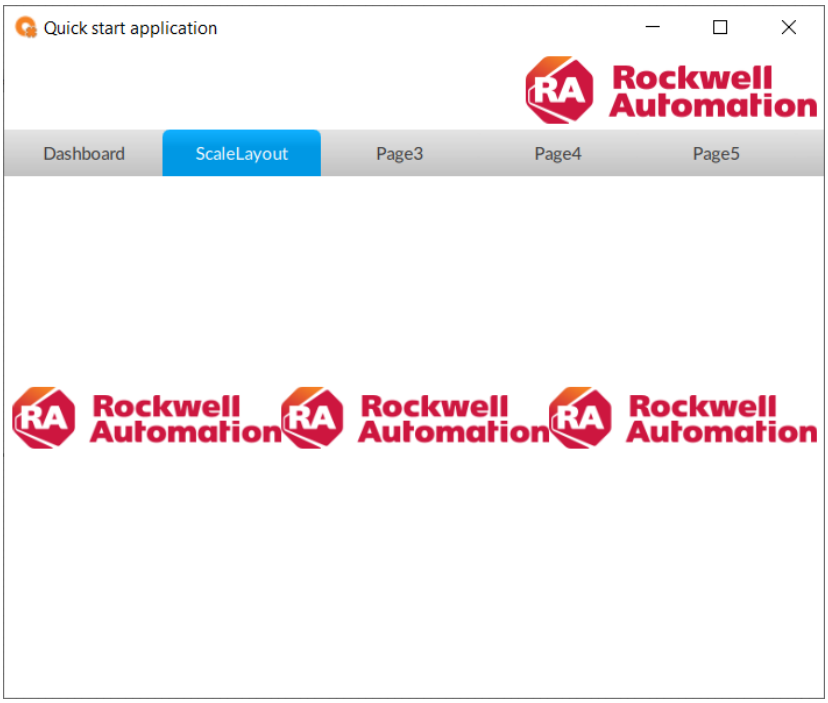


12. Select **ScaleLayout** from the navigation bar.



13. Change the size of the runtime window and watch the object maintain position and change size as needed.

© 2025 Rockwell Automation, Inc. All rights reserved.



14. Close the **Emulator**.

Responsiveness – Vertical and Horizontal Layouts

Develop a UI with responsive graphics using the **Vertical layout** and **Horizontal layout** objects.

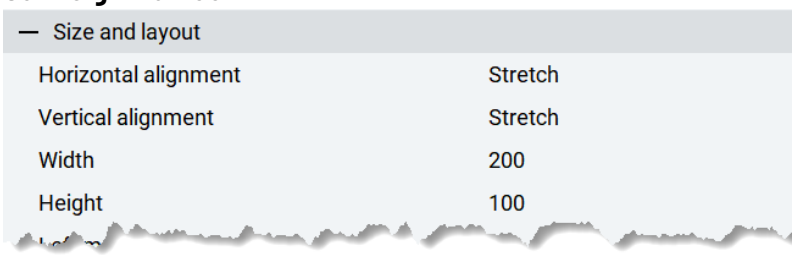
- **Horizontal layout** - panel that automatically arranges graphic objects horizontally next to each other
- **Vertical layout** - panel that automatically arranges graphic objects vertically next to each other

Create a Template Object

1. In Project view, right-click **Templates** and select **New** > **Container** > **Panel**.
2. Mouse-over **Panel1**, select Edit , and enter "MyObject" as the new name and hit enter.
3. Select **MyObject** to change its properties.
4. In **Properties**,
   a. Set **Horizontal alignment** to **Stretch**.
   b. Set **Vertical alignment** to **Stretch.**
   c. Set **Width** to "200".

© 2025 Rockwell Automation, Inc. All rights reserved.

d. Set **Height** to "100".



5. In Project view, expand, under **Templates** > **MyObjectScaled (type)**, right-click **Image1** and select **Copy**.
6. Right-click the newly created **MyObject** and select **Paste.**

   This will add **Image1** to the **MyObject** container.
7. Select **Image1** under **MyObject** to change its properties.
8. In **Properties**,

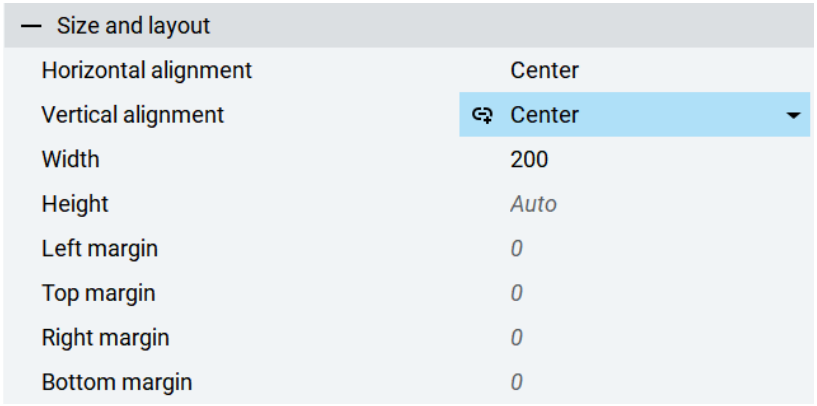
   a. Set **Horizontal alignment** to **Center**.

   b. Set **Vertical alignment** to **Center**.

   c. Set **Width** to "200".

   d. Set **Height** to **Auto**.

   e. Set all margins to "0".



Add Template Object to Panel

1. In Project view, right-click **Page13 (type)** and select **New** > **Containers** > **Vertical layout**.

   VerticalLayout1 is created under **Page13 (type).**
2. Select **VerticalLayout1** to change its properties.
3. In **Properties**,

© 2025 Rockwell Automation, Inc. All rights reserved.

a. Set **Horizontal alignment** to **Stretch.**

b. Set **Vertical alignment** to **Stretch.**

4. Right-click **VerticalLayout1** and select **New** > **Containers** > **Horizontal layout.** **HorizontalLayout1** is created under **VerticalLayout1.**

5. Select **HorizontalLayout1** to change its properties.

6. In **Properties**,

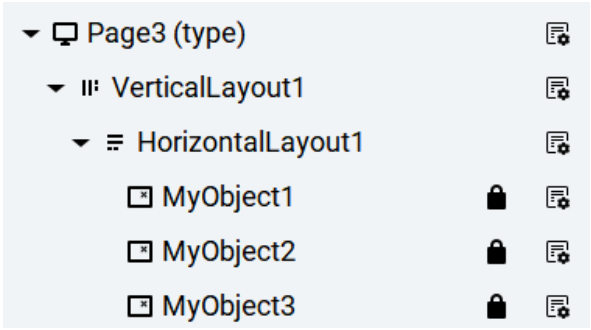
a. Set **Horizontal alignment** to **Stretch.**

b. Set **Vertical alignment** to **Stretch.**

7. Right-click **HorizontalLayout1** and select **New** > **Templates** > **MyObject**.

8. Repeat the previous step two more times.

You will have the following in the Project view:

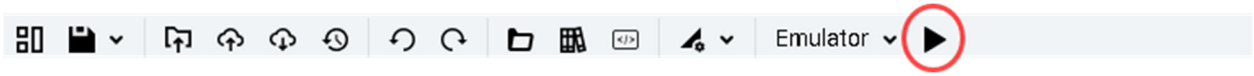


9. Mouse-over **Page13 (type)**, select the edit icon , and enter "**VH_Layouts**" as the new name and hit enter.

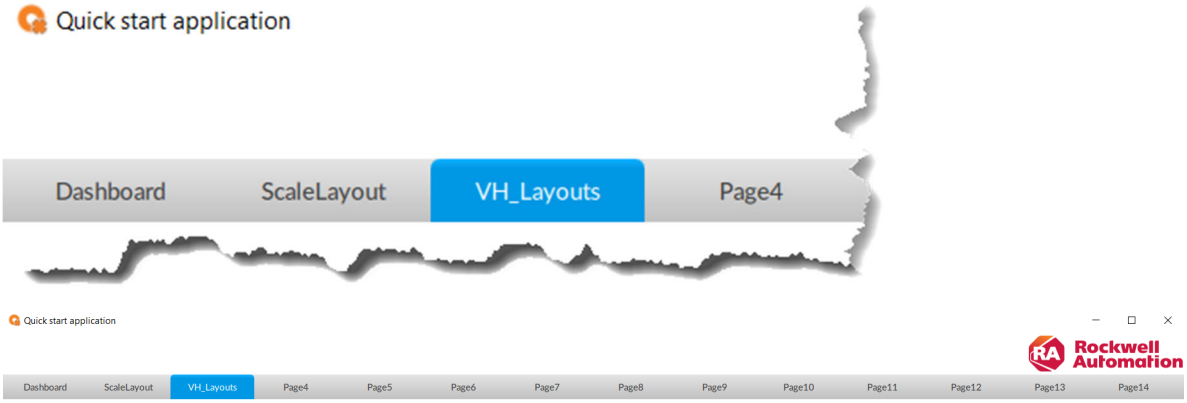
10. Save the project.

© 2025 Rockwell Automation, Inc. All rights reserved.

11. From the toolbar, with **Emulator** still displayed. Click the Run Emulator icon ▶.



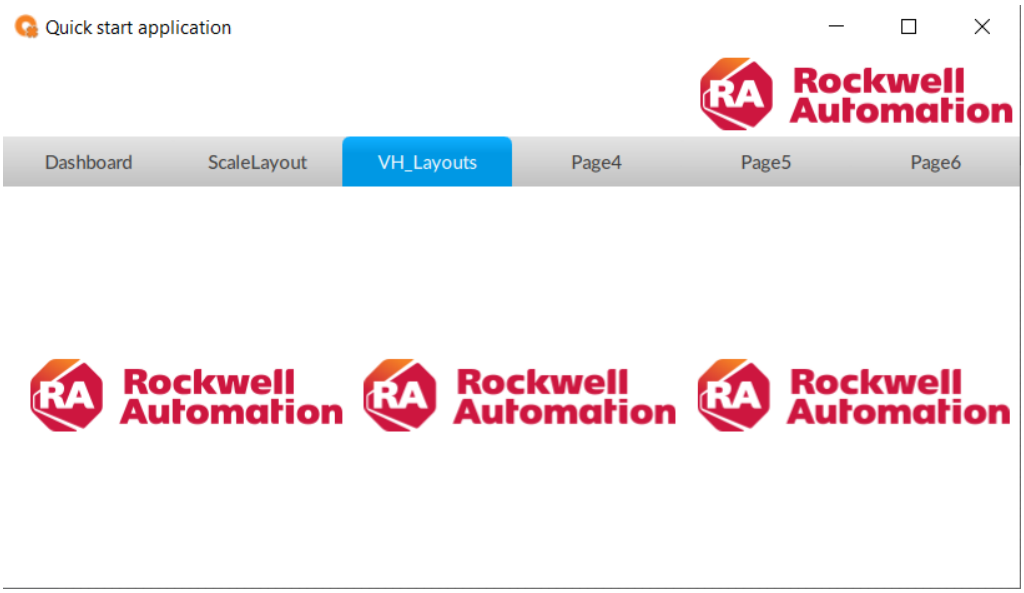
12. Select **VH_Layouts** from the navigation bar.



13. Change the size of the runtime window and watch the objects maintain relative positions.

Note: In this case, the size of the logos will be maintained because the **Scale layout** container has not been implemented.

© 2025 Rockwell Automation, Inc. All rights reserved.



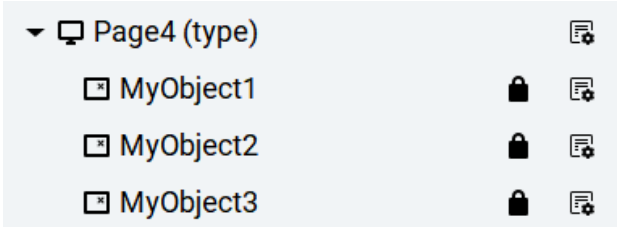
14. Close the **Emulator**.

Responsiveness – Vertical and Horizontal Alignments

Develop a UI with responsive graphics using the **Vertical alignment** and **Horizontal alignment** properties.

- **Horizontal alignment** - Alignment of the object on the horizontal axis
- **Vertical alignment** - Alignment of the object on the vertical axis

1. In Project view, right-click **Page14 (type)** and select **New** > **Templates** > **MyObject**.
2. Repeat the previous step 2 more times



© 2025 Rockwell Automation, Inc. All rights reserved.

3. Select **MyObject1** under **Page14 (type)** to change its properties. In **Properties**, set the following

— Size and layout	
Horizontal alignment	Left
Vertical alignment	Center
Width	*200*
Height	*100*
Left margin	50
Top margin	*0*
Right margin	*0*
Bottom margin	*0*

4. Select **MyObject2** under **Page14 (type)** to change its properties. In **Properties**, set the following

— Size and layout	
Horizontal alignment	Center
Vertical alignment	Center
Width	*200*
Height	*100*
Left margin	*0*
Top margin	*0*
Right margin	*0*
Bottom margin	*0*

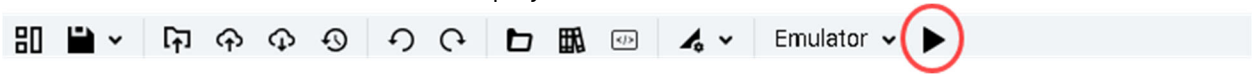
5. Select **MyObject3** under **Page14 (type)** to change its properties. In **Properties**, set the following

— Size and layout	
Horizontal alignment	Right
Vertical alignment	Center
Width	*200*
Height	*100*
Left margin	*0*
Top margin	*0*
Right margin	50
Bottom margin	*0*

6. Mouse-over **Page14 (type)**, select the edit icon , and enter "**VH_Alignment**" as the new name and hit enter.
7. Save.

© 2025 Rockwell Automation, Inc. All rights reserved.

8. From the toolbar, with **Emulator** still displayed. Click the Run Emulator icon ▶.



9. Select **VH_Alignment** from the navigation bar.
10. Change the size of the runtime window and watch the object maintain position and change size as needed.

    Note: **VH_Layouts** and **VH_Alignment** will respond to screen sizing in a similar way.

11. Close the **Emulator.**

SUMMARY

A FactoryTalk Optix application may be developed with responsive graphics in mind to handle window resizing and runtime deployment to different platforms.

In this section:

- We explored layouts using containers and their relative object positions within the container determined by their margins and alignments.
- We used three types of containers: **Scale layout**, **Horizontal layout**, and **Vertical layout** to demonstrate this feature.
- We learned that the UI is capable of responding to changes dynamically at runtime.
    - Recall that **Scale layout** was needed for the logos to change size as well as position when resizing the window.

© 2025 Rockwell Automation, Inc. All rights reserved.

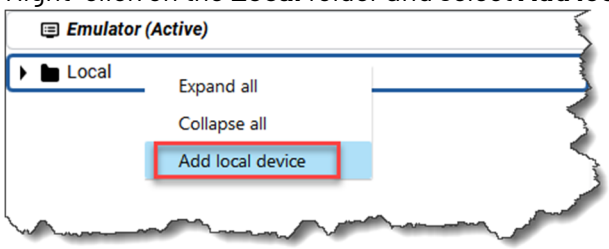
Appendix E - Transfer application to a runtime device

Deploy your project to the preferred target device (Check the latest release notes for more information)

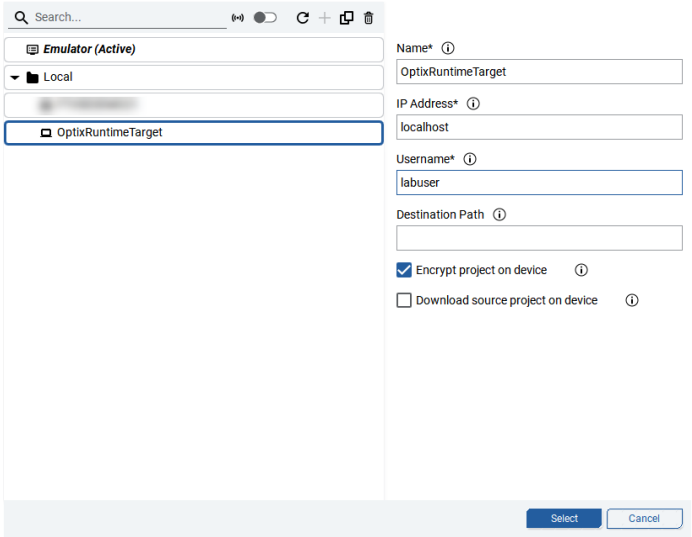
- A target device running a Windows operating system
- A target device running Ubuntu Linux operating system
- An OptixPanel Graphic terminal (these terminals run a Linux OS)
- A Logix Embedded Edge Compute module (These modules run a Linux OS)
- A Docker container (**New!** Starting with FactoryTalk Optix version 1.4)
- An OptixEdge (**New!** Planned AFC 2025)

1. In Optix Studio, click the dropdown arrow next to Emulator.

2. Right-click on the **Local** folder and select **Add local device**.



36. Modify the **Name** field of the target device to **OptixRuntimeTarget**.
37. Enter **localhost** in the **IP Address** field.
38. In the **Username** field enter the name of the local user of the target device.
10. Your target device configuration should look similar to this:

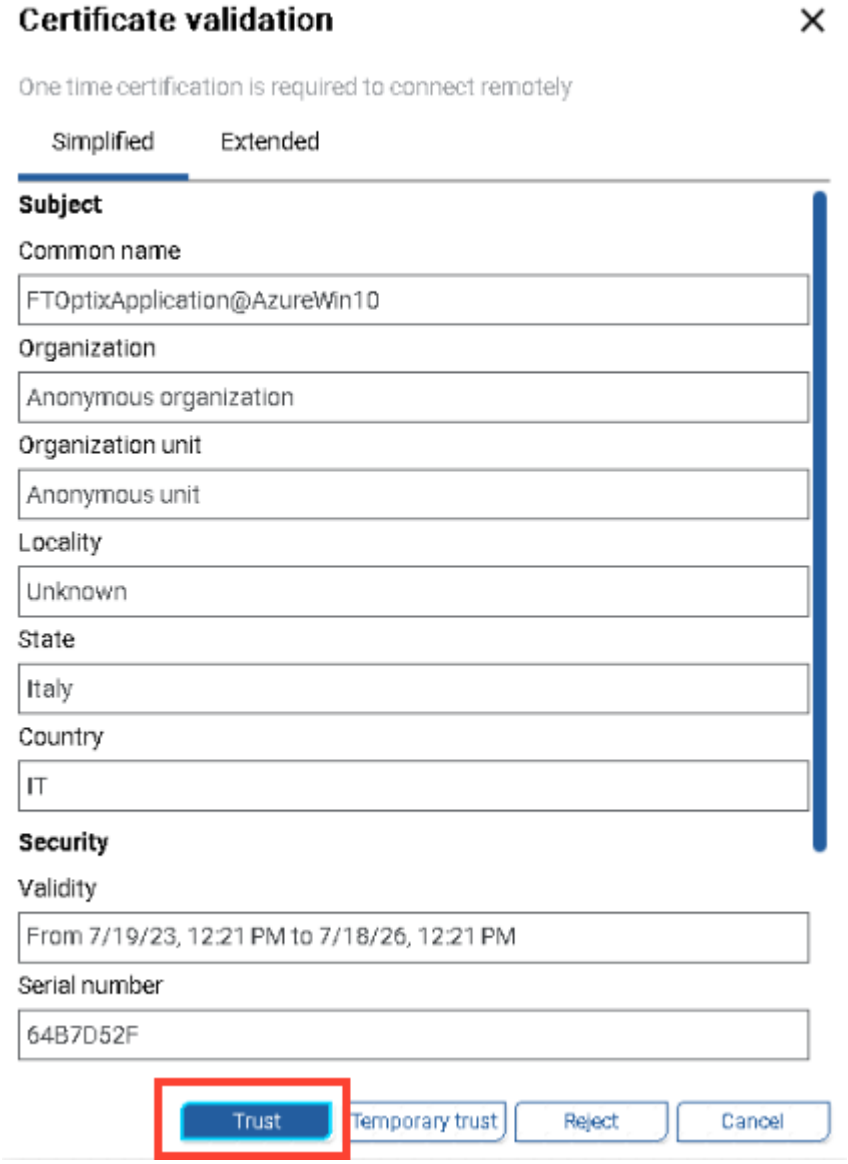


© 2025 Rockwell Automation, Inc. All rights reserved.

39. Click **Select** to close the configuration window and save the changes.
40. Save the project.

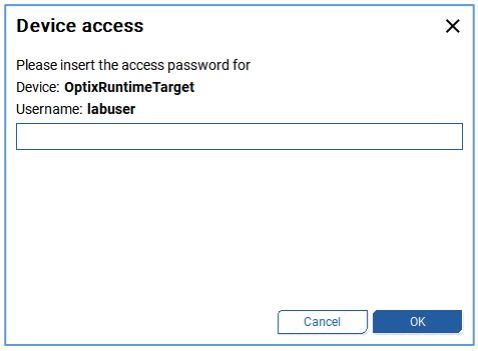
    Note: Observe how the target device in the toolbar changed from Emulator to OptixRuntimeTarget

41. Click the **Run** icon ▶.
42. If you are prompted with a certificate validation window, leave the defaults, and select **Trust.**



© 2025 Rockwell Automation, Inc. All rights reserved.

43. Enter the password of the user for the target device and click **OK.**



Note: Once deployed to the new target device, the application will start automatically.

New! Starting with FactoryTalk Optix version 1.3, you can deploy your project to a runtime target device with the "download source project to device" checkbox selected. This allows you to upload the source project from the runtime target device later. Otherwise, uploading the source project from a runtime target device is not possible.

New! Starting with FactoryTalk Optix version 1.4, you can customize the installation to install FactoryTalk Optix runtime as a service. If you do not install FactoryTalk Optix runtime as a service, it installs as an application by default. Therefore, when you deploy an application, the mode in which the FactoryTalk Optix Application runs depends on how FactoryTalk Optix Runtime is configured upon installation. The two FactoryTalk Optix Runtime modes are:

- FactoryTalk Optix running as an application mode: When FactoryTalk Optix is running as an application, the deployed application is accessible with the native presentation engine and web presentation engine. This mode is generally used by a local user logged onto the device or accessing the HMI and any remote users connecting from a web client if the web presentation was added.
- FactoryTalk Optix running as a service mode: In this mode, FactoryTalk Optix runs the deployed application as a Windows service. The service starts when the runtime device starts and automatically runs the deployed application before a user is logged in. This mode may use the Web Presentation Engine that you can add to Project view. Use this mode to interact with the application using the web browser of any device that can access the runtime device through an Ethernet IP address. Without a Web Presentation Engine, the application still runs without displaying on a user interface. This mode is generally used when the application has no user interface, such as a data gateway.

  Tip: The Native Presentation Engine is not supported in FactoryTalk Optix Runtime as a service mode.

© 2025 Rockwell Automation, Inc. All rights reserved.

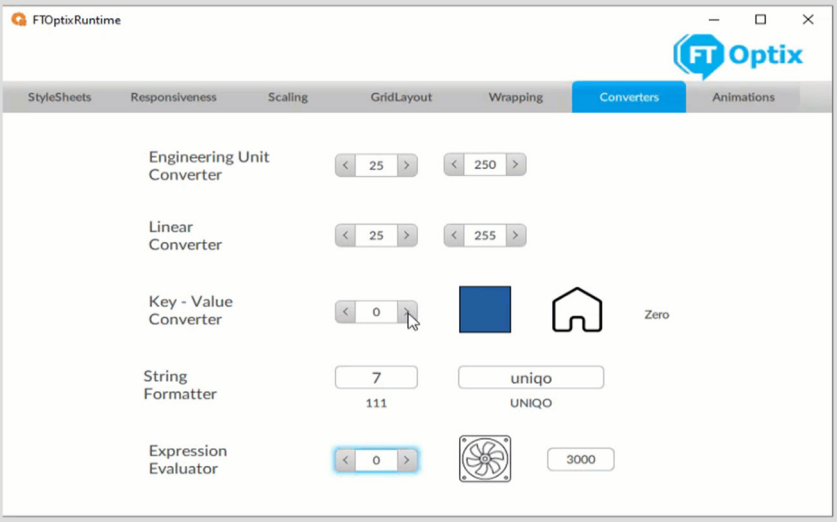
Appendix F – Converters

FactoryTalk Optix includes these predefined converter types:

- **Conditional converter** (Verify if the source variable value is True or False)
- **Engineering unit converter** (to convert raw data values into scaled values based on a range of customizable values)
- **Expression evaluator** (Calculate the result of an expression with integers, decimals, numeric variables, string variables for string comparisons, arithmetic and Boolean operators, or a set of functions)
- **Key-value converter** (Convert the value of the source to its corresponding parent value based on a table of key-value pairs. The key-value converter converts heterogeneous data, such as a color to value)
- **Linear converter** (Convert raw data values into scaled values based on the linear relationship applied to the input)
- **String formatter** (Modify the formatting of one or more values according to a customizable rule)

CONVERTERS

To learn more about converters, check the sample project called **Training_UserInterface** available through the public git repository https://github.com/factorytalk-Optix



You have completed this lab.

© 2025 Rockwell Automation, Inc. All rights reserved.