



FactoryTalk Design Workbench Getting Results Guide

Version 1.01.00



Important User Information

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

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

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

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

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

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

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

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



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



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

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

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



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



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



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

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



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

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

FactoryTalk Design Workbench overview	9
Installation.....	9
System requirements.....	9
Install FactoryTalk Design Workbench.....	12
Unattended or silent install.....	12
User interface.....	14
Welcome page.....	16
About dialog.....	16
Project manager.....	17
Project Organizer.....	17
Commands in Project Organizer.....	18
Menu bar.....	19
View tabs.....	20
View panes.....	20
Reset the layout.....	21
Keyboard shortcuts.....	21
Pane bar.....	23
Output pane.....	24
Properties pane.....	27
Controller application workspace.....	27
Online center.....	29
Online settings.....	30
Projects	32
Device and project compatibility.....	32
Create a project.....	33
Manage devices.....	33
Controllers catalog.....	34
Add a controller to the project.....	35
Discover and add a controller connected to a computer.....	36
Secure a controller with a password.....	37
Connect to a password-protected controller.....	37
Change the controller password.....	38
Clear the controller password.....	38
Data protection.....	39
Add variables for data protection.....	40
Change the Micro800 controller project.....	40

Change the Micro800 controller type.....	42
Build a project and view outputs.....	43
Set up the connection path.....	43
Add a serial-DF1 driver	44
Serial-DF1 properties.....	45
Connect to a Micro800 controller.....	46
Download a controller configuration to a controller.....	47
Upload a controller configuration from a controller.....	47
Monitor local and global variables.....	48
View the controller mode.....	49
Change the Micro800 controller Remote mode.....	49
View the controller fault status.....	49
Monitor and diagnose a Micro800 controller.....	50
Open the diagnostics pages.....	50
Disconnect from a Micro800 controller.....	51
Fault diagnostics.....	51
Fault codes and corrective actions.....	52
Controller run mode change	65
Run mode change actions.....	65
Run mode change toolbar.....	67
Make run mode changes.....	67
Project import and export.....	68
Import a project.....	68
Export projects.....	69
Export a project using the File menu.....	69
Export multiple projects using Project manager.....	70
Import and export project elements.....	70
Export file descriptions.....	71
Export programs, function blocks, variables, and user-defined data types.....	71
Export a program or function block.....	72
Export global variables.....	72
Export local variables.....	73
Export user-defined data types.....	73
Import controllers, programs, function blocks, variables, or user-defined data types.....	74
Import recommendations and considerations.....	74
Import a program or function block.....	74
Import global variables.....	75

Import user-defined data types.....	77
Import local variables.....	77
Work in multiple sessions of FactoryTalk Design Workbench.....	78
Copy a Program Organization Unit from one session and paste it to another.....	79
Programs.....	80
Add a program to a project.....	80
Add a user-defined function block using Project Organizer.....	81
Add a user-defined function using Project Organizer.....	81
Add Program Organization Unit descriptions.....	81
Duplicate a Program Organization Unit.....	81
Delete a Program Organization Unit.....	82
Verify a Program Organization Unit.....	82
Synchronize user-defined function block or user-defined function updates to its references.....	82
Interrupt overview.....	83
Add an interrupt.....	84
Interrupt instructions.....	85
I/O configurations.....	86
Embedded I/O configuration page.....	86
Configure a latch input.....	88
Plug-in modules.....	89
Add a plug-in module to the controller.....	90
Expansion modules.....	91
Add an expansion module to a controller project.....	92
Configure the scan interval for expansion modules.....	93
Replace an expansion module.....	94
Ethernet device settings.....	95
Add an Ethernet device.....	97
Language editors.....	99
Work with elements in language editors.....	99
Toolbox.....	100
Create a variable in language editors.....	100
View instruction block logic.....	101
View instruction block parameters.....	101
Ladder diagram language.....	102
Ladder diagram program development environment.....	102
Ladder editor Toolbox.....	103
Ladder diagram value display.....	104

Work in the ladder diagram language editor.....	107
Toggle Bit command.....	107
Ladder diagram program examples.....	108
Enable EN/ENO in ladder diagram instructions.....	109
Function block diagram language.....	111
Function block diagram program development environment.....	111
Function block diagram Toolbox.....	112
Function block diagram main format.....	113
Function block diagram program inputs and outputs.....	114
Instruction blocks in function block diagram programs.....	114
Execution order of function block diagram programs.....	115
Re-order elements in function block diagram programs.....	116
Monitoring function block diagram programs.....	116
Structured text (ST) language.....	117
Structured text program development environment.....	117
Structured text Toolbox.....	118
Commands in structured text editor.....	119
Code snippets.....	120
Insert a code snippet.....	120
Structured text main syntax.....	121
Call functions.....	122
Call function blocks.....	123
Structured text keyboard shortcuts.....	125
FactoryTalk Design Workbench settings.....	126
Switch FactoryTalk Design Workbench language.....	126
Product lifecycle information.....	126
Update the product lifecycle information.....	126
Configure general controller settings.....	127

Preface

The document set includes:

- *FactoryTalk Design Workbench Getting Results Guide* (publication FTDW-GR001). Use this manual to get started with the FactoryTalk® Design Workbench™ development environment. The manual is available in [Rockwell Automation® Literature Library](#).
- Help: Use this comprehensive Help for assistance when using FactoryTalk Design Workbench.
- *Micro800 Programmable Controllers General Instructions* (publication 2080-RM003): Use this manual for reference information about the instruction set available for developing programs for use in Micro800™ control systems. The manual is available in [Rockwell Automation Literature Library](#).

Summary of Changes

This manual has new and updated information. Use these reference tables to locate new or changed information. Grammatical and editorial style changes are not included in this summary.

Global changes

None for this release.

New or enhanced features

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

The identified changes will fall into one of these categories:

- New feature
- Enhanced feature
- Functional change
- Anomaly fix
- Usability improvement
- Clarification

Table 1. Summary of changes

Category	Topic name	Reason
New feature	User interface on page 14	Updated the user interface (UI) screenshot and UI item description to reflect the run mode change (RMC) function.
New feature	<ul style="list-style-type: none"> • Controller run mode change on page 65 • Run mode change actions on page 65 • Run mode change toolbar on page 67 • Make run mode changes on page 67 	Added topics to document the RMC function.
Clarification	Throughout the book	Replaced abbreviated controller names with their full Micro800 equivalents to improve consistency.

Table 1. Summary of changes (continued)

Category	Topic name	Reason
Clarification	Connect to a Micro800 controller on page 46	Enhanced the topic by adding guidance on discovering controllers and refining step 1 for better understanding.
Clarification	Discover and add a controller connected to a computer on page 36	Enhanced the topic by refining steps from 1 through 5 for better understanding.

About this book

This manual provides information on how to get started using the FactoryTalk Design Workbench software. It is not intended to be a comprehensive user guide. For complete information on all features, tasks, and components, refer to Help.

Legal Notices

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

Software and Cloud Services Agreement

View and sign the Rockwell Automation Software and Cloud Services Agreement [here](#).

Third-Party Software Licenses

View a full list of all third-party software used in this product [here](#), including:

- Open-source software
- Commercial software
- Other third-party free software

You may obtain Corresponding Source code for open-source packages included in this product from their respective project websites. Alternatively, you may obtain the complete Corresponding Source code by contacting Rockwell Automation via the Contact form at the bottom of the Rockwell Automation website: <http://www.rockwellautomation.com/global/about-us/contact/contact.page>. Please include "Open Source" as part of the comments for your General Inquiry request.

FactoryTalk Design Workbench overview

FactoryTalk Design Workbench is an on-premises development environment that provides web-based access for configuring devices and programming Micro800 controllers. FactoryTalk Design Workbench version 1.00.00 supports Micro800 controllers, allowing one controller in each project. Later versions will support more devices.

Installation

This section describes:

- System requirements
- How to install the FactoryTalk Design Workbench software

System requirements

FactoryTalk Design Workbench works within the system requirements of all Rockwell Automation® software products. For the latest compatibility information, refer to the [Product Compatibility and Download Center](#).

Hardware requirements

- Processor: Intel® Core™ i5 Standard Power processor (i5-3xxx) or equivalent recommended. At least 2 processors should be assigned when running FactoryTalk Design Workbench on a virtual machine.
- RAM: 8 GB minimum
- Hard disk space: 20 GB free

Software requirements

The following operating systems are supported and recommended*:

- Windows Server® 2025 Standard (Recommended OS)**
- Windows Server 2022 Standard (Recommended OS)**
- Windows Server 2019
- Windows® 11 Enterprise 64-bit (Recommended OS)
- Windows 11 Professional 64-bit (Recommended OS)
- Windows 11 Enterprise 2024 LTSC 64-bit (v24H2)
- Windows 11 IoT Enterprise 2024 LTSC 64-bit (v24H2)***
- Windows 10 Enterprise 2019 LTSC 64-bit (v1809)
- Windows 10 Enterprise 2021 LTSC 64-bit (v21H2)
- Windows 10 IoT Enterprise 2021 LTSC 64-bit (v21H2)***

*The recommended operating systems have priority for [Microsoft Patch Qualifications](#) with Rockwell Automation software.

**Windows Server Datacenter editions are supported. The key difference between Windows Server Standard and Datacenter edition is how Microsoft licenses the operating system. Windows Server Standard includes licensing for up to two virtual machines per core license, making it suitable for lightly virtualized environments. Datacenter edition allows unlimited virtual machines per core license, offering better value for highly virtualized or cloud-scale deployments.

***Windows 11 IoT Enterprise LTSC and Windows 11 Enterprise LTSC are binary equivalents but are licensed differently. Windows 10 IoT Enterprise LTSC and Windows 10 Enterprise LTSC are binary equivalents but are licensed differently.

Rockwell Automation's software installation policy is based on the lifecycle information of Microsoft operating systems. If an operating system's lifecycle state is approaching its end, you may encounter a warning message when trying to install Rockwell Automation software on it.

Microsoft lifecycle policy	Rockwell Automation installation warning policy	Rockwell Automation installation prevention policy
Fixed	Present a warning message during installation on Microsoft operating systems that are 18 months before Microsoft's extended end date.	Prevent installation on Microsoft operating systems that are six months past Microsoft's extended end date.
Modern	Present a warning message during installation on Microsoft operating systems that are 18 months before Microsoft's retirement date.	Prevent installation on Microsoft operating systems that are six months past Microsoft's retirement date.

The following table illustrates how the installation policy is applied using the current operating systems. For the latest compatibility information, refer to the [Microsoft Lifecycle Policy](#) and [Product Compatibility and Download Center](#).

Operating system	Microsoft lifecycle policy	Extended end date	Retirement date	Installation warning	Installation prevented
Windows 11 Enterprise and Professional editions	Modern	None	None	None	None
Windows 10 Enterprise and Professional editions	Modern	None	October 14, 2025	April 14, 2024	April 14, 2026
Windows Server 2025 Standard and Datacenter editions	Fixed	October 10, 2034	None	April 10, 2033	April 10, 2035
Windows Server 2022 Standard	Fixed	October 14, 2031	None	April 14, 2030	April 14, 2032

Operating system	Microsoft lifecycle policy	Extended end date	Retirement date	Installation warning	Installation prevented
and Datacenter editions					

Supported browsers

- Google® Chrome™ browser
- Microsoft Edge™
- Mozilla® Firefox®

High Resolution Display Support

Windows 10 version 1803 or later is the recommended operating system when using this software with a 2K or 4K high-resolution display. For optimal user experiences, ensure that the Windows recommended scaling rate is applied, especially for 4K displays. Using other scaling rate might cause scrollbar functionality issues.

Rockwell Automation Test Environment

Rockwell Automation tests software products under a standard configuration of operating systems and antivirus software. For additional information, see the Knowledgebase Document ID: [PN24 - Rockwell Software Products and Antivirus Software](#).

Security requirements

To learn about designing a secure networking environment, review this publication:

- [Configuring Firewalls for FactoryTalk Services Platform](#) (publication FTSP-RM001) shows the ports, services, and firewall rules necessary for FactoryTalk® Services Platform components to function properly.
- [Converged Plantwide Ethernet \(CPwE\) Design and Implementation Guide](#) (publication ENET-TD001) is built on, and adds to, design guidelines from the Cisco Ethernet-to-the-Factory (EttF) solution and the Rockwell Automation Integrated Architecture™.

To learn about implementing application-level security using FactoryTalk Security, review this publication:

- [FactoryTalk Security Application Technique](#) (publication SECURE-AT002) explains how to use FactoryTalk Security to implement authentication and authorization in your industrial automation system and how to enforce product-specific security for applications like Studio 5000 Logix Designer, FactoryTalk View, and FactoryTalk AssetCentre.

To learn about designing and implementing an architecture with device-level security using CIP Security, review these publications:

- [System Security Design Guidelines Reference Manual](#) (publication SECURE-RM001) provides guidelines for how to use Rockwell Automation products to improve the security of your industrial automation system.
- [Configure System Security Features User Manual](#) (publication SECURE-UM001) describes the system-level configuration requirements to use a controller that has achieved IEC 62443-4-2:2019 certification.
- [CIP Security with Rockwell Automation Products Application Technique](#) (publication SECURE-AT001) explains how to implement the Common Industrial Protocol (CIP) Security standard in your industrial automation control system (IACS).

Install FactoryTalk Design Workbench

Follow the steps to install FactoryTalk Design Workbench. Ensure that you have administrative rights in Windows.



To use Software Center Configuration Manager (SCCM), install it with a user account or manually install Visual Studio shell 2013 before installing FactoryTalk Design Workbench.

To install FactoryTalk Design Workbench

1. From [Product Compatibility and Downloads Center](#), download the current version of FactoryTalk Design Workbench.
2. In the installation package, double-click **Setup.exe**.
3. On the welcome page, select a language to be shown during the installation process. By default, your system language is selected.
4. To install all components available using the recommended settings, select **Install now**, and then skip to step 9.
5. To select which component to install, select **Customize**.
6. On the **Customize** page, select the components. There might be three options shown:
 - **Mandatory** (grayed-out and selected checkbox) indicates software that will be automatically installed as part of the selected application.
 - **Recommended** (selected checkbox) indicates software that Rockwell Automation recommends for the application. You might decide to clear the checkbox so the software does not install.
 - **Optional** (clear checkbox) indicates software that you might wish to include depending on your system. Select the box to include the software during installation.
7. Select the location for the Rockwell Automation software, and then select **Install**. The default location is `C:\`.
8. In **That's it!**, select **Close** to exit the installation, or:
 - Select **Installation Summary** to see the installation details.
 - Select **Register for updates** to learn how to receive email updates about product patches.
 - Select **Download it free** to install Adobe Acrobat Reader.

Unattended or silent install

Use command-line parameters to perform an unattended or silent installation of the software.

Installation Command-line parameters

Use command Setup.exe /? to display the usage options for installation parameters. Command-line parameters are case-insensitive. If a specified value includes a space, be sure to enclose the value in quotation marks (for example, "value with spaces").

Examples

The following examples show how to use the installation commands.

- To install the software with no user interface using the default settings during the installation process. (Silent install)

```
Setup.exe /Q /IAcceptAllLicenseTerms
```

- To install the Chinese language version of the software on the D: drive and display the progress, error, or complete messages during installation and restart the computer if necessary. (Unattended install)

```
Setup.exe /QS /IAcceptAllLicenseTerms /AutoRestart /  
SetupLanguage=CHS /InstallDrive=D:
```

Error codes

The following table identifies the error codes that can be returned by an installation.

Error Code	Value	Description
ERROR_SUCCESS	0	The installation completed successfully.
ERROR_INVALID_PARAMETER	87	One of the parameters was invalid.
ERROR_INSTALL_USEREXIT	1602	The installation was canceled by the user.
ERROR_INSTALL_FAILURE	1603	A fatal error occurred during installation.
ERROR_BAD_CONFIGURATION	1610	The configuration data for this product is corrupt. Contact your support personnel.
ERROR_SUCCESS_REBOOT_INITIATED	1641	The installer has initiated a restart. After restart, the installation will continue.
ERROR_SUCCESS_REBOOT_REQUIRED	3010	A restart is required to complete the installation. After restart, the product is successfully installed.
ERROR_REBOOT_PENDING	3012	Restart pending. Restart the computer for the installation to continue.
ERROR_SUCCESS_NOT_APPLICABLE	3013	The installation cannot proceed because the products are already installed.
ERROR_SUCCESS_WARNING_REBOOT	3014	The installation succeeded with warnings. Check the installation log file for details. To complete the installation, restart the computer.

User interface

Use the FactoryTalk Design Workbench user interface (UI) to create projects, specify which panes to view, and reset to the default layout if necessary.

Figure 1. FactoryTalk Design Workbench user interface

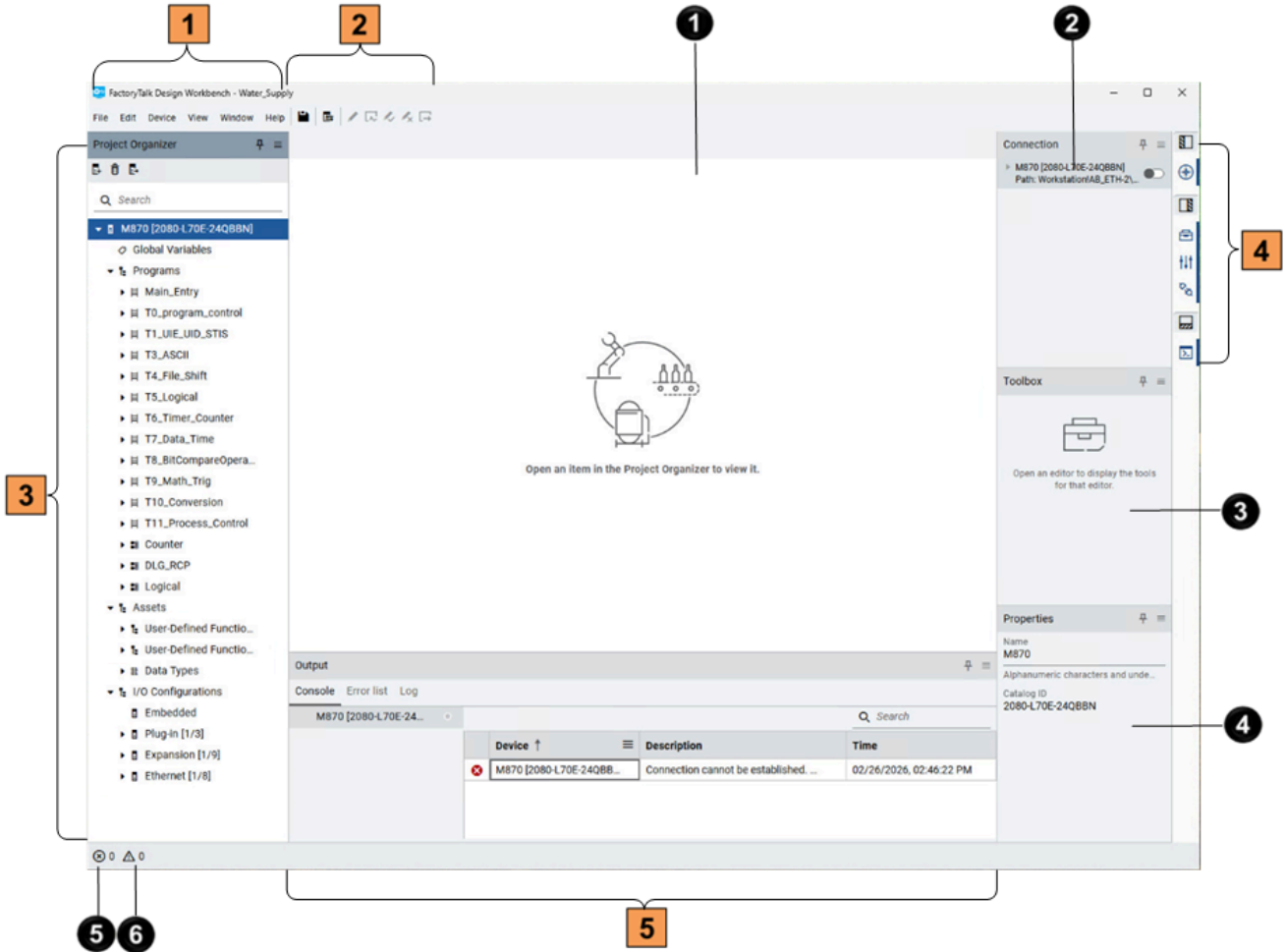


Table 2. Items on the UI

Item	Name	Description
1	Menu bar on page 19	Contains the menu items for FactoryTalk Design Workbench and the active window.
2	Toolbar	Contains: <ul style="list-style-type: none"> The Save and Online center buttons. The run mode change (RMC) toolbar with buttons to start online editing, test, accept, and finalize changes during RMC.

Table 2. Items on the UI (continued)



Item	Name	Description
		<p>If RMC changes are ongoing, not tested, or not accepted in a Program Organization Unit (POU), the following icons appear beside that POU in Project Organizer to indicate the changes:</p> <ul style="list-style-type: none"> -  indicates that there are ongoing or unsaved RMC changes. Before reconnecting to the controller, you can choose to resume editing. -  indicates that testing is in progress on a POU. Before reconnecting to the controller, you can choose to continue testing the POU.
3	Project Organizer on page 17	Displays the devices, I/O configurations, and program elements of the project in an organized tree view.
4	Pane bar on page 23	Contains the button toggles you use to show or hide specific panes.
5	Output pane on page 24	<p>Contains:</p> <ul style="list-style-type: none"> • A Console page that displays the added devices, their connection status, and log entries. • An Error list page that displays the descriptions, locations, and time stamps of errors, warnings, and messages. • A Log page that displays the log descriptions and time stamps.
1	Workspace	Contains the options to configure devices and build programs. The content varies with the page you open.
2	Connection pane	<p>For each added device, the Connection pane contains:</p> <ul style="list-style-type: none"> • The device name and catalog number. • The connection path to connect to the device if any. • The toggle to connect to or disconnect from the device. • The Controller mode list to view and change the controller connection mode. • The Fault button to open the Micro800 controller's fault diagnostics on page 51 page. • The Diagnose button to open the controller and communication diagnostics pages on page 50. • The Secure button to open the pages to set, clear, or change the controller password.
3	Toolbox on page 100	Contains the logic elements and instructions that can be added to the opened language editor.
4	Properties pane on page 27	<p>Displays the properties of a selected item.</p> <p>You can edit the properties in the Properties pane.</p>

Table 2. Items on the UI (continued)

Item	Name	Description
5	Error indicator	Displays the total number of errors in the current project, counting only errors from POUs that are built and verified.
6	Warning indicator	Displays the total number of warnings in the current project, counting only errors from POUs that are built and verified.

Welcome page

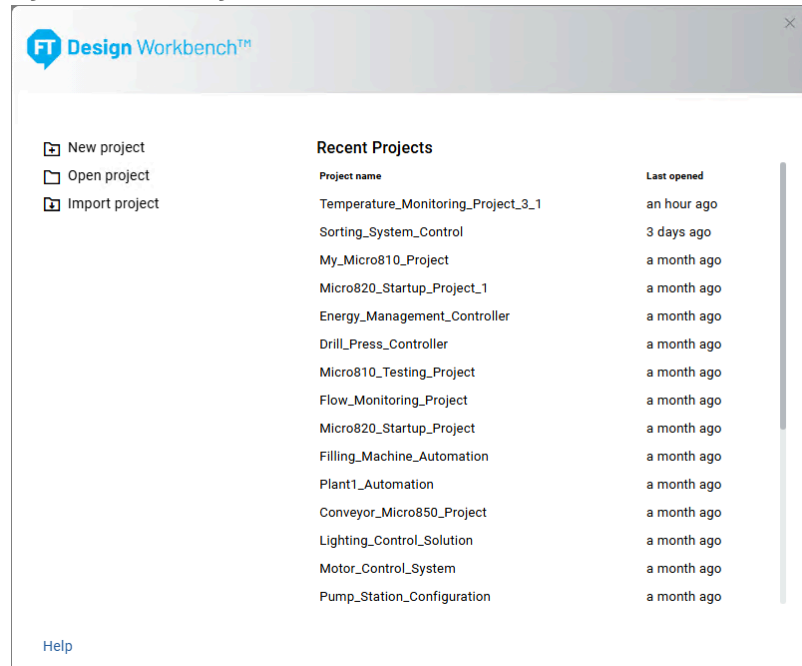
When you open FactoryTalk Design Workbench, the **Welcome** page opens by default.

Use **Welcome** to:

- Create a project.
- Open a project.
- Import a project.
- View the most recently opened projects. The information of up to 25 projects is displayed.

Select **X** to close **Welcome**. To reopen it, from the **File** menu, select **Welcome**.

Figure 2. Welcome page



About dialog

Use the **About** dialog to view the following information:

- The product name and version.
- The name and version of the added devices.

- The copyright information.
- The link to the release notes.

To open **About**, from the **Help** menu, select **About**.

Project manager

Project manager lists all FactoryTalk Design Workbench projects on your computer, with the project names and the time stamps of their last editing.

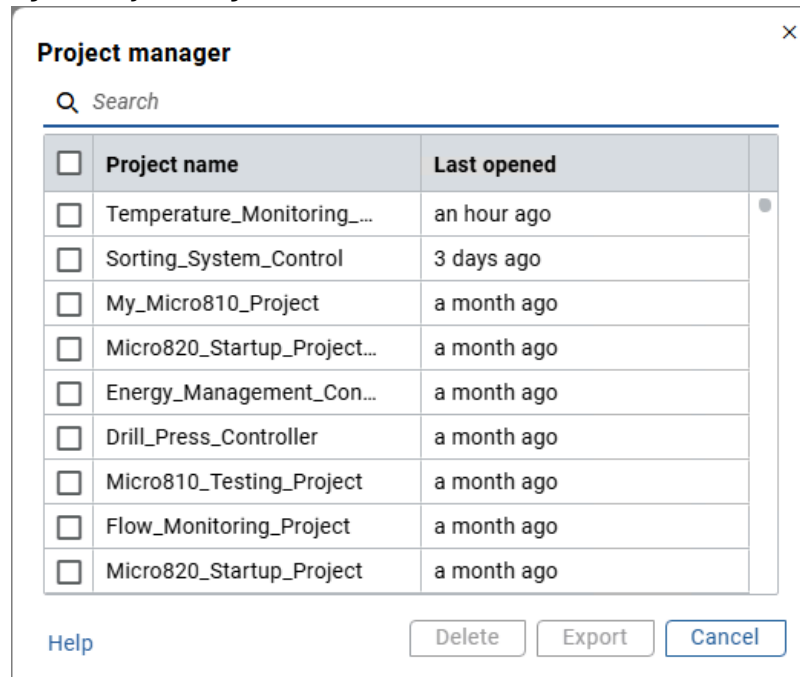
Use **Project manager** to:

- Batch-export projects to your computer.
- Batch-delete projects from your computer.
- Rename projects.

When a project is opened in FactoryTalk Design Workbench, you cannot perform any of the actions.

To open **Project manager**, from the **File** menu, select **Project manager**.

Figure 3. Project manager



Project Organizer

Project Organizer displays the devices and program elements of the project in an organized tree view. Use **Project Organizer** to open additional device configuration workspaces and manage program elements. If no device is added, use the [New device on page 35](#) or [Discover Device on page 36](#) button to add one. Elements displayed in **Project Organizer** vary with the added device.

For projects that contain a Micro800 controller, **Project Organizer** displays the following items that are associated with the controller:

- Global variables
- [Logic programs on page 80](#)
- Assets, including user-defined functions, user-defined function blocks, and user-defined data types
- I/O configurations

Figure 4. Project Organizer

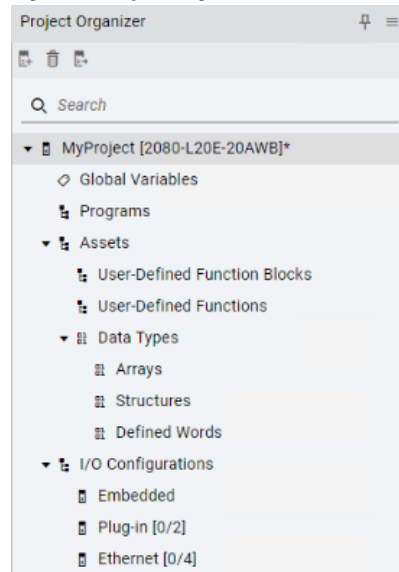





Table 3. Icons in Project Organizer

Item	Description
	Opens Add device to add devices to the project.
	Deletes the selected device from the project.
	Opens Select Device to view a list of devices that are connected to the computer.

Commands in Project Organizer

Use **Project Organizer** to open and edit additional device configuration options. Micro800 controllers and drives are available in **Project Organizer**.

Commands available for Micro800 controllers

Table 4. Commands available by right-clicking the Micro800 controller in Project Organizer

Right-click and select	To
Build	Build the current project on page 43.
Connect	Connect to a Micro800 controller on page 46.

Table 4. Commands available by right-clicking the Micro800 controller in Project Organizer (continued)

Right-click and select	To
Disconnect	Disconnect from a Micro800 controller on page 51.
Download	Download a controller configuration to a Micro800 controller on page 47.
Upload	Upload a controller configuration from a Micro800 controller on page 47.
Clean	Clean controller faults on page 51.
Fault	Open <controller-name>:Fault to view and clear controller faults.
Diagnose	Monitor and diagnose a Micro800 controller on page 50.
Secure	Set, change, or clear a controller password on page 37.
Change Controller	Replace the Micro800 controller in the project with a different Micro800 model on page 42.
Import > Global variables	Open Import global variables to import variables from a CSV or XLSX file on page 75.
Import > Data types	Open Import Data types to import data types from an M8A file on page 77.
Export > Global variables	Open Export global variables to export global variables to a CSV or XLSX file on page 72.
Export > Data types	Open Export data types to export data types to an M8A file on page 73.
Delete	Remove the Micro800 controller from Project Organizer . You can also use the Delete key.
Data protection	Open Data protection on page 39 to add or remove variables from a list of the protected variables.
Memory module backup and restore	Open Memory module backup and restore to configure the relevant settings and start the backup or restore process.

Commands available for Class 1 devices, plug-in modules, and expansion modules

Table 5. Commands available by right-clicking a Class 1 device, plug-in module, or expansion module in Project Organizer

Right-click and select	To
Delete	Remove a drive, plug-in module, or expansion module from Project Organizer . You can also use the Delete key to remove the device.

Menu bar

The menu bar contains the menu items for the active window.

Table 6. Items on the menu bar

Use this menu	To
File	<ul style="list-style-type: none"> • Create, open, or close a project. • Save a project to your local drive. • Duplicate a project to the same location using a different name. • Import or export projects on page 68. • Open Welcome on page 16. • Open Project manager on page 17. • Open Settings on page 126 to configure global options. • Close FactoryTalk Design Workbench.
Edit	Modify content in the active window.
Device	Perform the actions that are available by right-clicking the Micro800 controller in Project Organizer .
View	<ul style="list-style-type: none"> • Select which panes to display on page 20. • Reset the layout on page 21.
Window	Select which page to open in the workspace on page 20.
Help	Open About on page 16.

View tabs

In FactoryTalk Design Workbench, pages open for different functions. Use the **Window** menu to select which page to view or close.

To view tabs

1. From the menu bar, select **Window**.
2. From the **Window** menu:
 - Select **Next tab** to view the next opened page.
 - Select **Previous tab** to view the previously opened page.
 - Select **Close tab** to close the current page.
 - Select **Close all tabs** to close all opened pages.
 - Select **All tabs** to view a list of opened pages and select which page to view.

View panes

Use the **View** menu to display or hide specific panes. The location, pinned status, and visibility of panes are reserved after closing and reopening FactoryTalk Design Workbench.

Table 7. View menu items

Select	To
Pane > Pin	Pin the selected pane to the current location.

Table 7. View menu items (continued)

Select	To
Pane > Unpin	Unpin the selected pane.
Pane > Collapse	Hide the content of the selected pane.
Pane > Expand	Display the content of the selected pane.
Pane > Close	Close the selected pane.
Left panes	Display or hide the panes that are last pinned to the left side of the user interface (UI).
Right panes	Display or hide the panes that are last pinned to the right side of the UI.
Bottom panes	Display or hide the panes that are last pinned at the bottom of the UI.
Reset layout	Restore panes to their default locations, pinned status, and visibility.
Project organizer	Display or hide Project Organizer . Project Organizer is pinned to the left side of the UI by default.
Properties	Display or hide Properties . The pane is pinned to the lower right of the UI by default.
Toolbox	Display or hide Toolbox . The pane is pinned to the upper right of the UI by default.
Connection	Display or hide Connection . The pane is hidden by default.
Output > Console	Display Output with the Console page opened.
Output > Error list	Display Output with the Error list page opened.
Output > Log	Display Output with the Log page opened.

Reset the layout

Reset the layout to restore the panes to their default locations, pinned status, and visibility. Layout changes are reserved after closing and reopening FactoryTalk Design Workbench.

To reset the layout

- From the **View** menu, select **Reset layout**.

Keyboard shortcuts

The following table introduces the keyboard shortcuts that apply universally.

Table 8. Keyboard shortcuts

Name	Keyboard Shortcut	Description
Select all	Ctrl+A	Selects all elements on the current page.
Show/hide bottom panes	Ctrl+Shift+B	Displays or hides the panes that are last pinned at the bottom of the user interface (UI).
Copy	Ctrl+C	Copies the selected element to the clipboard.
Show/hide pane	Ctrl+Shift+C	Displays or hides the selected pane.

Table 8. Keyboard shortcuts (continued)

Name	Keyboard Shortcut	Description
Find references	Ctrl+E	Finds all references to the selected variable.
Select variable	Ctrl+L	Opens Select variable from the ladder diagram (LD), function block diagram (FBD), or structured text (ST) editor to select a variable for the selected operand.
Show/hide left panes	Ctrl+Shift+L	Displays or hides the panes that are last pinned to the left side of the UI.
Toggle inline value display	Ctrl+M	Displays or hides the values of the variables.
New project	Ctrl+N	Opens New project to create a project when no project is opened.
Open project	Ctrl+O	Opens Open project to select and open a recently opened project when no project is opened.
Pin/unpin panes	Ctrl+Shift+P	Pins or unpins the selected pane.
Add rung	Ctrl+R	Adds a rung below the selected rung in the LD editor.
Show/hide right panes	Ctrl+Shift+R	Displays or hides the panes that are last pinned to the right of the UI.
Save	Ctrl+S	Saves the controller project.
Save as	Ctrl+Shift+S	Saves a copy of the controller project.
Toggle bit	Ctrl+T	Switches a Boolean variable between on and off states.
Paste	Ctrl+V	Pastes the selected element to the selected position.
New variable	Ctrl+W	Opens New variable to create a variable at the current location in the LD, FBD, or ST editor.
Cut	Ctrl+X	Cuts the selected element to the clipboard.
Close pane	Ctrl+Shift+X	Closes the selected pane.
Redo	Ctrl+Y	Redoes the previous command.
Undo	Ctrl+Z	Undoes the previous command.
Show Connection	Ctrl+Alt+C	<ul style="list-style-type: none"> Displays the Connection pane if it is hidden. Highlights the Connection if it is already opened.
Show Error list	Ctrl+Alt+E	Displays the Error list page when the Output pane is selected.
Show Log	Ctrl+Alt+L	Displays the Log page when the Output pane is selected.
Show/hide Project Organizer	Ctrl+Alt+P	Displays or hides Project Organizer .
Show Console	Ctrl+Alt+S	Displays the Console page when the Output pane is selected.
Zoom in	Ctrl++	Zooms in the language editor.
Zoom out	Ctrl+-	Zooms out the language editor.
Previous tab	Ctrl+[Goes to the previous tab.

Table 8. Keyboard shortcuts (continued)

Name	Keyboard Shortcut	Description
Next tab	Ctrl+]]	Goes to the next tab.
Close and exit	Ctrl+F4	Closes the controller project and FactoryTalk Design Workbench.
Close current tab	Ctrl+Shift+F4	Closes the current tab.
Show tab list	Ctrl+F10	Shows a list of opened tabs to view or close.
Close project	Alt+F4	Closes the current project and returns to the welcome page.
View properties	Alt+Enter	Shows the properties of the selected element in the Properties pane.
Highlight Toolbox	Alt+Insert	Opens and highlights Toolbox . Alt+Insert has the same effect as Insert .
Show/hide FBD grid	Shift+*	Displays or hides the grids in the FBD editor.
Open Device menu	Alt+D	Opens the Device menu from the menu bar.
Open Edit menu	Alt+E	Opens the Edit menu from the menu bar.
Open File menu	Alt+F	Opens the File menu from the menu bar.
Go to rung	Alt+G	Opens Go to Rung to enter a rung number to go to.
Open Help menu	Alt+H	Opens the Help menu from the menu bar.
Open View menu	Alt+V	Opens the View menu from the menu bar.
Open Window menu	Alt+W	Opens the Window menu from the menu bar.
Open Help	F1	Opens the context-sensitive Help of FactoryTalk Design Workbench.
Edit	F2	Edit the content at the current location.
Delete	Delete	<ul style="list-style-type: none"> Deletes the selected element in the LD or FBD editor. Deletes the selected content or the content to the right of the cursor in the ST editor.

Pane bar

Use the pane bar to display or hide panes quickly. The following table introduces the button toggles on the pane bar.

Table 9. Items on the pane bar





Item	Name	Description	Keyboard Shortcut
	Left panes	Displays or hides the panes that are last pinned to the left side of the user interface (UI).	Ctrl+Shift+L
	Project Organizer	Displays or hides Project Organizer .	Ctrl+Shift+P
	Right panes	Displays or hides the panes that are last pinned to the right side of the UI.	Ctrl+Shift+R
	Toolbox	Displays or hides Toolbox .	Insert or Alt+Insert

Table 9. Items on the pane bar (continued)

Item	Name	Description	Keyboard Shortcut
	Properties	Displays or hides Properties .	Alt+Enter
	Connection	Displays or hides Connection .	Ctrl+Alt+C
	Bottom panes	Displays or hides the panes that are last pinned at the bottom of the UI.	Ctrl+Shift+B
	Output	Displays or hides Output .	<ul style="list-style-type: none"> • Ctrl+Alt+S Displays the Console page • Ctrl+Alt+E Displays the Error list page • Ctrl+Alt+L Displays the Log page

Output pane

The **Output** contains the following pages that provide information about the project or the currently opened page in the workspace:

- **Log**
- **Error list**
- **Console**

When opening the **Output** pane, the **Log** page displays by default unless an error occurs, in which case the **Error list** page is shown.

Use the **View** menu or the [pane bar on page 23](#) to open **Output**.

Console page

Figure 5. Console page

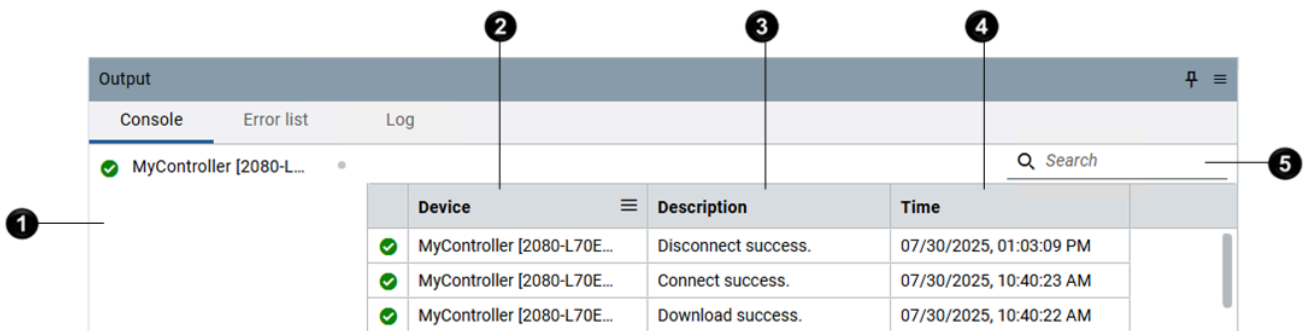



Table 10. Items on the Console page

Item	Name	Description
	Status overview pane	Lists the added devices.

Table 10. Items on the Console page (continued)

Item	Name	Description
		Select a device from the list to filter the log entries by device. When a device is selected,  displays in the Device column, indicating that the column is filtered.
2	Device	Lists the devices that are associated with the generated information.
3	Description	Displays the logs about the added devices
4	Time	Displays the time stamps of the logs.
5	Search box	Enters the device name and log description to find and filter the logs.

Error list page

Figure 6. Error list page

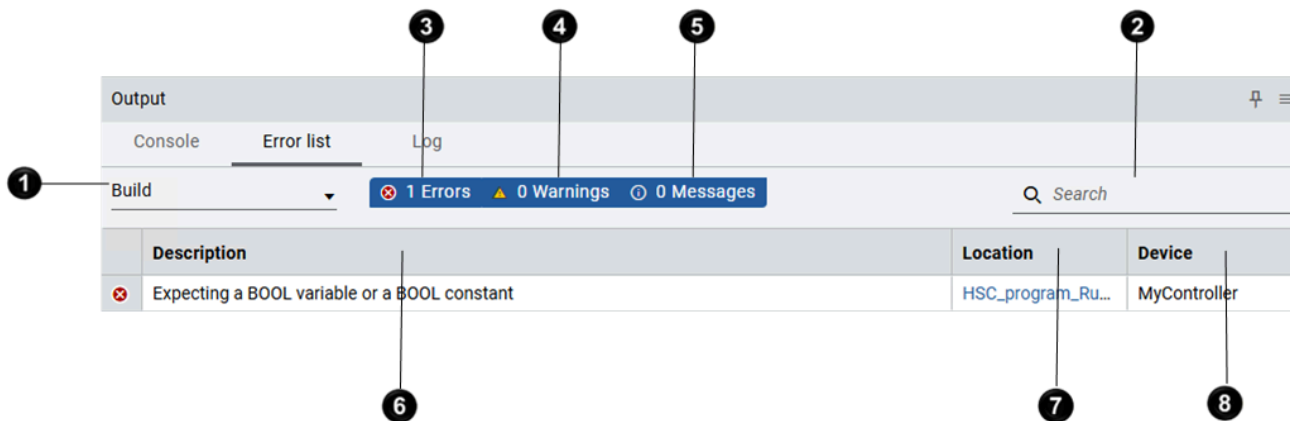


Table 11. Items on the Error list page

Item	Name	Description
1	Scope list	Selects a scope to filter error messages by specific context or category. The scope is General by default.
2	Search box	Enters the corresponding description, location, or device to find error messages that match.
3	Errors button	Displays or hides the errors.
4	Warnings button	Displays or hides the warnings.

Table 11. Items on the Error list page (continued)

Item	Name	Description
5	Messages button	Displays or hides the information messages. By default, errors, warnings, and messages are displayed on the Error list page. When there are errors, only errors are displayed.
6	Description	Displays the message content.
7	Location	Displays the location of errors and their links.
8	Device	Displays the device name in which the error occurred. You can select ☰ in the column header to search or filter errors by device.

Log page

Figure 7. Log page



Table 12. Items on the Log page

Item	Name	Description
1	Scope list	Selects a scope to filter log messages by specific context or category. The scope is General by default.
2	Clear log button	Clears the log entries about the selected scope.
3	Log entries	Displays the log entries in this controller project.

Properties pane

Use the **Properties** pane to view and edit the properties of items in **Project Organizer** or language editors. Properties that appear dimmed are read-only.

The content that displays in the **Properties** pane varies with the item that you selected.

Table 13. Items in the Properties pane

Selected Item	Content displayed in Properties
Controller	<ul style="list-style-type: none"> • Controller name • Controller catalog number
Program Organization Unit (POU)	<ul style="list-style-type: none"> • POU name • POU description
Instruction	<ul style="list-style-type: none"> • Instruction name • Instruction description • EN/ENO • Input Number of the input operands for the selected instructions, such as the ADD instruction

Controller application workspace

The controller application workspace is the main area where you view, configure, monitor, and troubleshoot Micro800 controllers in a FactoryTalk Design Workbench project.

The controller application workspace contains the **Controller** tree and controller configuration pages.

Figure 8. Controller tree and the General configuration page of an Micro870 L70E controller



Controller tree

The **Controller** tree lists the available configurations for a Micro800 controller.

The nodes in the **Controller** tree vary with the controller type. The following table lists the nodes in the **Controller** tree.

Table 14. Items in the Controller tree

Node	Available for
General	All
Memory	All
Startup/Fault	All
Serial Port	Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers
USB Port	Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers
Ethernet	Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers

Table 14. Items in the Controller tree (continued)

Node	Available for
Modbus Mapping	Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers
MicroLogix Mapping	Micro850 L50E and Micro870 L70E controllers
Real Time Clock	All
Motion	2080-L50E-xxQBB, 2080-L50E-xxQVB, and 2080-L70E-xxQBB(N) controllers
Remote LCD	Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers
LCD Module	Micro810 LC10 controllers
Data Log	Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers <ul style="list-style-type: none"> When an SD card is added, the data logs displayed after selecting Manage data log are loaded from the SD card. When an SD card is not added, the data logs displayed after selecting Manage data log are loaded from the computer.
Recipe	Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers The Manage recipe function is available when an SD card is added to the controller.

Controller configuration page

The content on the controller configuration pages varies with the controller type and the tree node that you selected. Use the controller configuration page to:

- View and configure the relevant settings of the current controller.
- Resize the page.

Online center

The **Online center** contains information on the added devices and the options to control the devices and configure the online settings:

- The name, catalog number, and connection path of the added devices.
- The **Download** and **Upload** button.
- The messages about the download and upload progress.
- A link to the [Online settings on page 30](#) page.

Use the **Online center** to:

- Download device configurations.
- Upload device configurations.
- View the download and upload progress.
- Change the device connection path.
- Open **Online settings** to view and change the [online settings on page 30](#).

Figure 9. Online center

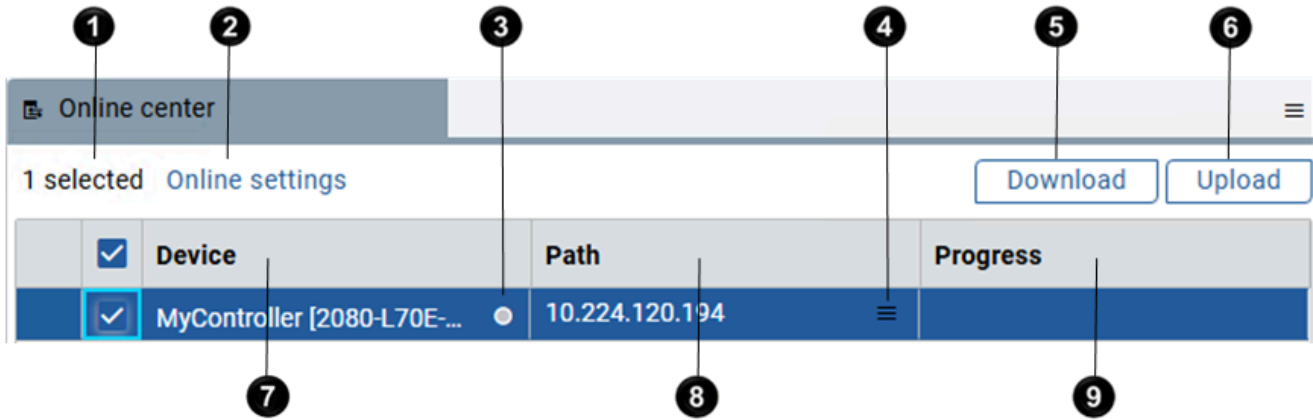



Table 15. Items in the Online center

Item	Description
1	Shows how many devices are selected in Online center .
2	Opens the Online settings on page 30 page to configure relevant settings.
3	Indicates the device connection status. <ul style="list-style-type: none"> Gray: The device is offline. Green: The device is online. Red: The device is offline and has faults.
4	Opens the Select device dialog to select a device to connect to.
5	Opens the Download device dialog to download the selected device configurations to the device.
6	Opens the Upload device dialog to upload the selected device configurations to FactoryTalk Design Workbench for editing.
7	Lists the added devices with the device names and catalog numbers.
8	Displays the device connection paths. To change the connection path, do one of the following: <ul style="list-style-type: none"> Double-click the Path box, and then enter the path. Select , and then in Select device, enter a path.
9	Displays the status and time stamps of the actions that are performed on the added devices.

Online settings

Use the **Online settings** dialog to configure the controller's settings for communication, downloading, uploading, and device discovery.

Table 16. Controller online settings

Category	Setting	Description
General	Use Cellular Network	Determines whether to use the cellular network for downloading and uploading. The default setting is No .
Download	Mode	Sets the controller mode when downloading is completed. The default setting is Remote program .
	Connection	Determines whether to connect to the controller when downloading is completed. The default setting is Connect .
	Project value	Determines whether to download the values of the variables. The default setting is No .
	Warning	Determines whether to download the controller project that contains build errors. The default setting is Yes .
	Communication	Determines whether to download the project when the driver settings for Ethernet, USB, or DF1 differ in the controller and the project. The controller cannot automatically establish a connection if you download mismatched configurations between the controller and the project. The default setting for Download Ethernet is No . The setting is applicable when using the same driver to download the corresponding driver settings for Ethernet, USB, or DF1. The default setting for Download Ethernet , Download USB , or Download Serial Port is No .
Upload	Connection	Determines whether to connect to the controller when uploading is completed. The default setting is Connect .
	Project value	Determines whether to upload the values of variables. The default setting is No .
Discover	Discover	Determines whether to connect to the controller and initiate debugging when discovery is completed. The default setting is Connect .

Projects

A FactoryTalk Design Workbench project contains different components that you use to design, configure, program, visualize, and maintain your devices.

A FactoryTalk Design Workbench project contains the following key elements:

- Controller programs with the main logic for the project, executing a set of instructions
- Configuration settings for controllers, drives, I/Os, and project-specific settings
- Variables that are both global and local
- User-defined data types, including arrays, structures, and defined words

Device and project compatibility

FactoryTalk Design Workbench supports devices from different releases. To use projects or devices from different releases, beware of the compatibility between releases.

Table 17. Compatibility between a FactoryTalk Design Workbench release and Micro800 controller firmware revision

FactoryTalk Design Workbench Release	Micro800 Firmware Revision
1.01.00	<ul style="list-style-type: none"> • Micro810 LC10 controllers: 7.x • Micro820 L20E, Micro850 L50E, or Micro870 L70E controllers: 23.xx and 22.xx
1.00.00	<ul style="list-style-type: none"> • Micro810 LC10 controllers: 7.x • Micro820 L20E, Micro850 L50E, or Micro870 L70E controllers: 23.xx and 22.xx

Table 18. Compatibility mechanism for earlier software versions

	Controller Project Version		Controller Firmware Revision
Create	R23.00 or R22.00 for Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers	Project can be downloaded into	R23.00 or R22.00 for Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers
	R7.00 for Micro810 LC10 controllers		R7.00 for Micro810 LC10 controllers
Import	R23.00 or R22.00 for Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers	Project can be downloaded into	R23.00 or R22.00 for Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers
	R7.00 for Micro810 LC10 controllers		R7.00 for Micro810 LC10 controllers
Upload or discover	R23.00 or R22.00 for Micro820 L20E, Micro850	Project can be downloaded into	R23.00 or R22.00 for Micro820 L20E, Micro850

Table 18. Compatibility mechanism for earlier software versions (continued)

	Controller Project Version		Controller Firmware Revision
	L50E, and Micro870 L70E controllers		L50E, and Micro870 L70E controllers
	R7.00 for Micro810 LC10 controllers		R7.00 for Micro810 LC10 controllers

Create a project

Use the **Welcome** page or the **File** menu to create a project. Upon project creation, a directory structure is generated, and the project is saved at the Documents folder.

Project naming conventions are as follows:

- The project name can contain up to 40 characters.
- The project name is restricted to alphanumeric characters and underscores.
- The project name cannot end with an underscore or contain consecutive underscores.
- The project name cannot be reserved words for Windows or Disk Operating System, such as "nul", "aux", "con", "com1", and "lpt1".

Prerequisites

- Close any open projects.

To create a project

1. Open **New project** by doing one of the following:
 - From the **Welcome** page, select **New project**.
 - From the menu bar, select **File > New project**.
2. In **New project**, enter the project name, and then select **Create**.



Because **Add device on creation** is selected by default, **Add device** opens after project creation. If you do not want to add a device when creating the project, clear **Add device on creation**.

3. (optional) Add a device to the project:
 - a. In **Add device**, select a controller.
 - b. (optional) In **Selection**, enter the device name, select the device revision, view the device product lifecycle information, and then select **Add**.

Manage devices

Use **Project Organizer** to add, rename, or delete devices in FactoryTalk Design Workbench.

Controllers catalog

Every controller in the catalog has a unique catalog ID that is related to its device properties and specifications.

Table 19. Micro800 controller specifications

Catalog Number	Digital I/O points	Number of Inputs and Type	Number of Outputs and Type	Input Power
Micro810 [®] LC10 controllers				
2080-LC10-12AWA	12	8-120/240 V AC	4-Relay	120/240 V AC
2080-LC10-12DWD	12	8 total 4-12 V DC 4-A IN 0-10 V	4-Relay	12 V DC
2080-LC10-12QBB	12	8 total 4-24 V DC 4-A IN 0-10 V	4-Source	24 V DC
2080-LC10-12QWB	12	8 total 4-24 V DC 4-A IN 0-10 V	4-Relay	24 V DC
Micro820 [®] L20E controllers				
2080-L20E-20AWB(R)	20	12 total 8-120 V AC 4-Analog	7-Relay 1-Analog	120 V AC
2080-L20E-20QBB(R)	20	12 total 8-24 V DC Sink/Source 4-DC/Analog	7-DC FET 1-Analog	24 V DC
2080-L20E-20QWB(R)	20	12 total 8-24 V DC Sink/Source 4-DC/Analog	7-Relay 1-Analog	24 V DC
Micro850 [®] L50E controllers				
2080-L50E-24AWB Supports DF1	24	14-24 V DC/V AC	10-Relay	24 V DC
2080-L50E-24QBB Supports DF1	24	14-24 V DC/V AC	10-Source	24 V DC

Table 19. Micro800 controller specifications (continued)

Catalog Number	Digital I/O points	Number of Inputs and Type	Number of Outputs and Type	Input Power
2080-L50E-24QVB Supports DF1	24	14-24 V DC/V AC	10-Sink	24 V DC
2080-L50E-24QWB Supports DF1	24	14-24 V DC/V AC	10-Relay	24 V DC
2080-L50E-48AWB Supports DF1	48	28-110 V AC	20-Relay	24 V DC
2080-L50E-48QBB Supports DF1	48	28-24 V DC/V AC	20-Source	24 V DC
2080-L50E-48QVB Supports DF1	48	28-24 V DC/V AC	20-Sink	24 V DC
2080-L50E-48QWB Supports DF1	48	28-24 V DC/V AC	20-Relay	24 V DC
Micro870 [®] L70E controllers				
2080-L70E-24AWB Supports DF1	24	14-24 V DC/V AC	10-Relay	24 V DC
2080-L70E-24QBB Supports DF1	24	14-24 V DC/V AC	10-Source	24 V DC
2080-L70E-24QBBN Supports DF1 and DNP3	24	14-24 V DC/V AC	10-Source	24 V DC
2080-L70E-24QWB Supports DF1	24	14-24 V DC/V AC	10-Relay	24 V DC
2080-L70E-24QWBN Supports DF1 and DNP3	24	14-24 V DC/V AC	10-Relay	24 V DC

Add a controller to the project

Use **Add device** to add a controller to a FactoryTalk Design Workbench project. FactoryTalk Design Workbench supports one controller for each project.

After selecting a controller, in the **Selection** pane of the **Add device** dialog, the following controller information is shown:

- Catalog ID
- Controller name for entry

- Controller revision
- [Product lifecycle information on page 126](#)

For more information about devices, see [Controllers catalog on page 34](#).

Prerequisites

- Create a project without adding a controller.

To add a device to a project

1. Open **Add device** by doing one of the following:
 - In **Project Organizer**, right-click a blank space, and then select **New device**.
 - In **Project Organizer**, select the **New device** button.
2. In **Add device**, double-click the controller you want to add or select the checkbox beside the controller.
3. In **Selection**, enter the controller name, select the controller revision, view the controller lifecycle information, and then select **Add**.
After adding a controller, it is recommended that you [secure the controller with a password on page 37](#). Be sure to record and store the password securely, because it cannot be recovered if lost.

Discover and add a controller connected to a computer



Use the discover device function to view and add the devices connected to a computer. You can use the function to:

- Replace the existing controller in the controller project with the discovered one.
- Add a controller and its controller project to an empty project.

Prerequisites

- The controller is offline.

To discover and add a controller connected to a computer

1. Open **Select device** by doing one of the following:
 - In **Project Organizer**, select .
 - In **Project Organizer**, select **Discover device**.
 - In **Project Organizer**, right-click a blank space, and then select **Discover device**.
 - Open [Add device on page 35](#), and then select **Discover device**.
2. If connecting to the device via Ethernet, in **Select device**, select , and then in **Configure Driver**, select **Add New**, enter the IP address of the device, and select **OK**.
3. In **Configure Driver**, select **Close**.
4. In **Select device**, select the node that contains the device to refresh the device list.
5. Select the device, and then select **OK**.
The device is added to the project and **Project Organizer**. An upload is then performed automatically, and the controller will be connected.
6. (optional) If you've already added a controller to the project and are selecting another controller from **Select device**, **Discover controller** opens, asking if you want to replace the existing controller with the selected one. Select **Replace**.

Secure a controller with a password

To prevent unauthorized access to the configuration and programming in a controller, it is recommended that you secure the controller with a password.

When a controller is configured with a password, you must enter the password before doing any of the following:


- [Connect to the controller on page 37.](#)
- [Change the controller password on page 38.](#)
- [Clear the controller password on page 38.](#)

IMPORTANT: If the existing password is forgotten or lost, it cannot be recovered, changed, or cleared. If a controller fault occurs during password configuration, restart the controller and reconfigure the password.

Prerequisites

- The controller is in **Remote program** mode.

To secure a controller with a password

1. Open **Change password for <controller-name>** by doing one of the following:
 - In **Connection**, select **Secure**, and then select **Set password**.
 - In **Project Organizer**, right-click the controller, and then select **Secure > Set Password**.
 - In **Project Organizer**, select the controller, and then from the menu bar, select **Device > Secure**.
2. In **Change password for <project-name>**, enter a password. Passwords must be 8 to 64 characters long, which can include letters, numbers, spaces, and any keyboard symbols.
3. In **Confirmation**, enter the exact password again.
4. (optional) To view the password, select .
5. Select **OK**.



Connect to a password-protected controller

You must have the password before connecting to a password-protected controller.

Micro800 controllers allow the connection of one configuration tool at a time. You cannot connect to the controller when:

- The controller is locked by another user.
- The type of the selected controller does not match the one defined in the opened project.

To connect to a password-protected controller

1. From the menu bar, select  to open **Online center**.
2. In **Online center**, do one of the following:
 - In the **Path** column, enter the controller connection path.
 - Select  to open **Select device**, find and select the controller, and then select **OK**.

3. Connect to the controller by doing one of the following:
 - In **Connection**, turn on the toggle beside the Micro800 controller.
 - In **Project Organizer**, right-click the controller, and then select **Connect**.
4. In **Enter password**, enter the controller password, and then select **OK**.

IMPORTANT: The controller will enter the locked-out state after a total of 11 or more unsuccessful password attempts, whether consecutive or spread out over time. In the locked-out state, the following conditions apply:

- A subsequent attempt with the correct password will only succeed after 300 seconds (locked-out duration).
 - If you do not make subsequent attempts, the controller will automatically reset the unsuccessful sign-in count and exit the locked-out state after one day (auto unlock duration).
-

Change the controller password

Before changing the controller password, you must have the current password.

IMPORTANT: If the existing password is forgotten or lost, there is no way to recover or change the password.

We recommend changing the controller password periodically to protect the programs and functions in the controller.


Passwords must be 8 to 64 characters long, which can include letters, numbers, spaces, and any keyboard symbols.

Prerequisites

- The controller is in **Remote program** mode.

To change the controller password

1. Open **Change password for <controller-name>** by doing one of the following:
 - In **Connection**, select **Secure**, and then select **Change password**.
 - In **Project Organizer**, right-click the controller, and then select **Secure > Change Password**.
 - In **Project Organizer**, select the controller, and then from the menu bar, select **Device > Secure > Change password**.
2. In **Change password for <controller-name>**, do the following:
 - a. Enter your current password.
 - b. Enter your new password.
 - c. Enter your new password again to confirm it.

To view the password, select .
3. Select **OK**.


Clear the controller password

Before clearing the password for a Micro800 controller, you must have the current password.

Prerequisites

- The controller is in **Remote program** mode.

To clear the controller password

1. Open **Clear password for <controller-name>** by doing one of the following:
 - In **Connection**, select **Secure**, and then select **Clear password**.
 - In **Project Organizer**, right-click the controller node, and then select **Secure > Clear Password**.
 - In **Project Organizer**, select the controller, and then from the menu bar, select **Device > Secure > Clear password**.
2. In **Clear password for <controller-name>**, enter the password.
To view the password, select .
3. Select **OK**.

Data protection

Use **Data protection** to create multiple user-defined global variables that retain their values when:

- Restoring a controller project from a memory module (available for Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers).
- Downloading a project, programs, variables, and controller configurations to a Micro800 controller (available for Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers).

In FactoryTalk Design Workbench, the variables that are defined for data protection retain their values if the project stored in the memory module and the current controller project meet the following criteria:

- The variable name, data type, and number of bytes are defined the same in both projects.
- The arrays and array dimensions are defined the same in both projects.
- The variable is configured with data protection in both projects.

Table 20. Information in the data protection grid

Column	Description
No.	Displays the index number of the variable. Deleting a variable causes the indexes of all the subsequent variables to decrement by one. Up to 60 global variables are supported for data protection.
Variable Name	Displays the user-defined variable name.
Data Type	Displays the data type of the global variable. Supported data types are: BOOL, BYTE, DATE, DINT, DWORD, INT, LINT, LREAL, LWORD, REAL, SINT, TIME, UDINT, UINT, ULINT, USINT, WORD, and USER ARRAYS.

Add variables for data protection

Use **Data protection** to select multiple user-defined global variables that retain their values when:

- Restoring a controller project from a memory module (available for Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers).
- Downloading a project, programs, variables, and controller configurations to a Micro800 controller (available for Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers).

Data protection supports up to 60 user-defined variables of the following types:

- Simple data types, excluding STRING
- Arrays with up to three dimensions

To add variables for data protection

1. Open **Data protection** by doing one of the following:
 - In **Project Organizer**, right-click the controller, and then select **Data Protection**.
 - In **Project Organizer**, select the controller, and then from the menu bar, select **Device > Data protection**.
2. In **Data protection**, double-click **Select variable** or select **""**.
3. In the variable selector, find and select the variable, and then select **Select**.
4. In **Data protection**, select **OK**.
To delete a variable from **Data protection**, right-click the variable, and then select **Delete**.

Change the Micro800 controller project

Use **Change controller** to [change the controller type on page 42](#) or update the controller with the latest controller revision.

IMPORTANT: Because Micro800 controllers support different hardware and software functions, changing the controller might require modifications to the following:

- Controller configurations
 - Programs
 - User-defined function blocks
 - User-defined functions
 - Global or local variables
-

Figure 10. Change controller dialog

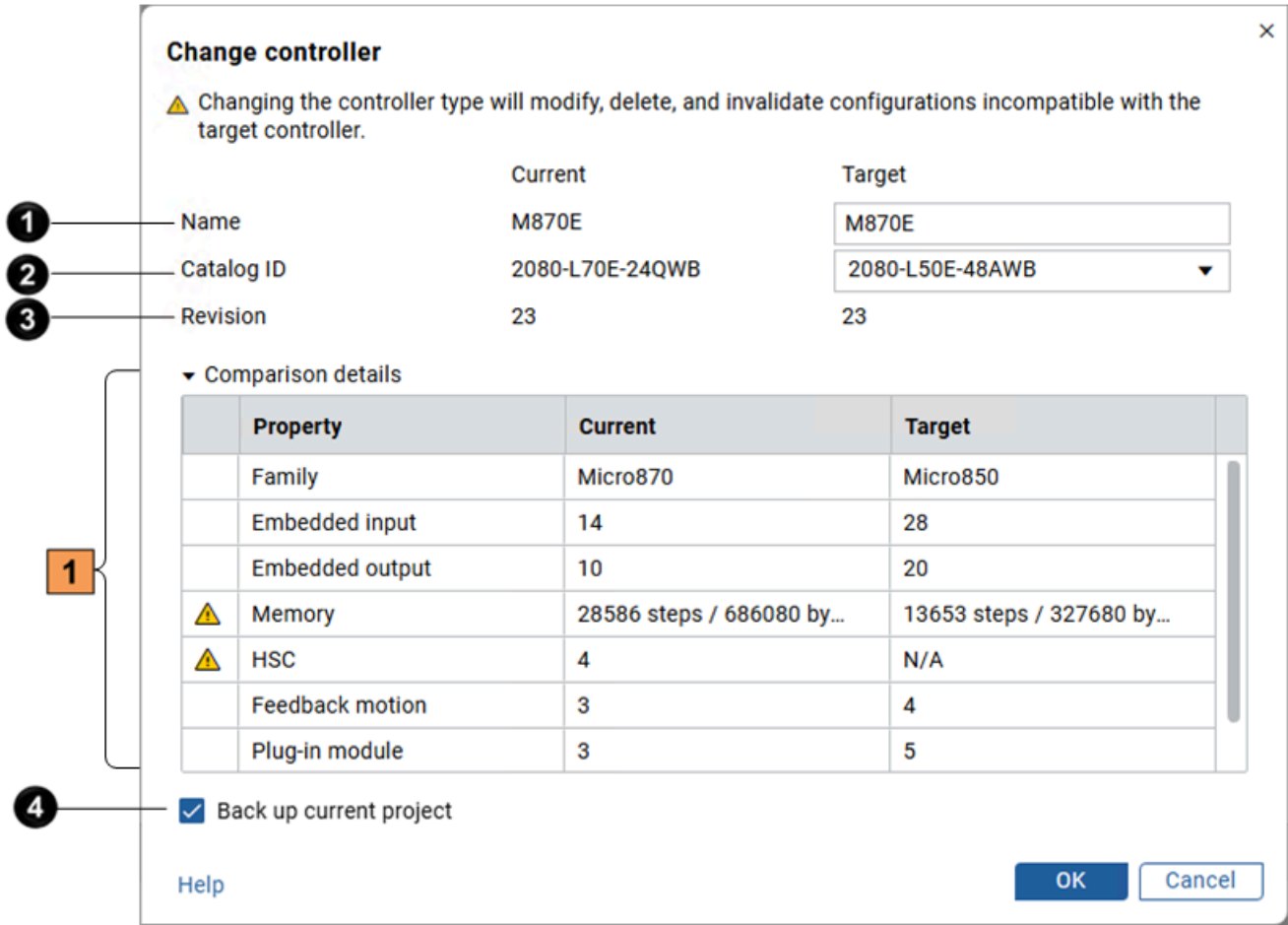







Table 21. Change controller dialog

Item	Name	Current	Target
1	Name	The name of the current Micro800 controller that you entered when adding the controller. The name is read-only.	The name of the target Micro800 controller. The default name of the target controller is the current controller's name. Enter a name for the target controller if needed.
2	Catalog ID	The catalog number of the current Micro800 controller.	The catalog numbers of the controllers that can be selected as the target controller.
3	Revision	The controller revision of the current controller project. The revision is read-only.	The latest controller revision of the selected target controller. The revision is read-only.

Table 21. Change controller dialog (continued)

Item	Name	Current	Target
	Comparison details	<p>A list of properties compared between the current controller and the target controller.</p> <p>Select  to show or hide the comparison details.</p> <p> indicates a reduction in capacity to the physical configuration of the project or a supported function. Hover over  to view the details and recommended actions.</p>	
	Back up current project	<p>A checkbox that determines whether to back up the current controller project. When selected, the current controller project is backed up at the FTDW folder within Documents.</p> <p>The checkbox is selected by default.</p>	

Change the Micro800 controller type

Change the controller type to save the current controller project and create another controller project based on the target controller type. FactoryTalk Design Workbench creates the project with the latest firmware revision of the target controller.

When changing the controller type, the following components are copied to the target controller project:


- Controller configurations
- Programs
- User-defined function blocks
- Global and local variables
- User-defined data types



The default location for saving the controller project and backing up the original controller project is the FTDW folder within Documents.

Prerequisites

- The controller is offline.
- The controller is not a Micro810 LC10 controller.

To change the Micro800 controller type

1. Open **Change controller** by doing one of the following:
 - In **Project Organizer**, select the controller, and then from the **Device** menu, select **Change controller**.
 - In **Project Organizer**, right-click the controller, and then select **Change Controller**.
2. From the **Catalog ID** list, select the target controller.
3. (optional) In the **Name** box, enter the controller name.
4. Select  to show the comparison details between the current and target controllers.

5. (optional) Hover over  to show the potential impacts of changing to the target controller, and then take the recommended actions as needed.
 indicates a reduction in capacity to the physical configuration of the project or a supported function.
6. (optional) Select **Back up current project** to back up the original project to your computer.
7. Select **OK**.

Build a project and view outputs

FactoryTalk Design Workbench builds the entire project before you download the program to the Micro800 controller. After the initial build, FactoryTalk Design Workbench decides whether to rebuild the entire project or only the components that have changed since the last build before downloading.

We recommend that you manually build the project before downloading to find and resolve any errors.

Prerequisites

- Save the project.
- No download operation is in process.


To build a project and view outputs

1. Build the project by doing one of the following:
 - In **Project Organizer**, right-click the controller node, and then select **Build**.
 - In **Project Organizer**, select the controller node, and then from the menu bar, select **Device > Build**.
 - On the controller memory configuration page, select **Build**.
 - In **Project Organizer**, right-click an individual Program Organization Unit, and then select **Verify Program**.
2. In the **Output** pane, on the **Log** page, monitor the building progress and identify build errors. If the project has build errors:
 - a. In the **Output** pane, on the **Error list** page, view and resolve any build errors.
 - b. Clean the build errors by doing one of the following, and then build again:
 - In **Project Organizer**, right-click the controller node, and then select **Clean**.
 - From the menu bar, select **Device > Clean**.

Set up the connection path

Use **Online center** to set up the connection path for the controller by entering its IP address.

To set up the connection path

1. Select  to open **Online center**.
2. In **Online center**, in the **Path** box, enter the IP address of the controller.
3. Select **Enter**.

To clear the connection path, double-click the **Path** box, delete the IP address, and then select **Enter**.

Add a serial-DF1 driver

Configure a serial-DF1 driver to establish RS-232 communication with Micro800 controllers. Each project supports one serial-DF1 driver.

IMPORTANT: Unlike Connected Components Workbench, which automatically adds a DF1 driver after you plug in the DF1 device, FactoryTalk Design Workbench requires you to manually add and configure a DF1 driver before:

- Connecting to a Micro810 LC10 controller.
- Connecting to a Micro820 L20E, Micro850 L50E, or Micro870 L70E controller via a USB device on a remote LCD module.

Prerequisites

- A controller is added to your project.
- CIP serial is set as the driver for the serial port settings.

To add a DF1 driver




1. Select  to open **Online center**.
2. In **Online center**, select .
3. In **Select device**, select .
4. In **Configure Drives**, from the **Available Driver Types** list, select **RS-232 DF1**, and then select **Add New**.
5. In **Configure RS-232 DF1** driver, enter the settings.
 - To connect to a Micro820 L20E, Micro850 L50E, or Micro870 L70E controller, the following settings must be correctly configured:

Table 22. Micro820 L20E, Micro850 L50E, and Micro870 L70E controller DF1 driver configuration

Via a DF1 device	Via a USB device on a remote LCD module
COM Port	COM Port
Baud rate: must be the same as that on the Serial Port configuration page	Error check: CRC
Error check: must be the same as that on the Serial Port configuration page	Baud rate: 38400
Protocol: must be the same as that on the Serial Port configuration page	Parity: None
Station number: must be the same as the Station address on the Serial Port configuration page	Station address: 1
Parity: must be the same as that on the Serial Port configuration page	

- To connect to a Micro810 LC10 device, the following settings must be correctly configured:

Table 23. Micro810 LC10 DF1 driver configuration

Setting	Value
COM Port	Specify the port number.
Baud rate	57600
Error check	CRC
Protocol	Full Duplex
Station number	1

6. Select **Test configuration** to test the settings. Before testing, ensure that your DF1 device is connected.

If the test fails, possible reasons include:

- The COM port is used by RSLinx® Classic or FactoryTalk® Linx.
- CIP serial is not set as the driver.
- The driver configuration does not match the connected driver.

Adjust the settings to resolve the configuration issues.

7. Select **OK**.

Serial-DF1 properties

The **Configure RS-232 DF1 driver** dialog contains the following information.

Table 24. Serial-DF1 properties

Setting	Description
Driver name	Specifies the name of the serial driver. If changed, the name must be unique across the entire topology and cannot exceed 255 characters.
COM Port	Lists the available COM ports on your computer. The driver transmits data to a device connected to the serial (COM) port of a computer. Select a port from the list. Available port numbers are from 0 through 31 (default is 1).
Device	Shows the DF1 device that is connected with your computer.
Baud rate	Specifies the baud rate for the driver that transmits data to your device. Available baud rates are 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200. The default is 19200.
Station number	Specifies the station number for the driver. Available station numbers are from 0 through 254 decimal (default is 0).
Parity	Matches the error-checking method for the connected driver (None, Even, or Odd).
Error check	Matches the setting of the device you are connected to (BCC or CRC). All devices on the network must be configured the same.
Stop bits	Specifies the amount of time between data transmissions. Basic DF1 protocol generally uses 1 (default is 1).
Protocol	Specifies the supported protocols (Full Duplex or Half Duplex).
Ack/poll timeout	Specifies the ACK value. The default is 3000 milliseconds.

Table 24. Serial-DF1 properties (continued)

Setting	Description
Maximum retries	Specifies a value for the maximum number of times the system should retry. The default is 3.
Test configuration	<p>Tests if the driver configuration is correct. Before testing, ensure that your DF1 device is connected.</p> <p>If the test fails, possible reasons include:</p> <ul style="list-style-type: none"> • The COM port is in use by RSLinx Classic or FactoryTalk® Linx. • CIP serial is not set as the driver. • The driver configuration does not match the connected driver.




Connect to a Micro800 controller

After adding a Micro800 controller to your project, connect to the controller, and then proceed to diagnose or secure the controller.

If no Micro800 controller exists in your project, use [Discover and add a controller connected to a computer on page 36](#) to discover and connect to a Micro800 controller.

IMPORTANT: When your computer is connected to a Micro800 controller, other users who are trying to connect to the controller will be informed that the controller is in use. The notification will display your computer name and username.

To connect to a Micro800 controller

1. Open **Online center** by doing one of the following:
 - Select .
 - In **Project Organizer**, right-click the controller node, and then select **Download**.
2. In **Online center**, select .  is available when the controller is disconnected.
3. In **Select device**, find and select the Micro800 controller, and then select **OK**.



- Selecting a controller with a different catalog number than the controller defined in the project will lead to a connection failure.
- Once selected, the connection path is displayed in **Online center**. [You can also manually set up the connection path on page 43](#).

4. Connect to the controller by doing one of the following:
 - In **Connection**, turn on the toggle beside the controller.
 - In **Project Organizer**, right-click the controller node, and then select **Connect**.
5. (optional) In **Enter password**, enter the controller password, and then select **OK**.
6. (optional) In **Connect controller**, select one of the following:
 - **Download current project to the controller**
 - **Upload current project from the controller**

Connect controller opens when the content in the current project does not match the content in the controller.

IMPORTANT: We recommend disconnecting from the controller when not in use to prevent connection loss due to the computer entering sleep mode.

Download a controller configuration to a controller

Use **Download device** to download a project from FactoryTalk Design Workbench to a Micro800 controller for running and testing. The project can contain programs, variables, and controller configurations.

When downloading the project, if the controller revision doesn't match the one defined in the FactoryTalk Design Workbench project, the **Update and download** dialog opens. Use the dialog to update the controller revision, and then download the project again.

When the download begins, FactoryTalk Design Workbench performs the following tasks:

- Builds any unbuilt programs.
- Compares the configuration of the connected controller with the project configuration.
- Downloads the project, programs, variables, and controller configuration to the controller.

If an error occurs when performing any of the tasks, the download is canceled, and the controller remains unchanged.

If the initial value for the variable is set to the default value for the data type, such as 0, 0.0, D#1970-01-01, FactoryTalk Design Workbench initializes the variable with its value.




When initializing variables with project values, the following conditions apply:

- Variables with a **Direction** of **VarInput** cannot be downloaded.
- Variables with an **Attribute** of **Read** or **No Access** cannot be downloaded.
- Complex variables, such as structures and arrays that contain members with initial values, cannot be downloaded.
- Variables with both a value and an initial value will be initialized with their initial value.

Prerequisites

- Set the connection path for the controller.
- The catalog number of the controller in the current project matches the one you're downloading to.

To download a controller configuration to a controller

1. Select  to open **Online center**.
2. From the **Device** list, select the controller, and then select **Download**.
3. In **Download device**, select **OK**.
4. Use the **Output** pane to view the result.
5. (optional) Resolve any errors, and then download the configuration again.


Upload a controller configuration from a controller

Use **Upload device** to upload a project containing programs, variables, and controller configurations from a Micro800 controller to FactoryTalk Design Workbench. All project programs, variables, and controller configurations are overwritten after an upload. Once the project is uploaded to FactoryTalk Design Workbench, you can edit the controller project, and then download it back to the controller.

Prerequisites

- Set the connection path for the controller.
- The catalog number of the controller in the current project matches the one you're uploading from.

To upload a controller configuration from a controller

1. Select  to open **Online center**.
2. In **Online center**, from the **Device** list, select the controller, and then select **Upload**.
3. In **Upload device**, select **OK**.



COP instruction blocks do not upload values for Src and Dest variables. Other instruction blocks and variables are not affected.

4. Use the **Output** pane to view the result.
5. If there are errors, resolve them, and then upload again.

Monitor local and global variables

Open the local or global variable grids to monitor the variable values in real time.

Each variable grid displays the values and force status of the variables in their respective columns. When the output value is not available, text appears in the output label.

For graphical programs and function blocks, the appearance of values vary with their type:

- Boolean data type output values are displayed in either black or gray, regardless of whether the value is True or False. A black output value indicates that it can be modified, and a gray output value indicates that it cannot be changed.
- Output values of the SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, LINT, ULINT, LWORD, REAL, LREAL, TIME, DATE, and STRING data types appear as a number or text value in the element. When the output is a structure type, the value appears as the selected member.

Prerequisites

- The controller is in **Remote run** mode.

To monitor local and global variables

1. Open the variable grid by doing one of the following:
 - To open the local variable grid, in **Project Organizer**, go to **Programs > <program-name>**, and then double-click **Local Variables**.
 - To open the global variable grid, in **Project Organizer**, double-click **Global Variables**.
2. Monitor the values.



- The background color of a cell shows its status: white for modifiable; gray for read-only.
- The **Value** of user-defined functions and user-defined function blocks are read-only.
- For Program Organization Units whose **Value** is modifiable, you can also modify the value in the language editor.


View the controller mode

When connected to a Micro800 controller, use **Connection** to view the controller mode.

Prerequisites

- The controller is online.

To view the controller mode

In **Connection**, select  to expand the controller information. Available controller modes are as follows:

- **Program:** The controller is in the non-running mode and can be programmed. The mode cannot be changed remotely.
- **Remote program:** The controller is in the non-running mode and can be programmed. The mode can be changed remotely.
- **Run:** The controller is in the running mode. The mode cannot be changed remotely.
- **Remote run:** The controller is in the remote running mode, displaying the connected values from the controller in FactoryTalk Design Workbench. The mode can be changed remotely.

Change the Micro800 controller Remote mode

When connected to a Micro800 controller, use **Connection** to change from **Remote** to **Program** or **Run** mode.



- For Micro850 L50E controllers, the toggle on the controller must be in the **Remote** center position to make the **Remote program** and **Remote run** options available.
- Micro810 LC10 and Micro820 L20E controllers do not have a toggle and are always in **Remote** mode.

Prerequisites

- The controller is online.

To change the Micro800 controller Remote mode

1. In **Connection**, select .
2. From the **Controller mode** list, select **Program** or **Run**.


View the controller fault status

When connected to the Micro800 controller, use **Connection** to view the controller fault status.

Prerequisites

- The controller is online.

To view the controller fault status

1. (optional) If **Connection** is hidden, open it by doing one of the following:
 - From the **View** menu, select **Connection**.
 - Select  to open the **Online center**.
2. In **Connection**, beside the controller name and catalog number, check the color of the toggle. If a fault occurs, the toggle turns red with an error icon.

Monitor and diagnose a Micro800 controller

FactoryTalk Design Workbench supports diagnostics for network connectivity, controller startup settings, controller faults, and controller projects stored in memory. Controller diagnostics are available when the controller is online.

To monitor and diagnose a Micro800 controller, use the **Device** menu, the **Diagnose** button in **Connection**, or the following controller configuration pages:

- Startup (Micro810 LC10 controllers)
- Startup/Faults
- Serial Port
- Ethernet
- Memory Card
- LCD Module (Micro810 LC10 controllers)
- Plug-in Modules - 2080-SERIALISOL
- Plug-in Modules - 2080-MEMBAK-RTC2
- Plug-in Modules - 2080-DNET20

Open the diagnostics pages


Use the **Device** menu, the **Connection** pane, or one of the controller configuration pages to open the controller diagnostics pages.

Prerequisites

- The controller is online.

To open the diagnostics pages

Do one of the following:

- In **Connection**, select  to expand the controller information, and then select **Diagnose**.
- In **Project Organizer**, select the controller, and then from the **Device** menu, select **Diagnose**.
- From one of the following pages, select **Diagnose**:
 - **Startup** (Micro810 LC10 controllers)
 - **Startup/Faults**

- **Serial Port**
- **Ethernet**
- **Memory Card**
- **LCD Module** (Micro810 LC10 controllers)
- **Plug-in Modules - 2080-SERIALISOL**
- **Plug-in Modules - 2080-MEMBAK-RTC2**
- **Plug-in Modules - 2080-DNET20**

Disconnect from a Micro800 controller

Disconnect from a controller to:

- Configure the Micro800 controller properties.
- [Add plug-in modules on page 90.](#)
- [Update programs on page 80.](#)

To disconnect from a Micro800 controller

Do one of the following:

- In **Project Organizer**, right-click the controller, and then select **Disconnect**.
- In **Connection**, turn off the toggle beside the controller.

Fault diagnostics

Use the **<controller-name>: Fault** view to:

- View information on any faults that occurred.
- Retrieve a detailed fault log or core dump file from the controller.
- Clear a fault.

The **<controller-name>: Fault** view shows the last seven recoverable and the last seven non-recoverable faults with:

- The fault code
- Fault location and the link if available
- A description of the fault

The **<controller-name>: Fault** view is available by selecting the **Fault** button from the **Connection** pane when the controller is in **Remote run** mode.

Figure 11. Controller Fault view

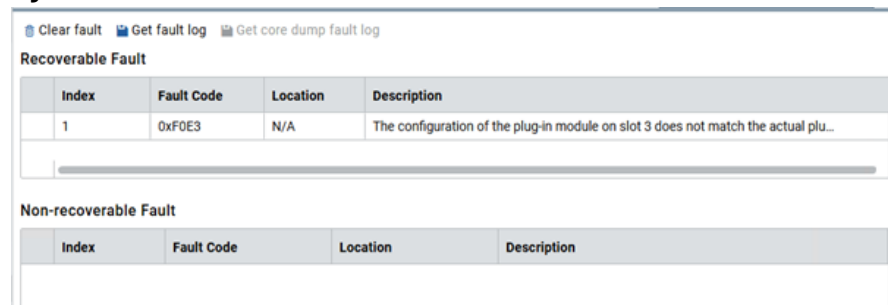


Table 25. Fault Diagnostics view items and descriptions

Item	Description
Clear fault	Deletes the faults and fault logs from the controller.
Get fault log	Retrieves the detailed faults and fault logs from the controller. The information can be used when seeking technical support. By default, the fault log is saved as a TXT file in the Downloads folder. You can customize the path as needed.
Get core dump fault log	Retrieves the core dump of Micro800 controller to a Bin file, which you can send to the Rockwell Automation engineering team for troubleshooting and debugging. By default, the core dump is stored with the name <i>Controller_Core_Dump_mm-dd-yyyy_hh_min_sec.bin</i> in the Downloads folder. You can customize the path as needed. The size limit for the core dump is 8,192 bytes. Upgrading the firmware clears the core dump.
Recoverable/Non-recoverable Fault	Displays two types of faults: <ul style="list-style-type: none"> Recoverable faults are the faults that can be cleared without having to power cycle the controller. Non-recoverable faults are the faults that require the controller to be power cycled before clearing the fault. Up to the last seven faults are recorded for each type of fault with the latest fault shown at the top. If more than seven faults occur, the oldest faults are overwritten.
Index	Displays the sequential number of the faults. The index number 1 indicates the last fault.
Fault Code	Displays the fault code in hexadecimal.
Location	Displays the fault location and provides a link to the program, if available.
Description	Displays the information about the fault.

Fault codes and corrective actions

Refer to the following table when clearing a Micro800 controller fault.

Before implementing any of the suggested corrective actions, complete the following steps:

- Clear all faults before downloading to the Micro800 controller or upgrading the firmware.
- Turn off the controller before adding or removing plug-in modules.
- Turn off the Micro800 controller if the red light-emitting diode (LED) is solid red, and then turn on the controller. If the LED is blinking red, it is not necessary to cycle power.

After taking the corrective action, if the fault persists, contact your Rockwell Automation technical support at <http://rockwellautomation.com/support>.

The extended fault code comes from the I/O module itself, not from the controller. Refer to the I/O module documentation for an explanation of the extended status code.

Table 26. Fault codes, descriptions, and corrective actions for a Micro800 controller fault

Fault Code	Description	Corrective Action
0xF000	<p>The controller was unexpectedly reset due to a noisy environment or an internal hardware failure.</p> <ul style="list-style-type: none"> A Micro800 controller of revision 2.xx attempts to save the program and clear the user data. If the system variable <code>_SYSVA_USER_DATA_LOST</code> is set, the controller is able to recover the user program but the user data is cleared. If not, the Micro800 controller program is cleared. A Micro800 of revision 1.xx clears the program. The system variable <code>_SYSVA_USER_DATA_LOST</code> is not available on Micro800 revision 1.xx controllers. 	<p>Do one of the following:</p> <ul style="list-style-type: none"> Download the program. Refer to the "Wire your controller" section in your controller's user manual: <ul style="list-style-type: none"> Micro810 Programmable Controller User Manual - 2080-UM001A-EN-E Micro820 Programmable Controller User Manual - 2080-UM005_-EN-E Micro830, Micro850, and Micro870 Programmable Controller User Manual - 2080-UM002_-EN-E
0xF01A	<p>The controller was unexpectedly reset due to uncommitted changes during Run Mode Change only.</p> <p>A Micro800 controller of revision 8.xx attempts to save the program and clear the user data. If the system variable <code>_SYSVA_USER_DATA_LOST</code> is set, the controller can recover the user program but the user data is cleared. If not, the Micro800 controller program is cleared.</p>	<p>Do one of the following:</p> <ul style="list-style-type: none"> Download the program. Refer to the "Wire your controller" section in your controller's user manual: <ul style="list-style-type: none"> Micro810 Programmable Controller User Manual - 2080-UM001A-EN-E Micro820 Programmable Controller User Manual - 2080-UM005_-EN-E Micro830, Micro850, and Micro870 Programmable Controller User Manual - 2080-UM002_-EN-E
0xF001	<p>The controller program has been cleared. This happened because:</p> <ul style="list-style-type: none"> A power-down occurred during program download or data transfer from the memory module. The cable was removed from the controller during program download. The RAM integrity test failed. 	<p>Do one of the following:</p> <ul style="list-style-type: none"> Download the program. Transfer the program using the memory module restore utility.
0xF002	<p>The controller hardware watchdog was activated. The controller hardware watchdog timeout happens if the program scan is approximately 3 seconds.</p> <ul style="list-style-type: none"> A Micro800 controller of revision 2.xxx and later attempts to save the program and clear the user data. If the system variable <code>_SYSVA_USER_DATA_LOST</code> is set, the controller is able to recover 	<p>Do the following:</p> <ol style="list-style-type: none"> Connect to the controller. Download the program.

Table 26. Fault codes, descriptions, and corrective actions for a Micro800 controller fault (continued)

Fault Code	Description	Corrective Action
	<p>the user program but the user data is cleared.</p> <ul style="list-style-type: none"> A Micro800 controller revision 1.xx clears the program. The system variable _SYSVA_USER_DATA_LOST is not available on Micro800 revision 1.xx controllers. 	
0xD00F	A particular hardware type, for example, embedded I/O, was selected in the user program configuration, but did not match the actual hardware base.	<p>Do one of the following:</p> <ul style="list-style-type: none"> Connect to the hardware that is specified in the user program. Reconfigure the program to match the target hardware type.
0xF003	<p>One of the following occurred:</p> <ul style="list-style-type: none"> There is a fault in the memory module. There is a connection fault for the memory module. The memory module was incompatible with the controller's firmware revision. 	<p>Do one of the following:</p> <ul style="list-style-type: none"> Remove the memory module and plug it in again. Obtain a new memory module. Upgrade the controller's firmware revision to be compatible with the memory module. <p>For more information on firmware revision compatibility, go to the Rockwell Automation Support site and search for "Firmware Updates".</p>
0xF004	<p>One of the following occurred:</p> <ul style="list-style-type: none"> There is a data transfer error. There is an embedded RTC data error (Micro820 L20E only). There is no user program inside of the controller. The memory is full. 	<ul style="list-style-type: none"> For memory module failure, attempt the data transfer again. If the error persists, replace the memory module. For embedded RTC failure, restart the controller. For the MicroSD card, use another MicroSD card or delete files from the card.
0xF005	The user program failed an integrity check while the Micro800 controller was in Run mode.	<p>Do one of the following:</p> <ul style="list-style-type: none"> Turn off your controller, and then download your program and start your system. Refer to the "Wire your controller" section in your controller's user manual: <ul style="list-style-type: none"> Micro810 Programmable Controller User Manual - 2080-UM001F-EN-E Micro820 Programmable Controller User Manual - 2080-UM005_-EN-E Micro830, Micro850, and Micro870 Programmable Controllers User Manual - 2080-UM002_-EN-E

Table 26. Fault codes, descriptions, and corrective actions for a Micro800 controller fault (continued)

Fault Code	Description	Corrective Action
0xF006	The user program is incompatible with the Micro800 controller firmware revision.	Upgrade the controller firmware revision using ControlFLASH. For more information on firmware revision compatibility, go to the Rockwell Automation Support site and search for "Firmware Updates".
0xF010	The user program contains an instruction block that the Micro800 controller does not support.	Do the following: <ol style="list-style-type: none"> 1. Modify the program to ensure that all instruction blocks are supported by the controller. 2. Build and download the program. 3. Change the controller to Run mode.
0xF014	An unexpected software fault occurred.	Re-program the memory module. If the fault persists, replace the memory module.
0xF015	An unexpected software fault occurred.	Do the following: <ol style="list-style-type: none"> 1. Turn off the controller. 2. Build and download your program, and then restore any necessary data. 3. Start your system. If the fault persists, refer to the "Wire your controller" section in your controller's user manual: <ul style="list-style-type: none"> • Micro810 Programmable Controller User Manual - 2080-UM001A-EN-E • Micro820 Programmable Controller User Manual - 2080-UM005_-EN-E • Micro830, Micro850, and Micro870 Programmable Controller User Manual - 2080-UM002_-EN-E
0xF016	An unexpected hardware fault occurred.	Do the following: <ol style="list-style-type: none"> 1. Turn off the controller. 2. Download the program, and then restore any necessary data. 3. Start your system.

Table 26. Fault codes, descriptions, and corrective actions for a Micro800 controller fault (continued)

Fault Code	Description	Corrective Action
		<p>If the fault persists, refer to the "Wire your controller" section in your controller's user manual:</p> <ul style="list-style-type: none"> • Micro810 Programmable Controller User Manual - 2080-UM001A-EN-E • Micro820 Programmable Controller User Manual - 2080-UM005_-EN-E • Micro830, Micro850, and Micro870 Programmable Controller User Manual - 2080-UM002_-EN-E
0xF017	<p>An unexpected software error occurred due to an unexpected hardware interrupt.</p> <p>If the system variable <code>__SYSVA_USER_DATA_LOST</code> is set, the controller was able to recover the user program but the user data is cleared. If not, the Micro800 controller program is cleared.</p>	<p>Do the following:</p> <ol style="list-style-type: none"> 1. Turn off the controller. 2. Build and download the program, or clear the controller fault if the user program still exists. 3. Start your system. <p>If the fault persists, refer to the wiring and grounding guidelines for using surge suppressors in the "Wire your controller" section of your controller's user manual:</p> <ul style="list-style-type: none"> • Micro810 Programmable Controller User Manual - 2080-UM001A-EN-E • Micro820 Programmable Controller User Manual - 2080-UM005_-EN-E • Micro830, Micro850, and Micro870 Programmable Controller User Manual - 2080-UM002_-EN-E
0xF018	<p>An unexpected software error occurred due to SPI communication failure.</p> <p>If the system variable <code>__SYSVA_USER_DATA_LOST</code> is set, the controller was able to recover the user program but the user data is cleared. If not, the Micro800 controller program is cleared.</p>	<p>Do the following:</p> <ol style="list-style-type: none"> 1. Turn off the controller. 2. Build and download the program, or clear the controller fault if the user program still exists. 3. Start your system. <p>If the fault persists, refer to the wiring and grounding guidelines for using surge suppressors in the "Wire your controller" section of your controller's user manual:</p> <ul style="list-style-type: none"> • Micro810 Programmable Controller User Manual - 2080-UM001A-EN-E • Micro820 Programmable Controller User Manual - 2080-UM005_-EN-E • Micro830, Micro850, and Micro870 Programmable Controller User Manual - 2080-UM002_-EN-E

Table 26. Fault codes, descriptions, and corrective actions for a Micro800 controller fault (continued)

Fault Code	Description	Corrective Action
0xF019	An unexpected software error occurred because of a memory or other controller resource issue.	Do the following: <ol style="list-style-type: none"> 1. Turn off the controller. 2. Build and download the program. 3. Start your system. If the fault persists, refer to the wiring and grounding guidelines for using surge suppressors in the "Wire your controller" section of your controller's user manual: <ul style="list-style-type: none"> • Micro810 Programmable Controller User Manual - 2080-UM001A-EN-E • Micro820 Programmable Controller User Manual - 2080-UM005_-EN-E • Micro830, Micro850, and Micro870 Programmable Controller User Manual - 2080-UM002_-EN-E
0xF020	There is a fault in the base hardware or the hardware is not compatible with the Micro800 controller firmware revision.	Do one of the following: <ol style="list-style-type: none"> 1. Upgrade the controller firmware revision using ControlFLASH. 2. Replace the controller. For more information on firmware revision compatibility, go to the Rockwell Automation Support site and search for "Firmware Updates".
0xF021	The I/O configuration in the user program is not valid or does not exist in the Micro800 controller. This might happen because there is no project in the controller or the previous downloading failed.	Do the following: <ol style="list-style-type: none"> 1. Verify that you have selected the correct controller. 2. Correct the plug-in module configuration in the user program to match that of the actual hardware. 3. Compile and load the program again. 4. Change the controller into Run mode. If the fault persists, use FactoryTalk Design Workbench programming software to develop and download the program.
0xF022	The user program in the memory module is not compatible with the Micro800controller firmware revision.	Do one of the following: <ul style="list-style-type: none"> • Use ControlFLASH to upgrade the controller firmware revision to become compatible with the memory module. • Replace the memory module. For more information on firmware revision compatibility, go to the Rockwell Automation Support site and search for "Firmware Updates".

Table 26. Fault codes, descriptions, and corrective actions for a Micro800 controller fault (continued)

Fault Code	Description	Corrective Action
0xF023	<p>The controller program is cleared. This occurs when:</p> <ul style="list-style-type: none"> • A power-down occurred during program download or data transfer from the memory module. • The Flash Integrity Test failed (Micro810 LC10 only). 	<p>Do one of the following:</p> <ul style="list-style-type: none"> • Download the program. • Transfer the program using the memory module restore utility.
<p>0xF030/0xF031/0xF032/0xF033 (Micro820 L20E only)</p>	<p>Power-down information in persistent memory might not be written properly because of a noisy environment or an internal hardware failure.</p> <p>If the system variable <code>__SYSVA_USER_DATA_LOST</code> is set, the controller was able to recover the user program but the user data is cleared. If not, the Micro800 controller program is cleared.</p>	<p>Download the program.</p>
0xF050	<p>The embedded I/O configuration in the user program is not valid.</p>	<p>Do the following:</p> <ol style="list-style-type: none"> 1. Correct the embedded I/O configuration in the user program to match that of the actual hardware. 2. Build and download the program. 3. Change the controller to Run mode <p>If the fault persists, use the FactoryTalk Design Workbench programming language to develop and download the program.</p>
0xF100	<p>There is a general configuration error in the motion configuration that was downloaded from FactoryTalk Design Workbench.</p>	<p>Do one of the following:</p> <ul style="list-style-type: none"> • Correct the motion configuration in the user program. • If the fault persists, upgrade FactoryTalk Design Workbench to the latest version.
0xF110	<p>There is a motion resource error. For example, the Motion_DIAG variable is not defined.</p>	<p>Do one of the following:</p> <ul style="list-style-type: none"> • Correct the motion configuration in the user program. • If the fault persists, upgrade FactoryTalk Design Workbench to the latest version.
<p>0xF12x x is the axis ID (0-3).</p>	<p>The motion configuration for axis x is not compatible with the controller, or the configuration for axis x has a resource conflict with a previously configured axis.</p>	<p>Do one of the following:</p> <ul style="list-style-type: none"> • Delete the axes using FactoryTalk Design Workbench, and then reconfigure motion by referencing Micro830, Micro850, and Micro870

Table 26. Fault codes, descriptions, and corrective actions for a Micro800 controller fault (continued)

Fault Code	Description	Corrective Action
		Programmable Controller User Manual - 2080-UM002_-EN-E. <ul style="list-style-type: none"> • If the fault persists, upgrade FactoryTalk Design Workbench to the latest version.
0x15x where x is the axis ID (0-3).	There was a motion engine logic error detected on axis x during a motion engine cycle operation. This error might indicate a motion engine data or memory failure.	Do one of the following: <ul style="list-style-type: none"> • Clear the fault, and then change the controller to Run mode. • If the fault persists, restart all the motion components, including the controller, drive, and moving mechanism. • Download the user program.
0xF210	The expansion I/O terminator is missing.	Do the following: <ol style="list-style-type: none"> 1. Turn off the controller. 2. Attach the expansion I/O terminator to the last expansion I/O module on the system. 3. Turn on the controller.
0xF230	The number of expansion power supply modules has exceeded the upper limit.	Do the following: <ol style="list-style-type: none"> 1. Turn off the controller. 2. Ensure that the number of expansion I/O modules is no more than four. 3. Turn on the controller.
0xF240	The number of expansion I/O modules has exceeded the upper limit.	Do the following: <ol style="list-style-type: none"> 1. Turn off the controller. 2. Ensure that the number of power supply modules is no more than four. 3. Turn on the controller
0xF250	There is a nonrecoverable error and one or more expansion I/O modules cannot be detected.	Restart your controller.
0xF300	The memory module/SD card is empty.	Do the following: <ol style="list-style-type: none"> 1. Ensure that the memory module contains a valid project. 2. Download the user program and back up the program to the memory module.

Table 26. Fault codes, descriptions, and corrective actions for a Micro800 controller fault (continued)

Fault Code	Description	Corrective Action
0xF301	The project in the memory module or SD card is not compatible with the project in the controller.	Do the following: <ol style="list-style-type: none"> 1. Ensure that there is a user program with a controller that has the correct controller catalog configured. 2. Download the user program and back up the program to the memory module
0xF302	The password in the memory module or SD card does not match the password in the controller.	Do the following: <ol style="list-style-type: none"> 1. Ensure that the user program in the memory module has the correct password. 2. Download the user program with a password and back up the program to the memory module. 3. Use FactoryTalk Design Workbench to enter the password.
0xF303	The memory module or SD card is not present.	Check to ensure that the memory module is present.
0xF26x x indicates the number of the expansion I/O. If x = 0, the slot number cannot be identified.	The expansion I/O cannot function properly because the controller cannot start I/O communications.	Restart your controller.
0xF27x x indicates the number of the expansion I/O. If x = 0, the slot number cannot be identified	A non-recoverable communication fault has occurred on the expansion I/O module.	Restart your controller.
0xF28x x indicates the number of the expansion I/O. If x = 0, the slot number cannot be identified.	The baud rate for the Expansion I/O has caused an error.	Do one of the following: <ul style="list-style-type: none"> • Restart your controller. • Replace the expansion module in slot number x.

Table 26. Fault codes, descriptions, and corrective actions for a Micro800 controller fault (continued)

Fault Code	Description	Corrective Action
0xF29x x indicates the number of the expansion I/O. If x = 0, the slot number cannot be identified.	A fault is detected on the expansion I/O module.	Do one of the following: <ul style="list-style-type: none"> • Restart your controller. • Replace the expansion module.
0xF2Ax x indicates the number of the expansion I/O. If x = 0, the slot number cannot be identified.	A power failure is detected on the expansion I/O module.	Do one of the following: <ul style="list-style-type: none"> • Restart your controller. • Replace the expansion module in slot number x.
0xF2Bx x indicates the number of the expansion I/O. If x = 0, the slot number cannot be identified.	An expansion I/O configuration fault is detected.	Do one of the following: <ul style="list-style-type: none"> • Correct the expansion I/O module configuration in the user program to match the hardware configuration. • Check the expansion I/O module operation and condition. • Restart your controller. • Replace the expansion module.
0xFFxx x indicates the number of the expansion I/O. If x = 0, the slot number cannot be identified.	A fault is detected on the expansion I/O module.	Do the following: <ol style="list-style-type: none"> 1. Optionally save the fault log from FactoryTalk Design Workbench. 2. Clear the recoverable fault using FactoryTalk Design Workbench. 3. If the error persists, contact your local Rockwell Automation technical support.
0xFOAz *(1)	A fault occurred in the plug-in module during operation.	Do one of the following: <ul style="list-style-type: none"> • Check the condition and operation of the plug-in module. • Restart your controller. If the fault persists, refer to your plug-in module user manual .
0xFOBz *(1)	The I/O module configuration for the specified plug-in module does not match the actual plug-in configuration detected.	Do one of the following: <ul style="list-style-type: none"> • Correct the individual plug-in module configuration in the user program to match that of the actual hardware configuration.

Table 26. Fault codes, descriptions, and corrective actions for a Micro800 controller fault (continued)

Fault Code	Description	Corrective Action
		<ul style="list-style-type: none"> • Check the condition and operation of the plug-in module. • Restart your controller. • Replace the plug-in module. <p>If the fault persists, refer to your plug-in module user manual.</p>
0xF0Dz *(1)	While power is applied, the plug-in I/O module is removed, plugged out, or a hardware fault occurred in the plug-in I/O module.	<p>Do the following:</p> <ol style="list-style-type: none"> 1. Correct the individual plug-in module configuration in the user program. 2. Build and download the program using FactoryTalk Design Workbench. 3. Change the Micro800 controller to Run mode.
0xF0Ez *(1)	The plug-in I/O module configuration does not match the actual detected I/O configuration.	<p>Do the following:</p> <ol style="list-style-type: none"> 1. Correct the plug-in I/O module configuration in the user program. 2. Build and download the program using FactoryTalk Design Workbench. 3. Change the Micro800 controller to Run mode.
0xD011	The program scan time exceeded the watchdog timeout value. The controller hardware watchdog timeout happens if the program scan is approximately three seconds.	<p>Do one of the following:</p> <ul style="list-style-type: none"> • Determine if the program is caught in a loop, and then resolve the problem. <p>Faults might occur if your structured text program contains a For loop with the upper limit set to the maximum value of the variable. For example, setting the limit of a USINT variable to 255, or setting the limit of a UINT variable to 65535.</p> <p>Use the following steps to correct the fault:</p> <ol style="list-style-type: none"> 1. Correct the program to ensure that the upper limit is not reached. One method is to use a data type with a larger maximum value. 2. Build and download the program. 3. Change the controller to Run mode. <ul style="list-style-type: none"> • If your program is designed to have a scan time longer than three seconds, in the user program, increase the watchdog timeout value that is set in

Table 26. Fault codes, descriptions, and corrective actions for a Micro800 controller fault (continued)

Fault Code	Description	Corrective Action
		the system variable <code>_SYSVA_TCYWDG</code> , and then build and download the program.
0xF830	A fault occurred in the EII configuration.	Review and change the EII configuration in the controller properties.
0xF840	A fault occurred in the HSC configuration.	Review and change the HSC configuration in the controller properties.
0xF850	A fault occurred in the STI configuration.	Review and change the STI configuration in the controller properties.
0xF860	A divide by zero error occurred.	Do the following: <ol style="list-style-type: none"> 1. Correct the program to ensure that there is no data overflow. 2. Build and download the program using FactoryTalk Design Workbench. 3. Change the controller to Run mode.
0xF870	An index address was out of the data space.	Do the following: <ol style="list-style-type: none"> 1. Correct the program to ensure that there is no index used to access an array element beyond the array boundaries. 2. Build and download the program. 3. Change the controller mode to Run mode.
0xF878	An index used to access a bit is beyond the boundaries of the data type.	Do the following: <ol style="list-style-type: none"> 1. Correct the program to ensure that there is no index used to access a bit beyond the boundaries of the data type. 2. Build and download the program. 3. Change the controller mode to Run mode.
0xF880	A data conversion fault occurred.	Do the following: <ol style="list-style-type: none"> 1. Correct the program to ensure that there is no data conversion fault. 2. Build and download the program. 3. Change the controller to Run mode.
0xF888	The call stack of the Micro800 controller cannot support the sequence of calls to the function blocks in the current project. Too many function blocks are within another block.	Change the project to reduce the quantity of function blocks that are being called within a block.

Table 26. Fault codes, descriptions, and corrective actions for a Micro800 controller fault (continued)

Fault Code	Description	Corrective Action
0xF898	A fault occurred in the plug-in module's user interrupt configuration or operation.	Correct the plug-in module's user interrupt configuration to match that of the actual hardware.
0xF8A0	The TOW parameters are not valid.	Do the following: <ul style="list-style-type: none"> • Correct the program to ensure that all parameters are valid. • Build and download the program. • Change the controller to Run mode.
0xF8A1	The DOY parameters are not valid.	Do the following: <ul style="list-style-type: none"> • Correct the program to ensure that all parameters are valid. • Build and download the program. • Change the controller to Run mode.
0xF8A4	A fault indicating that there is a communication loss between the controller and at least one device when switching to Run mode. The extended fault code is 0x0000.	Do the following: <ol style="list-style-type: none"> 1. On the Ethernet modules on page 95 configuration page, identify the EtherNet/IP device that shows the communication-lost error. 2. Resolve the error and clear the controller fault. 3. Change the controller to Run mode.
0xFFzz *(2)	A user-created fault from FactoryTalk Design Workbench has occurred.	Do the following: <ul style="list-style-type: none"> • Optionally save the fault log from FactoryTalk Design Workbench. • Clear the recoverable faults using FactoryTalk Design Workbench. • If the problem persists, contact Rockwell Automation technical support with the fault log.
0xD00F	A particular hardware type, for example, embedded I/O, was selected in the user program configuration, but did not match the actual hardware base.	Do the following: <ul style="list-style-type: none"> • Correct the program to ensure that all parameters are valid. • Build and download the program. • Change the controller to Run mode.
0xD011	The program scan time exceeded the watchdog timeout value.	Determine if the program is caught in a loop and correct the problem. Faults might occur if your structured text program contains a For loop with the upper limit set to the maximum value of the variable. For example, setting the limit of a USINT variable to 255, or setting the limit of a UINT variable to 65535.

Table 26. Fault codes, descriptions, and corrective actions for a Micro800 controller fault (continued)

Fault Code	Description	Corrective Action
		Use the following steps to correct the faults: <ul style="list-style-type: none"> • Correct the program to ensure that the upper limit is not reached. One method is to use a data type with a larger maximum value. • Build and download the program. • Change the controller to Run mode. If your program is designed to have a scan time longer than three seconds, in the user program, increase the watchdog timeout value.

1. "z" indicates the slot number of the plug-in module. If z = 0, then the slot number cannot be identified.

2. "zz" indicates the last byte of the program number. Only program numbers up to 0xFF can be displayed. For program numbers from 01x00 through 0xFFFF, only the last byte is displayed.

Controller run mode change

Run mode change (RMC) allows [specific project changes on page 66](#) without interrupting the controller mode. RMC is available for projects using Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers.

To enter RMC mode, ensure that the controller is in one of the following modes after downloading, uploading, or discovering a project:

- **Remote program**
- **Remote run.** The controller is not in Hard Run, Hard Program, Fault, Suspend, or Download mode.



- Memory limits for RMC are defined on the memory configuration page. If the limits are exceeded, your RMC changes cannot complete.
- Renaming a Program Organization Unit (POU) in RMC mode creates a POU in the allocated memory of RMC.

Run mode change actions

The following table explains the supported and unsupported changes during run mode change (RMC).

Table 27. Run mode change actions

Category	Supported Action	Unsupported Action
Configuration	<ul style="list-style-type: none"> • Manage data logs and recipes. • Export data logs, recipes, DNP3 mappings, Modbus mappings, and PCCC mappings. 	Change, add, or delete controller or I/O configurations.
Variables and data types	<ul style="list-style-type: none"> • Create global variables and change their attributes. • Change the following attributes of global variables that are added before entering RMC: <ul style="list-style-type: none"> - Comments - Alias - Retained • Change or remove local variables that are not referenced by Micro800 controller configuration. • Delete or change the following attributes of local variables that are referenced by Micro800 controller configuration: <ul style="list-style-type: none"> - Comments - Alias - Retained • Create, change or remove defined words. • Export global and local variables. 	<ul style="list-style-type: none"> • Delete or change the attributes of local variables that are referenced by Micro800 controller configuration, excluding Comments, Alias, and Retained. • Delete existing global variables. • Change variable attributes or add or delete variables in a user-defined function block (UDFB) that is referenced by a global variable. • Create, remove, or change the attributes of user-defined arrays and structures, excluding Comments. • Change or remove the project value of variables. • Change the data type of dimensional variables.
Projects and POUs	<ul style="list-style-type: none"> • Change logic in Program Organization Units (POUs). • Copy/paste POUs within or between sessions. • Rename or delete POUs whose variables are not referenced by Micro800 controller configuration. • Reset initial values of UDFB instances. • Synchronize updates to references from UDFBs or UDFs. • Save or export projects. • Change the order of POUs in Project Organizer. • In the Parameter window, change parameters of UDFs or UDFBs that are not referenced by global variables. • Export programs. 	<ul style="list-style-type: none"> • Rename or delete POUs whose variables are referenced by Micro800 controller configuration. • Add or delete interrupt POUs and change the properties of interrupts. • Remove or rename devices. • Import projects, POUs, or variables with the XLSX format. • Perform download or upload. • Change controller.






Tip:

- Changing a defined word in RMC mode does not download the changes to the controller. To download the defined word changes to the controller, include additional changes along with the defined word change.
- RMC changes that add comments to arrays or structures not referenced by the Micro800 controller configuration cannot be downloaded to the controller.
- After changing the project value of global or local variables in RMC mode, when reconnecting to the controller, you cannot enter RMC mode again.

Run mode change toolbar

The run mode change (RMC) toolbar includes buttons to start online editing, test, accept, cancel, and finalize changes. This global-scope toolbar is at the right side of the FactoryTalk Design Workbench toolbar.

Table 28. RMC toolbar

Item	Name	Description
	Start online editing	Enter RMC mode.
	Test edits	Build the Program Organization Unit to identify errors and warnings.
	Accept edits	Accept the changes. To retain and review them the next time you connect to the controller, accept the changes before disconnecting.
	Cancel edits	Cancel the changes you've made.
	Finalize edits	Download the changes to the controller without testing or accepting them.

Make run mode changes


Run mode change (RMC) allows specific project changes without interrupting the controller mode. RMC is available for projects using Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers.





Prerequisites

The controller is in one of the following modes after downloading, uploading, or discovering a project:

- **Remote program**
- **Remote run.** The controller must be free of faults and is not in Hard Run, Hard Program, Fault, Suspend, or Download mode.

To make RMC changes

1. From the RMC toolbar, select  to enter RMC mode.
2. Make the changes as needed.

3. Download the project to the controller by doing one of the following:
 - Select  to download the project to the controller without testing. The finalize action cannot be undone.
 - When you select  to finalize edits:
 - First, any untested or unaccepted edits are accepted. If a verification error occurs in any of these edits, the finalize process will stop.
 - Next, if there is no verification error, all RMC changes will be downloaded to the controller.
 - Finally, if the controller is in **Run** mode, the edits will begin to execute.
 - Select  to build the Program Organization Unit, resolve errors or warnings as needed, and then select  to download the project to the controller.



If your changes are not tested, accepted, or finalized before disconnecting from the controller, when reconnecting to the controller, you can enter the previous stage to continue making the RMC changes.

Project import and export

Export a FactoryTalk Design Workbench project that contains one or more devices. Once exported, the project can be imported to FactoryTalk Design Workbench when the controller is offline.

By default, the project is exported to the Downloads folder with the name *<project-name>.ftdwa*.

An exported project maintains the general structure of the project and might include the following project configuration options and program elements:

- Configuration options
 - Controller configuration
 - Class1 device configuration
 - I/O module configurations
 - Interrupts
- Program elements
 - Programs
 - Function blocks
 - Global variables
 - Local variables

Import a project

Use the **Welcome** page or the **File** menu to import a project to FactoryTalk Design Workbench.

- You cannot import a project created in a later version of FactoryTalk Design Workbench to an earlier version.
- You can import a project with the following controller types from Connected Components Workbench version 22.00 or 23.00 to FactoryTalk Design Workbench:

Table 29. Controllers supporting legacy project import

Controller Type	Supported Revision
Micro810 LC10	R7.00
Micro820 L20E	R23.00
Micro850 L50E	R22.00 or later
Micro870 L70E	R22.00 or later

- You cannot import a project created in FactoryTalk Design Workbench to Connected Components Workbench.
- The import file format is CCWARC for Connected Components Workbench projects or FTDWA for FactoryTalk Design Workbench projects.

Prerequisites

- Close any open project.
- The controller is offline.

To import a project

1. Open **Import project** by doing one of the following:
 - From the **Welcome** page, select **Import project**.
 - From the **File** menu, select **Import project**.
2. In **Import project**, select **Browse**.
3. In **Open**, locate and select the file, and then select **Open**.
4. In **Import project**, select **Import**.
Duplicated project names are appended with a number to avoid overwriting the existing project files.

Export projects

Use **Project manager** or the **File** menu to export projects to your computer.

By default, the project is exported to the Downloads folder with the name *<project-name>.ftdwa*.

Export a project using the File menu

Use the **File** menu to export the currently opened project to your computer.

Prerequisites

- The project contains one or more devices.
- The controller is offline.

To export a project using the File menu

1. Open the project.
2. From the **File** menu, select **Export project**.
3. In **Export project**, select the export path and enter the file name as needed.

By default, the project is exported to the Downloads folder with the name *<project-name>.ftdwa*.

4. Select **Save**.

Export multiple projects using Project manager

Use **Project manager** to export multiple projects to your computer.

To export multiple projects using Project manager

1. From the **File** menu, select **Project manager**.
2. In **Project manager**, select one or more projects.
3. Select **Export**.
4. In **Export project**, select the export path and enter the file name as needed.
By default, the projects are exported as a ZIP file named *<FTDW_projects_YYYY-mm-dd_hh-mm-ss>*. The file contains the M8A files for all exported projects.
5. Select **Save**.

Import and export project elements

Import and export project elements to transfer the elements from one project to another. Depending on the element type, the file formats for import and export are different.

Table 30. File format for import and export

Project Element	File Format
Micro800 controller programs	M8A
Micro800 controller user-defined function blocks	M8A
Micro800 controller user-defined functions	M8A
Micro800 controller global or local variables	CSV or XLSX
Micro800 controller user-defined data types	M8A
Micro800 controller data logs	M8A
Micro800 controller recipes	M8A
Micro800 controller configuration - Modbus mapping	CSV
Micro800 controller configuration - MicroLogix™ mapping	CSV
Micro800 controller configuration - DNP3 mapping	CSV
Micro800 controller configuration - DNP3 data set descriptor	M8A
Micro800 controller configuration - DNP3 data set prototype	M8A

Export file descriptions

During exporting, the selected element and all its sub-elements are copied from the project and exported to files of specific formats.

The following table lists the types of export, what is included in the exported file, and the file naming conventions.

Table 31. Export file description table

Export Type	File Naming Convention	Exported File Contents
Projects on page 69	<project-name>.ftdwa	All project elements, including device configurations and programs
Programs on page 72	<project-name_controller-name_program-name>.m8a	All program elements, including local variables and excluding global variables
Local variables on page 72	<project-name_controller-name_program-name>.csv/xlsx	All local variables of the selected program or function block
Global variables on page 75	<project-name_controller-name_global>.csv/xlsx	All global variables of the project, excluding automatically generated variables
User-defined data types on page 77	<project-name_controller-name_data_type>.m8a	All user-defined data types in the project, including arrays and defined words, excluding automatically generated data types

Export programs, function blocks, variables, and user-defined data types

Depending on the element type, the file formats for import and export are different.

- The following elements are exported to M8A files:
 - Programs, including local variables and interrupts
 - Function blocks, including local variables
 - Functions, including local variables
 - User-defined data types, including array data types, defined words, and structure data types
 - Data logs
 - Recipes
- The following elements are exported to CSV or XLSX files:
 - Modbus mappings
 - MicroLogix mappings
 - DNP3 mappings
 - Global and local variables

Export a program or function block

Use **Project Organizer** to export the following Program Organization Units (POU) with their variables to an M8A file:

- Structured text POUs
- Ladder diagram POUs
- Function block diagram POUs
- User-defined function blocks (UDFB) in POUs
- User-defined functions (UDF) in POUs

By default, the POU is exported to the Downloads folder with the file name:

- `<project-name_controller-name_program-name>.m8a` for programs
- `<project-name_controller-name_UDFB-name_UDFB>.m8a` for UDFBs
- `<project-name_controller-name_UDF-name_UDF>.m8a` for UDFs

To export a program or function block

1. In **Project Organizer**, locate the POU you want to import.
2. Right-click the POU, and then select **Export > Program**.
3. In **Export program**, select the export path and enter the file name as needed.
4. Select **Save**.

Export global variables

Use **Project Organizer** or the **Device** menu to export the global variables of a controller project to a CSV or XLSX file. By default, the file is exported to the Downloads folder with the name `<project-name_controller-name_global>.csv/xlsx`.

The following automatically generated variables are not exported:

- I/O variables, such as `_IO_EM_DO_00`
- Motion variables, such as a motion axis variable with the `AXIS_REF` data type
- Interrupt variables, such as UFR

To export global variables

1. Open **Export global variables** by doing one of the following:
 - From the **Device** menu, select **Export > Global variables**.
 - In **Project Organizer**, right-click **Global variables**, and then select **Export global variables**.
 - In **Project Organizer**, right-click the controller node, and then select **Export > Global variables**.
2. In **Export global variables**, enter the file name, select the file format, and select the export path as needed.
3. Select **Save**.

Export local variables

Use **Project Organizer** to export the local variables of the following Program Organization Units (POU) to a CSV or XLSX file:

- Structured text program
- Ladder diagram program
- Function block diagram program
- User-defined function block
- User-defined function

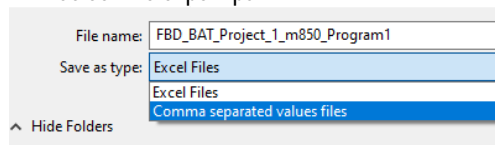
The following automatically generated variables are not exported:

- I/O variables, such as `_IO_EM_DO_00`
- Motion variables, such as a motion axis variable with the `AXIS_REF` data type
- Interrupt variables, such as UFR

By default, the file is exported to the Downloads folder with the name `<project-name_controller-name_POU-name>.csv/xlsx`.

To export local variables

1. In **Project Organizer**, locate the POU that contains the local variables you want to export.
2. Right-click the POU, and then select **Export > Local variables**.
3. (optional) In **Export local variables**:
 - Enter the file name.
 - Select the file format.
 - Select the export path.



4. Select **Save**.

Export user-defined data types

Use **Project Organizer** or the **Device** menu to export all user-defined data types to an M8A file. Exporting a single data type is not supported. Automatically generated data types are not exported.

By default, the file is exported to the Downloads folder with the name `<project-name_controller-name_data_type>.m8a`.

To export user-defined data types

1. Open **Export data types** by doing one of the following:
 - In **Project Organizer**, under the **Assets** node, right-click **Data Types**, and then select **Export data types**.
 - From the **Device** menu, select **Export > Data types**.

2. In **Export data types**, enter the file name and select the export path as needed.
3. Select **Save**.

Import controllers, programs, function blocks, variables, or user-defined data types

Depending on the element type, the file formats for importing and exporting are different.

- The following elements are imported from an M8A file:
 - [Programs on page 74](#), including local variables and interrupts
 - Function blocks, including local variables
 - Functions, including local variables
 - User-defined data types, including array data types, structure data types, and defined words
 - Data logs
 - Recipes
- The following elements are imported from a CSV or XLSX file:
 - Modbus mappings
 - MicroLogix mappings
 - DNP3 mapping
 - Global and local variables

Via CSV or XLSX files, you can manipulate the data and edit a large variable data set.

Import recommendations and considerations

Before importing, review the following recommendations and considerations:

- After importing or before downloading the project code to the Micro800 controller, verify the project code to ensure that the original functionality has been preserved.
- After importing and replacing programs, in **Project Organizer**, under the **Programs** node, restore the program execution order by moving the program before or after other programs if needed.
- When importing a program to a different type of controller than the one it was exported from, some [configured interrupts on page 83](#) might not be imported.
- Only elements that have been previously exported to an M8A, CSV, or XLSX file can be imported.
- Always verify imported variables for accuracy before using them in a project.

Import a program or function block

Use **Project Organizer** to import one of the following Program Organization Units (POUs) with its local variables from an M8A file:

- Structured text POUs
- Ladder diagram POUs

- Function block diagram POUs
- User-defined function blocks in POUs
- User-defined functions in POUs



When importing a function block diagram program from Connected Components Workbench to FactoryTalk Design Workbench:


- Program elements might overlap.
- Variable values might be partially hidden.

Adjust the location and size of the elements in the language editor as needed.

Prerequisites

- [Export a program or function block on page 72.](#)

To import a program or function block

1. In **Project Organizer**, right-click one of the following, and then select **Import <POU-name>**.
 - **Programs**
 - **User-Defined Function Blocks**
 - **User-Defined Functions**
2. In **Import <POU-name>**, select **Browse**.
3. In **Open**, locate and select the M8A file, and then select **Open**.
4. (optional) In **Import <POU-name>**, hover over  to view the errors, resolve them, and then repeat steps from 1 through 4 until all errors are resolved.
5. Select **Import**.

Import global variables

Use **Project Organizer** or the **Device** menu to import the global variables of a controller from a CSV or XLSX file. An existing variable with the same name as the imported one will be overwritten.

Variables with a user-defined data type are not supported for import unless the project contains the identical user-defined data type. To avoid problems when importing variables with a user-defined data type, consider the following steps:

- Before importing the CSV or XLSX file, create the user-defined data type in FactoryTalk Design Workbench.
- After importing the CSV or XLSX file, manually create the user-defined data type, and then change the data type of the corresponding variables to the user-defined data type you created.

If a variable property is not valid or the CSV or XLSX file does not meet the formatting requirements, FactoryTalk Design Workbench might override variables with one or more default properties. FactoryTalk Design Workbench might not import the variables correctly if the CSV or XLSX file does not meet the following requirements:

- The first two rows of the CSV or XLSX file must contain the following header values:
 - Name
 - Data Type

- The following header values are optional:
 - Dimension
 - String Size
 - Communication
 - Initial Value
 - Direction (for user-defined functions and user-defined function blocks only)
 - Attribute
 - Comment
 - Alias
 - Retained (Micro810 LC10, Micro870 L70E, and Micro820 L20E controllers only)
 - Value
- If a cell in the CSV or XLSX file contains an incorrect variable property or is empty, the incorrect property or empty cell will be replaced with the default value from the following table.

Table 32. Default values of variable properties

Variable Property	Default Value
Name	NewVariableX, where X is unique and increments by one.
Alias	<Blank>
Data Type	BOOL or Undefined
String Size	<Blank>
Communication	Read/Write
Initial Value	<Blank>
Direction (for UDF and UDFBs only)	Var
Attribute	Read/Write
Retained	False
Comment	<Blank>
Value	<Blank>

Prerequisites

- [Export global variables on page 72.](#)

To import global variables

1. Open **Import global variables** by doing one of the following:
 - From the menu bar, select **Device > Import > Global variables**.
 - In **Project Organizer**, right-click **Global Variables**, and then select **Import global variables**.
 - In **Project Organizer**, right-click the controller node, and then select **Import > Global variables**.
2. In **Import global variables**, select **Browse**.
3. In **Open**, locate and select the file, and then select **Open**.
4. In **Import global variables**, select **Import**.

Import user-defined data types

Use **Project Organizer** or the **Device** menu to import user-defined data types from an M8A file. An existing user-defined data type with the same name as the imported one will be overwritten.

To import user-defined data types

1. Open **Import data types** by doing one of the following:
 - In **Project Organizer**, under **Assets**, right-click **Data Types**, and then select **Import data types**.
 - From the menu bar, select **Device > Import > Data types**.
2. In **Import data types**, select **Browse**.
3. In **Open**, select the M8A file, and then select **Open**.
4. In **Import data types**, select **Import**.

Import local variables

Use **Project Organizer** to import the local variables of one of the following from a CSV or XLSX file:

- Structured text program
- Ladder diagram program
- Function block diagram program
- User-defined function block (UDFB)
- User-defined function (UDF)

An existing variable with the same name as the imported one will be overwritten.

Variables with a user-defined data type are not supported for import unless the project contains the identical user-defined data type. To avoid problems when importing variables with a user-defined data type, consider the following steps:

- Before importing the CSV or XLSX file, create the user-defined data type in FactoryTalk Design Workbench.
- After importing the CSV or XLSX file, manually create the user-defined data type, and then change the data type of the corresponding variables to the user-defined data type you created.

If a variable property is not valid or the CSV or XLSX file does not meet the formatting requirements, FactoryTalk Design Workbench might override variables with one or more default properties. FactoryTalk Design Workbench might not import the variables correctly if the CSV or XLSX file does not meet the following requirements:

- The first two rows of the CSV or XLSX file must contain the following header values:
 - Name
 - Data Type
- The following row header values are optional:
 - Dimension
 - String Size
 - Initial Value
 - Direction (For UDFs and UDBFs only)
 - Attribute

- Comment
 - Alias
 - Retained (Micro810 LC10, Micro870 L70E, and Micro820 L20E controllers only)
 - Value
- If a cell in the CSV or XLSX file contains an incorrect variable property or is empty, the incorrect property or empty cell will be replaced with the default value from the following table.

Table 33. Default values of variable properties

Variable Property	Default Value
Name	NewVariableX, where X is unique and increments by one.
Data Type	BOOL or Undefined
Dimension	<Blank>
String Size	<Blank>
Initial Value	<Blank>
Direction (For UDFs and UDBFs only)	Var
Attribute	Read/Write
Retained	False
Comment	<Blank>
Alias	<Blank>
Value	<Blank>

Prerequisites

- [Add a program to a project on page 80.](#)
- [Export local variables on page 73.](#)

To import local variables

1. In **Project Organizer**, locate the POU to which you want to import variables.
2. Right-click the POU, and then select **Import local variables**.
3. In **Import variables**, select **Browse**.
4. In **Open**, locate and select the file, and then select **Open**.
5. In **Import variables**, select **Import**.

Work in multiple sessions of FactoryTalk Design Workbench

Create or open separate projects at the same time using different sessions of FactoryTalk Design Workbench. With multiple sessions opened, you can:

- Open an existing project, and then without closing the existing session, start another session and use it to create a project.
- Open two separate projects, each in a different session.
- Create two projects, each in a different session.

- [Copy a Program Organization Unit \(POU\) from one session and paste it to another on page 79.](#)
 - Go online in different sessions at the same time with their respective controllers.
-

IMPORTANT:

- The same project cannot be opened in different sessions at the same time.
 - Simultaneous builds or POU verifications in multiple FactoryTalk Design Workbench sessions are not supported. Finish the build or verification in one session before starting in another session.
-

Copy a Program Organization Unit from one session and paste it to another

Open multiple sessions of FactoryTalk Design Workbench, and then copy Program Organization Units (POU) from one open session to another open session. The copied POUs reserve their configuration options.

To copy a POU from one session and paste it to another

1. Open a session of FactoryTalk Design Workbench, and then create a project or open an existing project.
2. Open another session and another project.
3. In one session, right-click the POU, and then select **Copy**.
4. In another session, in **Project Organizer**, double-click the node under which you want to paste the POU, and then select **Paste**.

Programs

A program contains the functions, function blocks, and the associated definitions in devices. A project can contain up to 256 programs.

Create programs using the following programming languages:

- [Function block diagram \(FBD\) on page 111](#)
- [Ladder diagram \(LD\) on page 102](#)
- [Structured text \(ST\) on page 117](#)

In **Project Organizer**, use the **Programs** node to:

- Create controller programs.
- Import a program from your computer to FactoryTalk Design Workbench.
- Paste a copied program under the node to duplicate the program.
- Add interrupts to the controller project.
- Change the program execution order by moving the program before or after other programs.
- Open a program.
- Rename programs. Renaming a user program automatically updates all references to the program.

Add a program to a project

Use **Project Organizer** to add a structured text (ST), ladder diagram (LD), or function block diagram (FBD) program to a controller project.

Prerequisites

- The controller is offline.

To add a program to a project

1. In **Project Organizer**, right-click **Programs**, and then select **New program**.
2. In **New program**, enter the program name, select the program type, and then enter the description as needed.



The program name must be 1 to 128 characters long, starting with a letter or underscore, followed by letters, numbers, or underscores. The program name cannot contain consecutive underscores.

3. Select **Create**.



To rename a program, in **Project Organizer**, select the program, and then in the **Name** box of the **Properties** pane, enter the new name. Renaming a user program automatically updates all references to the program.

Add a user-defined function block using Project Organizer

Use **Project Organizer** to add a user-defined function block to the controller project.

To add a user-defined function block using Project Organizer

1. In **Project Organizer**, right-click **User-Defined Function Blocks**, and then select **New user-defined function block**.
2. In **New user-defined function block**, enter the UDFB name, select the UDFB type, and then enter the description as needed.
3. Select **Create**.

Add a user-defined function using Project Organizer

Use **Project Organizer** to add a user-defined function (UDF) to the controller project.

To add a UDF using Project Organizer

1. In **Project Organizer**, expand **Assets**.
2. Right-click **User-Defined Functions**, and then select **New user-defined function**.
3. In **New user-defined function**, enter the UDF name, select the UDF type, and then enter the description as needed.
4. Select **Create**.

Add Program Organization Unit descriptions

Use the **Description** box in the **Properties** pane to document individual Program Organization Units (POUs). For example, you can document the implementation details of a POU.

Descriptions are automatically saved and stored in rich text format (*.rtf).

To add POU descriptions

1. In **Project Organizer**, select the POU.
2. In the **Properties** pane, in the **Description** box, enter the description.

Duplicate a Program Organization Unit

In **Project Organizer**, copy and paste a Program Organization Unit (POU) to duplicate it in the same or between different FactoryTalk Design Workbench sessions, especially when you have similar POUs. The pasted POU will have a number appended to it if a POU with the same name exists.



- The pasting action pastes the latest version of the copied POU. Once a copied POU is pasted, any further edits won't update the pasted POU.
 - It is not supported to copy a program to a user-defined function block (UDFB).
 - It is not supported to paste a copied POU after it has been deleted.
-

Prerequisites

- The controller is offline.

To duplicate a POU

1. In **Project Organizer**, right-click the POU, and then select **Copy**.
2. Right-click the node under which you want to paste the POU, and then select **Paste**.

Delete a Program Organization Unit

Deleting a Program Organization Unit (POU) removes the POU from **Project Organizer**. Its local variables are also deleted.

Prerequisites

- The POU is offline.

To delete a POU

Do one of the following:

- In **Project Organizer**, right-click the POU, and then select **Delete**.
- In **Project Organizer**, select the POU, and then from the **Edit** menu, select **Delete**.

Verify a Program Organization Unit

Use the **Verify program** context menu to verify the logic of a single Program Organization Unit (POU) without building the whole solution. The verification result shows on the **Output** pane. On **Project Organizer**, a symbol will display next to the POU name to indicate any errors or warnings.

To verify a POU

- In **Project Organizer**, right-click the POU, and then select **Verify program**.

Synchronize user-defined function block or user-defined function updates to its references

After updating the input or output parameters of a user-defined function block (UDFB) or a user-defined function (UDF), use the **Sync updates to references** context menu to synchronize all the updates to its references.

To synchronize UDFB or UDF updates to its references

1. In **Project Organizer**, right-click the updated UDFB or UDF, and then select **Sync updates to references**.
2. Use the following steps to open the instances of the updated UDFB or UDF to verify that the variables are assigned correctly.
 - a. Open the program that calls the UDFB or UDF.
 - b. In the language editor, right-click the UDFB or UDF, and then select **Open instruction logic**.
 - c. Verify that variables are assigned correctly.
3. Build the controller project and resolve any build errors.

Interrupt overview

Micro800 controllers support interrupts that are used in controller programs.

Use the **Programs** node in **Project Organizer** to do the following actions:

- Open **New interrupt** to create interrupts of different kinds for Micro800 controller programs.
- View a list of interrupts created for the Micro800 controller programs.
- Select an interrupt and edit its properties using the **Properties** pane.
- Import variables to an interrupt.
- Export an interrupt to your computer.
- Open and edit an interrupt using the language editor.
- Add variables to an interrupt using the local variable grid.
- Delete interrupts.

Table 34. Interrupt types and parameters

Interrupt Type	Interrupt Parameters
Event input interrupt (EII)	<ul style="list-style-type: none"> • Auto start • Embedded input
Selected timed interrupt (STI)	<ul style="list-style-type: none"> • Auto start • Embedded input
User fault routine	User fault routine interrupts start when the controller goes to Run mode, if the Program Organization Unit (POU) is configured to start automatically. There is no available parameter.
Universal plug-in module (UPM)	Auto start
High speed counter (HSC)	<ul style="list-style-type: none"> • Auto start • Overflow mask • Underflow mask • High preset mask • Low preset mask

Table 35. Interrupt types supported by different controllers

Controller Type	EII	STI	User fault routine	UPM	HSC
Micro810 LC10	No	No	Yes	No	No
Micro820 L20E	No	Yes	Yes	Yes	No
Micro850 L50E AWB	Yes	Yes	Yes	Yes	No

Table 35. Interrupt types supported by different controllers (continued)

Controller Type	EII	STI	User fault routine	UPM	HSC
Micro850 L50E QBB	Yes	Yes	Yes	Yes	Yes
Micro850 L50E QVB	Yes	Yes	Yes	Yes	Yes
Micro850 L50E QWB	Yes	Yes	Yes	Yes	Yes
Micro870 L70E AWB	Yes	Yes	Yes	Yes	No
Micro870 L70E QBB(N)	Yes	Yes	Yes	Yes	Yes
Micro870 L70E QWB(N)	Yes	Yes	Yes	Yes	Yes

Add an interrupt

Use the **New interrupt** dialog to create interrupts for Micro800 controller programs. The available interrupt types vary with the controller selection. When you are adding, configuring, or deleting interrupts, the controller must be offline.

Use interrupts to:

- Set time intervals between executions of ladder subroutines.
- Force user interrupt functions to start without using ladder logic.
- Allow or prohibit overflows and underflows for high-speed counter (HSC) interrupts.
- Allow or prohibit high and low presets for HSC interrupts.

Prerequisites

- Create a Micro800 controller project.
- The controller is offline.

To add an interrupt

1. In **Project Organizer**, right-click **Programs**, and then select **New interrupt**.
2. In **New interrupt**, do the following:
 - a. In the **Name** box, enter the interrupt name.
 - b. From the **Type** list, select an interrupt type.
 - c. From the **ID** list, select an identifier for the interrupt.
 - d. From the **Program type** list, select the program type in the project for the interrupt usage.
3. Select the parameters for the interrupt type.
4. Select **Create**.



- To delete an interrupt, in **Project Organizer**, right-click the interrupt, and then select **Delete**.
- To edit an added interrupt, in **Project Organizer**, select the interrupt, and then edit the properties in the **Properties** pane.

Interrupt instructions

To avoid potentially corrupting global variables and unnecessary warnings during an interrupt, use the interrupt instructions.

Table 36. Interrupt instructions

Function Block	Description
UIC	Clears Interrupt Lost bit for the selected user interrupts.
UID	Turns off the specific user interrupts.
UIE	Turns on a user input.
UIF	Removes a pending user input for the selected user interrupts.

I/O configurations

Use the **I/O configurations** node in **Project Organizer** to add and configure the following I/O components:

- [Embedded I/Os on page 86](#)
- [Plug-in modules on page 89](#)
- [Expansion modules on page 91](#)
- [Ethernet devices on page 95](#)

Embedded I/O configuration page

Use the **<controller-catalog-number> Embedded I/O** page to:

- View the variables associated with the I/O points by selecting the **Variable** node.
- View and configure the embedded I/O parameters of a Micro800 controller by selecting the **Configuration** node.

The number of I/O parameters varies with the Micro800 controller type. For Micro850 L50E and Micro870 L70E controllers, you can configure the inputs as [latch inputs on page 88](#) that detect the rising and falling edge of external inputs.

To open the embedded I/O configuration page, do one of the following:

- In **Project Organizer**, expand **I/O Configurations**, and then double-click **Embedded**.
- In **Project Organizer**, expand **I/O Configurations**, right-click **Embedded**, and then select **Properties**.

Figure 12. Micro800 controller embedded I/O Configuration - Variable page

2080-L70E-24QWBN Embedded I/O
Parent: MyController

<p>Information</p> <div style="background-color: #f0f0f0; padding: 2px; margin-bottom: 5px;">Variable</div> <p>Configuration</p> <p>Configuration</p>	<h3 style="margin: 0;">Variable</h3> <table border="1" style="width: 100%; border-collapse: collapse; text-align: left;"> <thead> <tr style="background-color: #f0f0f0;"> <th style="width: 15%;"></th> <th style="width: 60%;">Variable Name</th> <th style="width: 25%;">Data Type</th> </tr> </thead> <tbody> <tr><td>Input 0</td><td>_IO_EM_DI_00</td><td>BOOL</td></tr> <tr><td>Input 1</td><td>_IO_EM_DI_01</td><td>BOOL</td></tr> <tr><td>Input 2</td><td>_IO_EM_DI_02</td><td>BOOL</td></tr> <tr><td>Input 3</td><td>_IO_EM_DI_03</td><td>BOOL</td></tr> <tr><td>Input 4</td><td>_IO_EM_DI_04</td><td>BOOL</td></tr> <tr><td>Input 5</td><td>_IO_EM_DI_05</td><td>BOOL</td></tr> <tr><td>Input 6</td><td>_IO_EM_DI_06</td><td>BOOL</td></tr> <tr><td>Input 7</td><td>_IO_EM_DI_07</td><td>BOOL</td></tr> <tr><td>Input 8</td><td>_IO_EM_DI_08</td><td>BOOL</td></tr> <tr><td>Input 9</td><td>_IO_EM_DI_09</td><td>BOOL</td></tr> <tr><td>Input 10</td><td>_IO_EM_DI_10</td><td>BOOL</td></tr> <tr><td>Input 11</td><td>_IO_EM_DI_11</td><td>BOOL</td></tr> <tr><td>Input 12</td><td>_IO_EM_DI_12</td><td>BOOL</td></tr> <tr><td>Input 13</td><td>_IO_EM_DI_13</td><td>BOOL</td></tr> <tr><td>Output 0</td><td>_IO_EM_DO_00</td><td>BOOL</td></tr> <tr><td>Output 1</td><td>_IO_EM_DO_01</td><td>BOOL</td></tr> <tr><td>Output 2</td><td>_IO_EM_DO_02</td><td>BOOL</td></tr> <tr><td>Output 3</td><td>_IO_EM_DO_03</td><td>BOOL</td></tr> <tr><td>Output 4</td><td>_IO_EM_DO_04</td><td>BOOL</td></tr> <tr><td>Output 5</td><td>_IO_EM_DO_05</td><td>BOOL</td></tr> <tr><td>Output 6</td><td>_IO_EM_DO_06</td><td>BOOL</td></tr> <tr><td>Output 7</td><td>_IO_EM_DO_07</td><td>BOOL</td></tr> <tr><td>Output 8</td><td>_IO_EM_DO_08</td><td>BOOL</td></tr> <tr><td>Output 9</td><td>_IO_EM_DO_09</td><td>BOOL</td></tr> </tbody> </table>		Variable Name	Data Type	Input 0	_IO_EM_DI_00	BOOL	Input 1	_IO_EM_DI_01	BOOL	Input 2	_IO_EM_DI_02	BOOL	Input 3	_IO_EM_DI_03	BOOL	Input 4	_IO_EM_DI_04	BOOL	Input 5	_IO_EM_DI_05	BOOL	Input 6	_IO_EM_DI_06	BOOL	Input 7	_IO_EM_DI_07	BOOL	Input 8	_IO_EM_DI_08	BOOL	Input 9	_IO_EM_DI_09	BOOL	Input 10	_IO_EM_DI_10	BOOL	Input 11	_IO_EM_DI_11	BOOL	Input 12	_IO_EM_DI_12	BOOL	Input 13	_IO_EM_DI_13	BOOL	Output 0	_IO_EM_DO_00	BOOL	Output 1	_IO_EM_DO_01	BOOL	Output 2	_IO_EM_DO_02	BOOL	Output 3	_IO_EM_DO_03	BOOL	Output 4	_IO_EM_DO_04	BOOL	Output 5	_IO_EM_DO_05	BOOL	Output 6	_IO_EM_DO_06	BOOL	Output 7	_IO_EM_DO_07	BOOL	Output 8	_IO_EM_DO_08	BOOL	Output 9	_IO_EM_DO_09	BOOL
	Variable Name	Data Type																																																																										
Input 0	_IO_EM_DI_00	BOOL																																																																										
Input 1	_IO_EM_DI_01	BOOL																																																																										
Input 2	_IO_EM_DI_02	BOOL																																																																										
Input 3	_IO_EM_DI_03	BOOL																																																																										
Input 4	_IO_EM_DI_04	BOOL																																																																										
Input 5	_IO_EM_DI_05	BOOL																																																																										
Input 6	_IO_EM_DI_06	BOOL																																																																										
Input 7	_IO_EM_DI_07	BOOL																																																																										
Input 8	_IO_EM_DI_08	BOOL																																																																										
Input 9	_IO_EM_DI_09	BOOL																																																																										
Input 10	_IO_EM_DI_10	BOOL																																																																										
Input 11	_IO_EM_DI_11	BOOL																																																																										
Input 12	_IO_EM_DI_12	BOOL																																																																										
Input 13	_IO_EM_DI_13	BOOL																																																																										
Output 0	_IO_EM_DO_00	BOOL																																																																										
Output 1	_IO_EM_DO_01	BOOL																																																																										
Output 2	_IO_EM_DO_02	BOOL																																																																										
Output 3	_IO_EM_DO_03	BOOL																																																																										
Output 4	_IO_EM_DO_04	BOOL																																																																										
Output 5	_IO_EM_DO_05	BOOL																																																																										
Output 6	_IO_EM_DO_06	BOOL																																																																										
Output 7	_IO_EM_DO_07	BOOL																																																																										
Output 8	_IO_EM_DO_08	BOOL																																																																										
Output 9	_IO_EM_DO_09	BOOL																																																																										

Figure 13. Micro800 controller embedded I/O Configuration - Configuration page

2080-L70E-24QWBN Embedded I/O

Parent: MyController

Information	Configuration		
Variable			
Configuration			
Configuration			

Input	Input Filter	
0-1	DC 5 μ s	▼
2-3	Default	▼
4-5	Default	▼
6-7	Default	▼
8-9	Default	▼
10-11	Default	▼
12-13	Default	▼

Input	Input Latch		EII Edge	
0	Disabled	▼	Falling	▼
1	Disabled	▼	Falling	▼
2	Disabled	▼	Falling	▼
3	Disabled	▼	Falling	▼
4	Disabled	▼	Falling	▼
5	Disabled	▼	Falling	▼
6	Disabled	▼	Falling	▼
7	Disabled	▼	Falling	▼
8	Disabled	▼	Falling	▼
9	Disabled	▼	Falling	▼
10	Disabled	▼	Falling	▼
11	Disabled	▼	Falling	▼
12	Disabled	▼	Falling	▼
13	Disabled	▼	Falling	▼

Configure a latch input

A latch input, or a pulse-catching input, captures a fast pulse and holds it for a controller scan. The input filter determines the captured pulse width. For Micro850 L50E and Micro870 L70E controllers, you can configure certain inputs as latch inputs, set the input filter value, and then configure the rising and falling edge behaviors of each latch input.



Latch inputs change states faster than can be detected and cannot normally be detected by the input scan portion of the Micro800 controller's scan cycle.

To configure a latch input

1. In **Project Organizer**, expand **I/O Configurations**.
2. Open the embedded I/O configuration page by doing one of the following:
 - Double-click **Embedded**.
 - Select **Embedded**, and then select the **Enter** key.
3. On the **<controller-catalog-number Embedded I/O>** page, select **Configuration**.
4. In the **Input Latch** column, select one of the following for each latch input:
 - Select **Rising** to detect the pulse on the rising edge of the external input.
 - Select **Falling** to detect the pulse on the falling edge of the external input
5. Select the EII edge direction as needed.

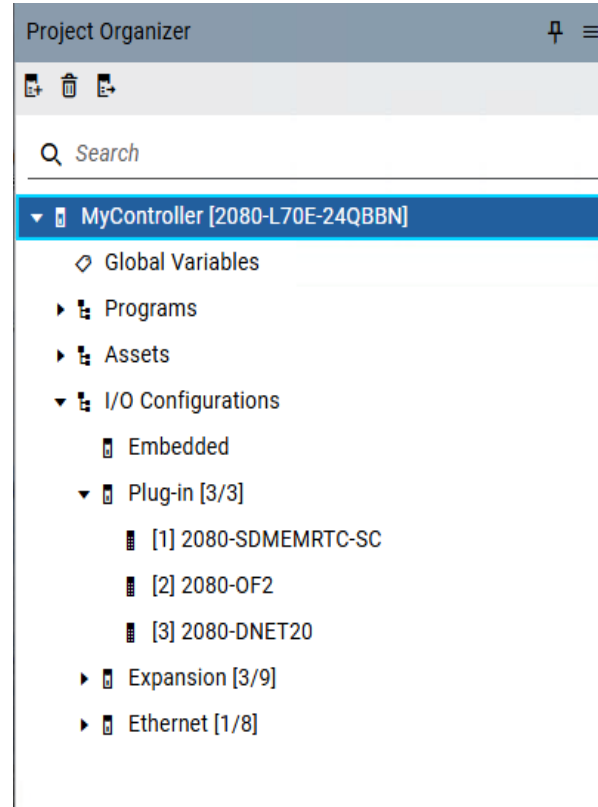
Plug-in modules

The 2080 plug-in modules are attached to the Micro800 controller to extend controller capability without increasing the controller's footprint. Use plug-ins for adding I/Os, communications, memory backup, and specialty capabilities.

Micro820 L20E, Micro850 L50E, and Micro870 L70E controllers support plug-in modules. You can configure whether to fault the controller when the plug-in module is absent.

The following image shows a Micro870 L70E controller configuration with three plug-in modules. Use the PLUGIN_INFO instruction to verify whether a plug-in module is present.

Figure 14. Micro870 L70E controller configuration with three plug-ins



After attaching a plug-in module to the controller:

- In **Project Organizer**, the **<plug-in-catalog-number>** node is created under **I/O Configurations > Plug-in**. To open the **<plug-in-catalog-number>** page, double-click the **<plug-in-catalog-number>** node.
- On the **<plug-in-catalog-number>** page, select a node to open the corresponding page and configure the plug-in settings.

Table 37. Supported 2080 plug-in modules

Catalog Number	Type	Description
2080-IF2	Analog	2-channel Voltage/Current Input
2080-IF4	Analog	4-channel Voltage/Current Input
2080-OF2	Analog	2-channel Voltage/Current Output
2080-RTD2	Analog	2-channel Resistance Thermometer Detector

Table 37. Supported 2080 plug-in modules (continued)

Catalog Number	Type	Description
2080-TC2	Analog	2-channel Thermocouple
2080-DNET20	Communication	DeviceNet
2080-SERIALISOL	Communication	RS232/485 isolated serial port
2080-IQ4	Digital	4-point digital input module
2080-IQ4OB4	Digital	8-point digital combination module
2080-IQ4OV4	Digital	8-point digital combination module
2080-OB4	Digital	4-point digital output module
2080-OV4	Digital	4-point digital output module
2080-OW4I	Digital	4-point relay output module
2080-MEMBAK-RTC2	Specialty	Memory backup and real-time clock (RTC)
2080-SDMEMRTC-SC	Specialty	MicroSD card and high accuracy RTC It is not an Allen-Bradley® plug-in. It is from Rockwell Automation Encompass™ Product Partner-Spectrum Controls.
2080-MOT-HSC	Specialty	Operates in high-speed counter mode or motion axis feedback mode
2080-TRIMPOT6	Specialty	6-channel trimpot analog input

Add a plug-in module to the controller

Use plug-ins to expand Micro800 controllers for extra I/Os, communications, memory backup, and specialty capabilities.

To add a plug-in module to the controller

1. In **Project Organizer**, go to **I/O Configurations > Plug-in**.
2. Right-click **Plug-in**, and then select **New module**.
3. In **Add module**, select a plug-in module, and then select **Add**.
4. In **Devices definition**, configure the slot, revision, and controller fault settings, and then select **OK**.



To open the configuration page of a plug-in module, in **Project Organizer**, double-click **<plug-in-ID>**.

Expansion modules

The 2085 expansion modules are attached to the right side of Micro850 L50E and Micro870 L70E controllers to extend their capabilities. The number of supported expansion modules varies with the controller type:

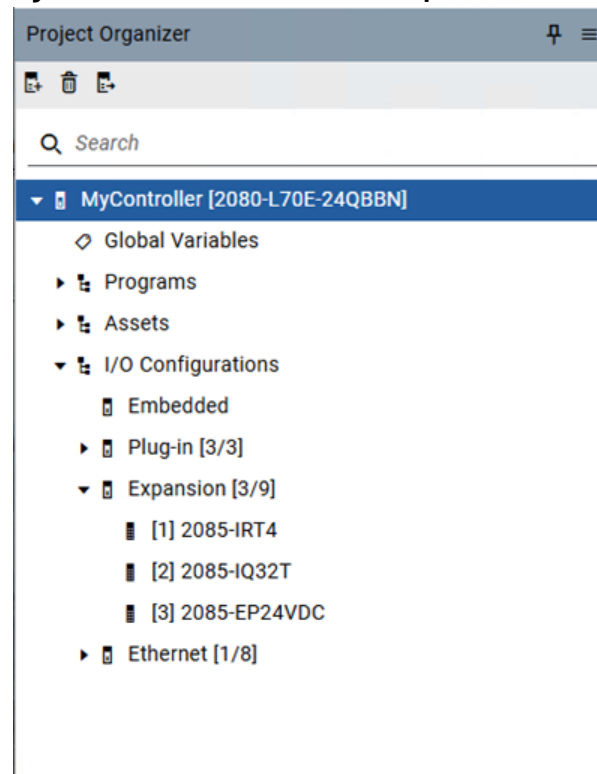
- Micro850 L50E controllers with less than 24 digital I/O points support up to four expansion I/O modules and up to 132 digital I/O points.
- Micro850 L50E controllers with 48 digital I/O points support up to four expansion I/O modules and up to 192 digital I/O points.
- Micro870 L70E controllers support up to eight expansion I/O modules if a power supply module is attached, and support up to 304 digital I/O points.

In FactoryTalk Design Workbench, you can configure whether to fault the controller if an expansion module is absent. To check if an expansion module is present, use the MODULE_INFO instruction.

The following image shows a Micro870 L70E controller configuration with three expansion modules and a power supply module. The power supply module must be in slot 2, 3, 4, or 5.

Because the Micro870 L70E expansion module does not have inputs or outputs, it uses a physical slot instead of a logical slot.

Figure 15. Micro870 L70E controller with expansion modules



After adding an analog or digital expansion module to the controller:

- Global variables are created in the controller project to support the additional I/O points.
- The module's catalog and slot number are shown in **Project Organizer**. To open the expansion module's configuration page, in **Project Organizer**, go to **I/O Configurations > Expansion**, and then double-click the expansion module.

Table 38. 2085 expansion modules

Model Number	Type	Description
2085-IF4	Multi-channel analog	4-channel 14-bit analog and current input module
2085-IF8	Multi-channel analog	8-channel 14-bit analog and current input module
2085-IRT4	Multi-channel analog	4-channel RTD and TC temperature module
2085-OF4	Multi-channel analog	4-channel 12-bit analog voltage and current output module
2085-IA8	Discrete I/O digital	8-channel 120V AC input module
2085-IM8	Discrete I/O digital	8-channel 240V AC input module
2085-IQ16	Discrete I/O digital	16-channel 24V DC sink/source input module
2085-IQ32T	Discrete I/O digital	32-channel 24V DC sink/source input module
2085-OA8	Discrete I/O digital	8-channel 120/240V AC output module
2085-OB16	Discrete I/O digital	16-channel DC source output module
2085-OV16	Discrete I/O digital	16-channel DC sink output module
2085-OW16	Discrete I/O digital	16-channel relay output module
2085-OW8	Discrete I/O digital	8-channel relay output module
2085-EP24VDC	Power supply	24V DC power supply module

For more information about expansion modules, see [Micro830, Micro850 and Micro870 Programmable Controller User Manual - 2080-UM002_-EN-E](#).

Add an expansion module to a controller project

Expansion I/O modules extend the capabilities of Micro850 L50E and Micro870 L70E controllers by optimizing the flexibility of the I/O count and type.

For Micro850 L50E controllers:

- Add no more than four expansion modules.
- To avoid program compiling problems, limit the total number of digital inputs and outputs to:
 - 192 or less for Micro850 L50E controllers with 48 digital I/O points.
 - 132 or less for other Micro850 L50E controllers.

For Micro870 L70E controllers:

- Add no more than eight expansion modules and no more than one power supply expansion module.
- To avoid program compiling problems, limit the total number of digital inputs and outputs to 304 or less.

To add an expansion module to a controller project

1. In **Project Organizer**, go to **I/O Configurations > Expansion**.
2. Right-click **Expansion**, and then select **New module**.
3. In **Add module**, select an expansion module, and then select **Add**.
4. In **Devices definition**, select the slot, revision, and controller fault behavior as needed, and then select **OK**.

Configure the scan interval for expansion modules

Micro870 L70E controllers support configuring the scan interval for individual expansion modules, except the power supply module. This allows the controller to selectively skip scanning specific expansion modules during **Run** mode. For example, setting the scan interval for analog modules to every five cycles reduces the over-scanning of slowly-changing analog data, reducing the program cycle time.



Optimizing the scan interval to reduce the program cycle time is effective only when there are more than four expansion modules.

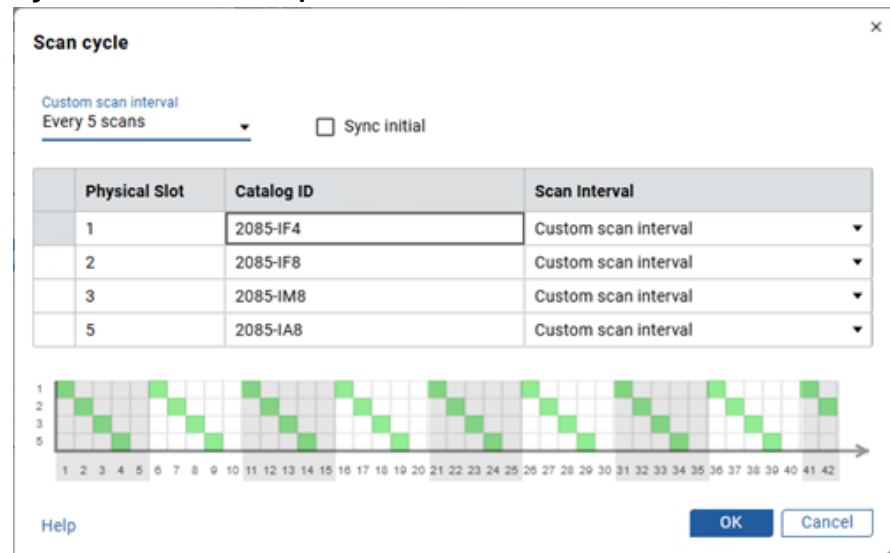
Prerequisites

- Add a Micro870 L70E controller to the project.
- [Add an expansion module to the controller on page 92.](#)

To configure the scan interval for expansion modules

1. In **Project Organizer**, go to **I/O Configurations > Expansion**.
2. Right-click **Expansion**, and then select **Scan cycle**.
3. In **Scan cycle**, from the **Custom scan interval** list, select the scan interval. By default, analog modules use the selected **Custom scan interval**, and digital modules are set to **Every Scan**.
4. Select **Sync initial** to start scanning all modules simultaneously as needed. During the first scan after changing from **Program** to **Run** mode, all modules will be scanned, regardless of their **Sync initial** configuration.
5. From the **Scan interval** list, select **Every scan** or **Custom scan interval** for individual modules as needed.
6. Select **OK**.

Figure 16. Scan interval for expansion modules



Replace an expansion module

Use these steps to replace an expansion module with another one in a controller project.

Prerequisites

- To replace a module using the **Replace module** menu, at least one module must be added to the controller.
- To replace a module using the **New module** menu, at least two modules must be added to the controller, and the module you want to replace is not in the last slot.

To replace an expansion module

Do one of the following:

- To replace an expansion module using the **Replace module** menu
 1. In **Project Organizer**, go to **I/O Configurations > Expansions**.
 2. Expand **Expansions**, right-click the module you want to replace, and then select **Replace module**.
 3. In **Add module**, select the new module, and then select **Add**.
 4. In **Devices definition**, configure the settings as needed, and then select **OK**.
- To replace an expansion module using the **New module** menu
 1. In **Project Organizer**, go to **I/O Configurations > Expansions**.
 2. Expand **Expansions**, right-click the module you want to replace, and then select **Delete**.
 3. In **Warning**, select **NO**.
 4. Under **Expansions**, right-click **Available**, and then select **New module**.
 5. In **Add module**, select the new module, and then select **Add**.
 6. In **Devices definition**, configure the settings as needed, and then select **OK**.

Ethernet device settings

After [adding an Ethernet device on page 97](#), a **<device-name>** node is created under **Ethernet** in **Project Organizer**. To view and configure the Ethernet devices, do one of the following:

- Double-click the node.
- Right-click the node, and then select **Properties**.

The settings displayed on the **<device-catalog-number: device-name>** page vary with type of the device that you added.

Table 39. Ethernet device settings

Setting	Description
Information - Overview	Displays: <ul style="list-style-type: none"> • The device catalog ID. • The device vendor. • The electronic keying and Ethernet address settings you entered when adding the device. • The IP address of the device. • The drive rating of the device. • The Device definition button. Select the button to open the Device definition dialog and change the settings as needed.
Settings in the Devices definition dialog	
Device type	Displays the catalog number of the added device. It is read-only.
Name	Displays the device name that you entered. The name must be unique. Do not set a name using the device type because it contains reserved words. We recommend that you attach a number to the device type. For example, powerflex523_1.
Description	Displays the additional information about the device. It is optional.
Ethernet address	Specifies the IP address of the added device. The default address is 1.1.1.1.
Revision	Displays the device revision, comprising a major revision number followed by a minor revision number. <ul style="list-style-type: none"> • Major revision indicates the revision of the interface to the device. It is from 0 through 127. The default value is 0. • Minor revision indicates the firmware revision. It is from 0 through 255. The default value is 0.
Electronic keying	Displays the selected electronic keying option. Electronic keying reduces the possibility of using an incorrect device in the control system by

Table 39. Ethernet device settings (continued)

Setting	Description
	<p>comparing the device specified in the project with the installed device. The available options vary with the selected device type:</p> <ul style="list-style-type: none"> • Compatible module allows the installed device to accept the key of the device that is defined in the project when the installed device can emulate the defined device. It is typically used in the following scenarios when replacing one device (device A) with another (device B): <ul style="list-style-type: none"> - Device A and device B have the same catalog ID. - Device B has the same or a later major revision than device A. - When device A and device B have the same major revision, device B must have the same or a later minor revision than device A; when device B has a later major revision than device A, device B's minor revision can be any number. • Disable keying disregards the keying attributes when attempting to communicate with a device and might result in communicating with a device whose type is different from the type specified in the project. • Exact match indicates that all keying attributes must align to establish communication. If any attribute does not match precisely, communication with the device will not be established.
Mode	<p>Specifies the device operating mode. Options include:</p> <ul style="list-style-type: none"> • Velocity • Position <p>The setting is available for the PowerFlex® 525-EENET and PowerFlex 525-E2P device type.</p>
Drive rating	<p>Lists the power rating with global voltage classes of the drive. It is available when you select a PowerFlex drive.</p>
Configuration - Connection	
Requested packet interval	<p>Specifies the connection update rate. The connection transfers data to or from the device at least as frequently as specified.</p> <p>The value is from 5 through 9999.9. The default value is 20.0. The range for Kinetix® 5100 is from 5 through 3200.</p> <p>The setting is configurable when the controller is online.</p>
Unicast connection over EtherNet/IP	<p>Determines whether to use a unicast connection over EtherNet/IP or multicast connections.</p> <p>The setting is configurable when the controller is online.</p>
Inhibit module	<p>Determines whether to inhibit the connection to the module.</p> <p>The setting is configurable when the controller is online.</p>
Major fault on controller if connection fail while in run mode	<p>Enables the controller to check the Class 1 communication with the device when switching from Program to Run mode. If connected, the controller enters Run mode. Otherwise, the 0xF8A4 fault occurs.</p>
Connection - Connection fault	<p>Shows the connection fault information if any while online.</p>
Configuration - Comm config	

Table 39. Ethernet device settings (continued)

Setting	Description
Communication format	<p data-bbox="873 254 1489 317">Specifies the device communication format. The setting is available for generic devices.</p> <p data-bbox="873 338 1489 495">The selected communication format for an I/O module determines how the module communicates with the controller, which configuration pages are available, and the tag structure and configuration method that you will use to configure the module. Specifically, the communication format determines:</p> <ul style="list-style-type: none"> <li data-bbox="881 516 1390 543">• The format of the input and output data for the I/O module <li data-bbox="881 564 1489 695">• The format of the configuration data for the I/O module <ul style="list-style-type: none"> <li data-bbox="914 600 1489 695">- Input Assembly instance specifies the input connection point for the primary connection. The value is from 1 through 65535. The default value is 1. <li data-bbox="914 716 1489 1413">- Input size specifies the size of the input assembly instance. The size range varies with the selected Communication format: <ul style="list-style-type: none"> <li data-bbox="946 800 1243 827">- Data - DINT: From 1 through 125 <li data-bbox="946 835 1235 863">- Data - INT: From 1 through 250 <li data-bbox="946 871 1243 898">- Data - REAL: From 1 through 125 <li data-bbox="946 907 1243 934">- Data - SINT: From 1 through 500 <li data-bbox="946 942 1289 970">- Input data - DINT: From 1 through 125 <li data-bbox="946 978 1289 1005">- Input data - INT: From 1 through 250 <li data-bbox="946 1014 1289 1041">- Input data - REAL: From 1 through 125 <li data-bbox="946 1050 1289 1077">- Input data - SINT: From 1 through 500 <li data-bbox="914 1085 1489 1180">- Output Assembly instance specifies the output connection point for the primary connection. The value is from 1 through 65535. The default value is 1. <li data-bbox="914 1201 1489 1413">- Output size specifies the size of the output assembly instance. The size varies with the selected Communication format: <ul style="list-style-type: none"> <li data-bbox="946 1285 1243 1312">- Data - DINT: From 1 through 124 <li data-bbox="946 1320 1235 1348">- Data - INT: From 1 through 248 <li data-bbox="946 1356 1243 1383">- Data - REAL: From 1 through 124 <li data-bbox="946 1392 1243 1419">- Data - SINT: From 1 through 496 <li data-bbox="881 1434 1489 1497">• Configuration Assembly instance specifies the target of the connection. The value is from 1 through 65535. The default value is 1. <li data-bbox="881 1518 1489 1675">• Configuration Size specifies the number of 8-bit configuration words that is required to communicate with the module. The assembly comprises the default configuration array and the additional portions from other arrays. The value is from 0 through 400. The default value is 0.

Add an Ethernet device

Use **Project Organizer** to add an Ethernet device to a project.

To add an Ethernet device

1. In **Project Organizer**, expand **I/O Configurations**.
2. Right-click **Ethernet**, and then select **New device**.
3. In **Add device**, select a device, and then select **Add**.
4. In **Devices definition**, configure the [parameters on page 95](#), and then select **OK**.
A **<device-name>** node is created under **Ethernet**.

Language editors

Use the language editor to develop the content of Micro800 programs.

A language editor holds elements from the following programming languages:

- Ladder diagram (LD)
- Structured text (ST)
- Function block (FB)

There is an additional language editor for user-defined function blocks (UDFBs).

In language editors:

- Each program in a project opens a separate language editor in the application workspace, with program names shown on the workspace tab.
- You can open several workspaces at the same time.
- Use **Toolbox** to inset elements to the language editor.






Work with elements in language editors

Use the language editors to manage individual or multiple elements for Micro800 controller programs. The selected elements are highlighted in blue.

Table 40. Working with elements in language editors

To	Do this
Select an individual element	Select the element in the language editor.
Select multiple elements	Do one of the following: <ul style="list-style-type: none"> • Select the first element, and then select the last element while holding Shift. • Select each individual element while holding Ctrl. The elements must be within the same rung.
Insert an element to a language editor	From Toolbox , drag the element to the language editor.
Delete elements	Do one of the following: <ul style="list-style-type: none"> • In the language editor, select one or more elements, and then select the Delete key. • In the language editor, right-click an element, and then select Delete. Deleting a variable does not delete the variable from the dictionary.
Move elements	In the language editor, select one or more elements, and then drag the elements to another location within the language editor. You can select multiple elements only within the same rung.

Table 40. Working with elements in language editors (continued)

To	Do this
Resize elements	Select the element, and then drag the element handle until it reaches the desired size.
Open the local variables grid	In the language editor, select  .
Show or hide grids in function block diagram programs	In the language editor, select  or  .
Zoom in or zoom out	Do one of the following: <ul style="list-style-type: none"> Select  or . Scroll the mouse wheel upward or downward while holding Ctrl.

Toolbox







Use **Toolbox** to find and add program elements to a language editor.

Toolbox contains the elements associated with the [ladder diagram language on page 103](#), [function block diagram language on page 112](#), or [structured text language on page 118](#).

In **Toolbox**:

- Use **Search** or select a category from the list to filter program elements.
- Hover over an element to view its name, description, and image.
- Drag the border between the category and list pane to resize the panes.

Table 41. Working in Toolbox

To	Do this
Pin or unpin Toolbox	Do one of the following: <ul style="list-style-type: none"> Select  or . Select , and then select Pin or Unpin.
Shrink down Toolbox and leave just a tab	Select  , and then select Collapse .
Show the content of Toolbox	Select  , and then select Expand .
Hide Toolbox	Select  , and then select Close .

Create a variable in language editors

In language editors, create variables for an instruction. The variable will simultaneously be created in the local variable, global variable, or the defined word grid, depending on the scope you selected.

To create a variable in language editors

1. Open **New variable** by doing one of the following:
 - In the ladder language editor, on an instruction, right-click an operand, and then select **New variable**.
 - In the function block diagram language editor, right-click an input reference, output reference, or variable, and then select **New variable**.
 - In the structured text language editor, right-click anywhere, and then select **New variable**.
2. In **New variable**:
 - On the **General** page, enter the variable name. The scope and data type of the variable are automatically filled.
 - On the **General** and **Advanced** pages, specify other parameters as needed.
3. Select **Create**.

View instruction block logic

Open the instruction block logic to view and edit the logic, input, and output of a user-defined function (UDF) or user-defined function block (UDFB).

To view instruction block logic

1. Open the program that contains the UDF or UDFB.
2. Right-click the UDF or UDFB, and then select **Open instruction logic**.

View instruction block parameters

In language editors, use the **Open instruction parameters** context menu to view the local variables that are assigned to the instruction block.

To view instruction block parameters

- In the language editor, right-click the instruction block, and then select **Open instruction parameters**.

Ladder diagram language

A ladder diagram (LD) is a graphical representation of Boolean equations that combines contacts (input arguments) with coils (output results).

The LD language uses graphic symbols in a program chart that is organized like a relay ladder wiring diagram to describe the tests and modifications of Boolean data.

LD graphic symbols are organized within the chart as an electrical contact diagram. The term "ladder" comes from the concept of rungs connected to vertical power rails at both ends, where each rung represents an individual circuit.

FactoryTalk Design Workbench includes an LD language editor and supports specific elements and instructions that are provided exclusively by the FactoryTalk Design Workbench software.

Ladder diagram program development environment

This section introduces the ladder diagram (LD) program language editor, where you develop LD Program Organizational Units (POU).

Figure 17. Ladder diagram program development environment

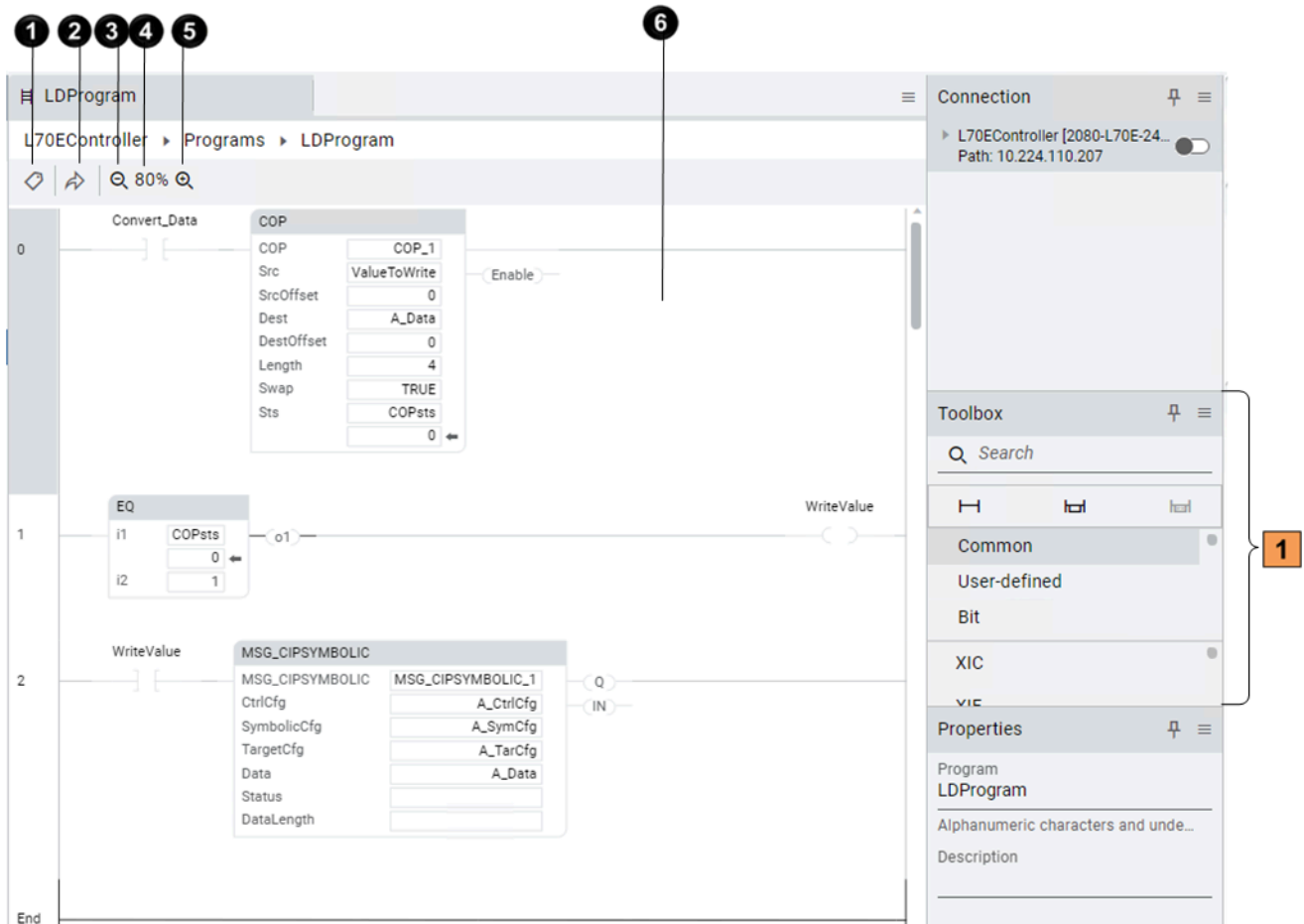


Table 42. LD program development environment

Item	Description
1	Opens the local variable grid to view the variables used by the current program.
2	Goes to the rung with the entered rung number.
3	Zooms in the editor. It resets when the tab is closed and does not affect other tabs.
4	Zooms out the editor. It resets when the tab is closed and does not affect other tabs.
5	Resets the editor to 80% zoom.
6	Makes up the primary portion of the LD editor. In this area, you edit logic using graphical instruction elements.
1	Adds instructions and elements to the editor.

Ladder editor Toolbox

Use the ladder editor **Toolbox** to add program elements to the ladder diagram.

Toolbox is displayed by default. If it is hidden, do one of the following:


- From the pane bar, select .
- From the **View** menu, select **Toolbox**.

Figure 18. Ladder editor Toolbox

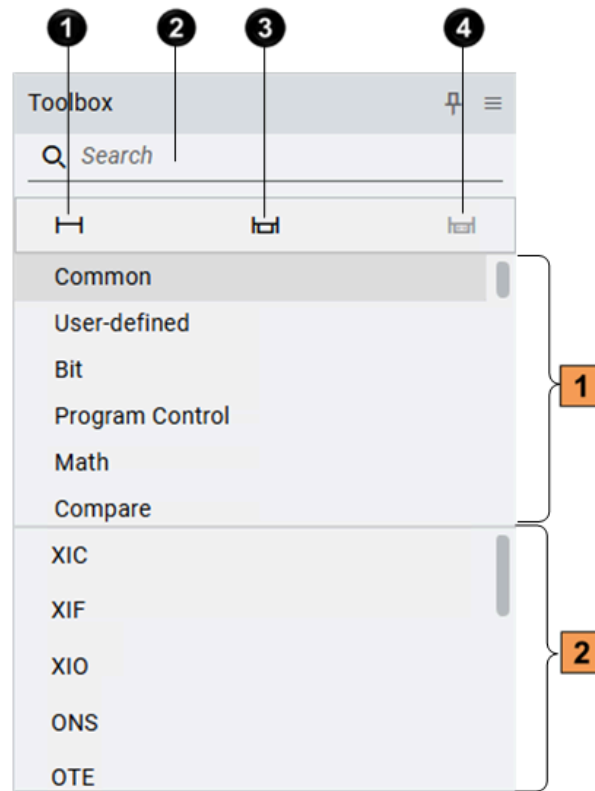


Table 43. Ladder editor Toolbox

Item	Description
1	Adds a rung to the ladder diagram.
2	Contains the entered element name for quick finding. As you enter, elements that contain the search text appear in Toolbox .
3	Adds a branch to the ladder diagram.
4	Adds a parallel branch to an existing branch.
1	Displays the categories of the program elements. Selecting a category filters the displayed program elements.
2	Displays the program elements of the selected category.

Ladder diagram value display

This section explains how variable values are displayed in ladder diagram programs for boxed instructions when the controller is offline.

For programs

When there is no initial value for a variable:

- The default value is displayed in the instruction box and variable grid. For the STRING data type, the default value is empty.

Name	Value	Initial Value
a	0	

Name ↑	Value	Initial Value
string1		

- You can change the value of the variable in the instruction box or variable grid.

Name	Value	Initial Value
a	2	

Name ↑	Value	Initial Value
string1	'This is a string'	

When there is an initial value for a variable:

- The default value or a previously entered value is displayed in the instruction box or variable grid. For the STRING data type, the default value is empty.

Name ↑	Value	Initial Value
a	0	3

Name	Value	Initial Value ↑
string1		'This is the initial value of L...

- You cannot change the value in the instruction box or variable grid. The initial value is displayed in the variable grid.

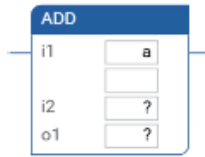
Name ↑	Value	Initial Value
a	2	3

Name	Value	Initial Value ↑
string1	'MyString'	'This is the initial value of L...

For user-defined function block (UDFB) Program Organizational Units (POU)

When there is no initial value for a variable:

- The value is empty and read-only.

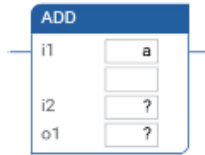


- The variable grid does not include a **Value** column.

Name	Initial Value
▶ a	

When there is an initial value for a variable:

- The value in the instruction box is empty and read-only.



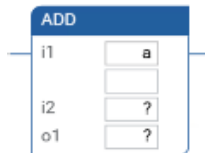
- The variable grid does not include the **Value** column, but a previously entered initial value displays in the **Initial Value** column.

Name	Initial Value
▶ a	3

For user-defined function (UDF) POUs

When there is no initial value for a variable:

- The value in the instruction box is empty and read-only.

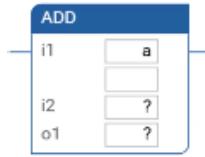


- The variable grid does not include a **Value** column.

Name	Initial Value
▶ a	

When there is an initial value for a variable:

- The value in the instruction box is empty and read-only.

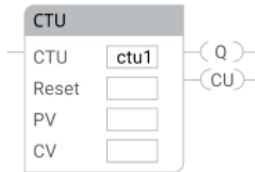


- The variable grid does not include the **Value** column, but the **Initial Value** column displays a previously entered initial value.

Name	Initial Value
▶ a	3

For an instance variable of a function block in a program, UDFB, or UDF

- The value of the instance is empty for each operand of the function block.
- After uploading the project, the value of all operands, excluding the instance variable, is empty.



Work in the ladder diagram language editor

Elements are added to the ladder diagram language editor according to the following rules:

- The first element on a rung is added at the position that you selected.
- Subsequent elements are added to the right of the selected element on the rung.
- It is not supported to insert an element to the right of a coil return or jump.

The methods to add elements to the LD program are as follows:

Table 44. Methods to add an element to the LD program

Method	Description
Ladder editor	Adds, deletes, or copies and pastes elements.
Toolbox	Adds elements and instructions to the LD program.

Toggle Bit command

Use the Toggle Bit command to change the state of a Boolean tag.

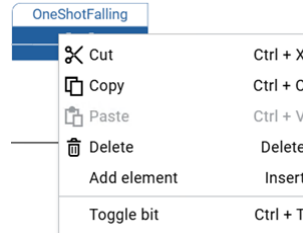
- If the tag is on, toggle turns it off.
- If the tag is off, toggle turns it on.

If the controller is offline or in **Program** mode, the toggled Boolean state will remain until manually changed.

If the controller is in **Run** mode, the toggled Boolean state might be overwritten by:

- The rung scan, if the ladder logic has outputs that reference the toggled Boolean tag.
- The input scan, if the Boolean tag references an I/O input module.
- Forcing, if the Boolean tag references an I/O module.

Figure 19. Toggle Bit command



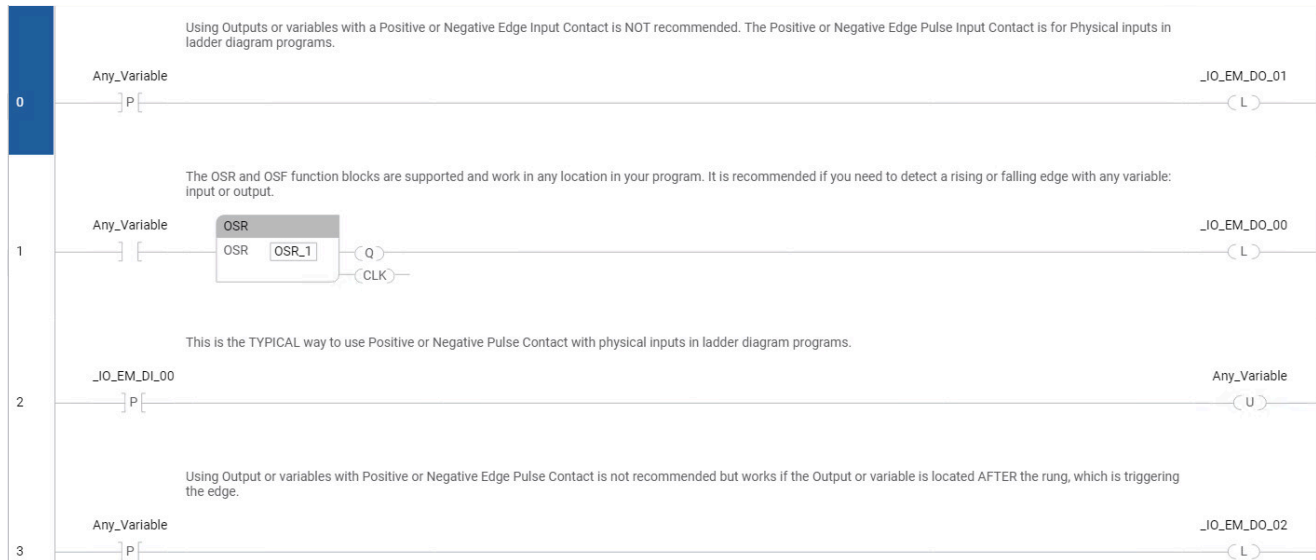
Ladder diagram program examples

The following examples demonstrate ladder diagram programs.

Example: OSR function block

The following example program shows the recommended usage of an OSR function block to detect an edge while connected to the controller.

Figure 20. Example: OSR function block



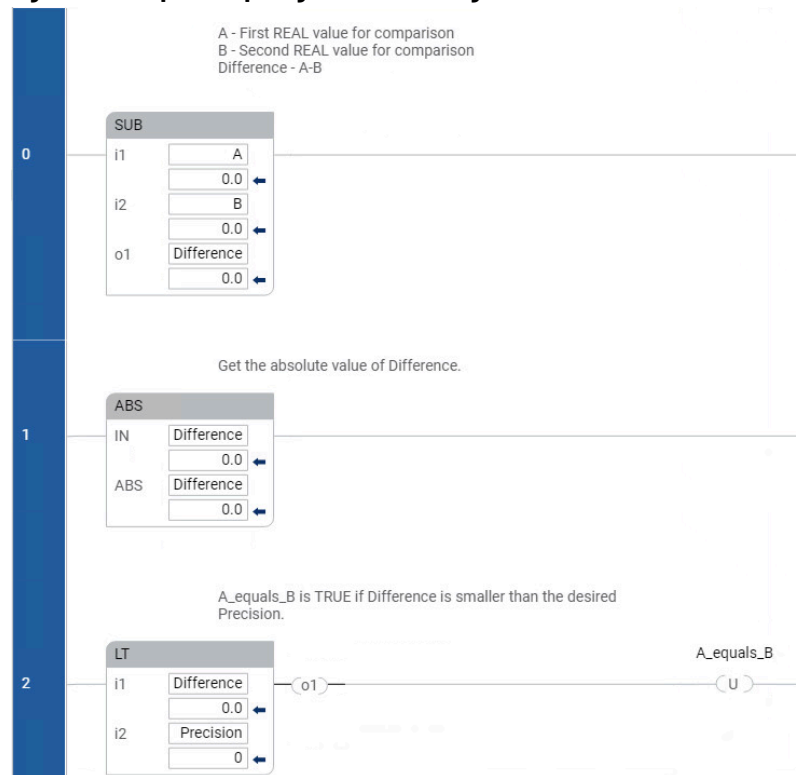
Example: Comparing REAL Values using SUB (-), ABS, and LT (<)

It is not recommended to use the REAL data type for equality comparisons due to rounding differences. Two values might appear equal in FactoryTalk Design Workbench but evaluate as false.

For example, 23.500001 and 23.499999 both display as 23.5 but are not equal in the controller.

To compare REAL values, use a SUB instruction to find the difference and check if it is less than a set precision value. See the LD program example for details.

Figure 21. Example: Comparing Real Values using SUB (-), ABS, and LT (<)



Enable EN/ENO in ladder diagram instructions

The rung state in a ladder diagram (LD) diagram is always Boolean, and the first input and first output of an instruction are connected to the rung. For instructions that have EN inputs or ENO outputs, use the **Properties** pane to enable or disable EN/ENO:

- When enabled, the value of the EN input or ENO output parameter is shown on the bit lag.
- When disabled, the value of the EN input or ENO output parameter is shown in the instruction box.

When EN/ENO is disabled:

- Assign a variable to or enter the value for the EN input or ENO output parameter for instructions that have instances.
- Assign a variable to or enter the value for EN input or ENO output parameters for instruction that do not have instances.

IMPORTANT:

- A user-defined input parameter cannot be used to enable or disable user-defined functions (UDF) or user-defined function blocks (UDFB) because the input parameter only enables or disables the instructions inside the UDF or UDFB.
- To enable or disable the running of a UDF or UDFB, select the **EN/ENO** checkbox using the **Properties** pane for the specified UDF or UDFB. For example, when **EN/ENO** is turned off, the UDFB does not execute and the output parameters are not overwritten.

To enable EN/ENO in ladder diagram instructions

1. In the ladder diagram language editor, select an instruction block instance.
2. In **Properties**, turn on **EN/ENO**.

In some cases, enabling parameters are required for the instructions that execute on call. The following example shows an SUS instruction block with an **Enable** input.

SUS	
SUS	SUS_1
Enable	_IO_EM_DO_00
	FALSE
SusID	__SYSVA_SUSPEND_ID
	0

Function block diagram language

A function block diagram (FBD) program is a graphical representation of the input and output relationships in a control system. Build your FBD programs using existing functions from a standard library, function section, or function block section.

The FBD language supports using the following instruction blocks that are supplied with FactoryTalk Design Workbench only:

- Operators
- Function blocks
- Functions
- User-defined function blocks

Function block diagram program development environment

The following diagram shows the language editor for a function block diagram (FBD) program, where you develop FBD Program Organizational Units (POU).

Use the FBD **Toolbox** or FBD [keyboard shortcuts on page 21](#) to add elements to an FBD POU.

Figure 22. FBD program development environment

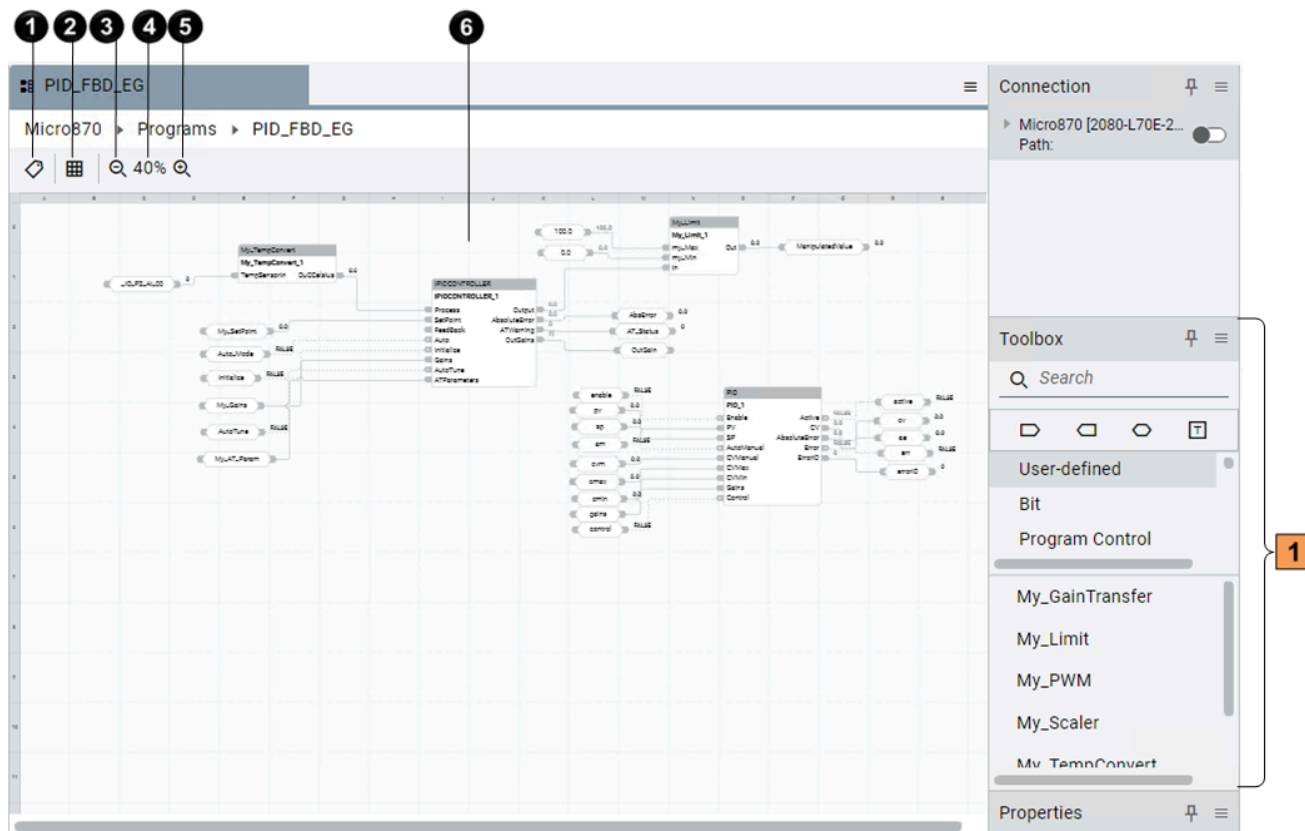


Table 45. FBD program development environment

Item	Description
1	Opens the local variables grid to view the variables that are used by the current program.
2	Displays or hides the grid lines. It resets when the tab is closed and does not affect other tabs.
3	Zooms in the editor. It resets when the tab is closed and does not affect other tabs.
4	Zooms out the editor. It resets when the tab is closed and does not affect other tabs.
5	Resets the editor to 100% zoom.
6	Makes up the primary portion of the FBD editor. In this area, you edit logic using graphical instruction elements.
1	Adds instructions and elements to the editor.

Function block diagram Toolbox

Use the function block diagram (FBD) **Toolbox** to add program elements to the FBD program.

Toolbox is displayed by default. If it is hidden, do one of the following:

- From the pane bar, select .
- From the **View** menu, select **Toolbox**.

Figure 23. FBD Toolbox

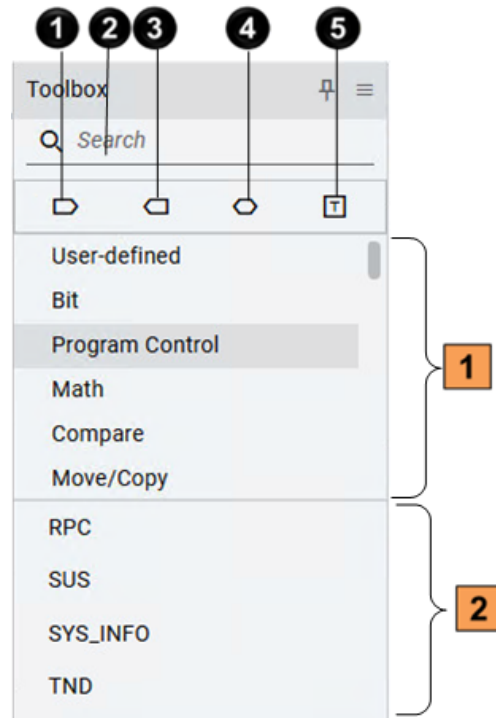


Table 46. Function block diagram Toolbox

Item	Description
1	Creates a reference to an input variable.
2	Contains the entered element name for quick finding. As you enter, elements that contain the search text appear in Toolbox .
3	Create a reference to an output variable.
4	Creates a reference to an input or output variable.
5	Contains the entered FBD program description. Descriptions are free-format text used for documentation purposes only and are not acted upon by the program.
1	Displays the categories of the program elements. Selecting a category filters the displayed program elements.
2	Displays the program elements of the selected category.

Function block diagram main format

A [function block diagram \(FBD\) on page 111](#) includes instruction blocks with input and output variables.

The outputs of instruction blocks are connected to the inputs of other instruction blocks using connection lines.

In FactoryTalk Design Workbench, an instruction block refers to the following types of instructions:

- Operator
- Function
- Function block
- User-defined function block

Figure 24. FBD example



Function block diagram program inputs and outputs

This section introduces the input and output variables, and the connection lines in a function block diagram (FBD) program.

Input and output variables

In an FBD program, each input variable must be connected to a block input and the data type of the input is correct. A block input can be a literal value, an internal variable, an input variable, or an output variable from another block.

Each output variable must be connected to a block output and the data type of the output is correct. A block output can be an internal variable, an output variable, or the name of a block (for functions only). If the output is the name of an edited function, the output represents the assignment of the return value for the function, and the output value is returned to the calling program.

Connecting inputs and outputs

Variables, inputs, and outputs in an FBD program are connected or wired using connection lines or links. You can connect any of the following logical points in an FBD diagram using a single line:

- Connect an input variable to a block input.
- Connect an output variable to a block output.
- Connect a block output to the input of another block.

Connections are oriented from left to right, carrying data from left to right. The left connection and the right connection must be of the same data type.

A single left connection point can be connected to multiple right connections, called divergences, to broadcast information to multiple points. All connections must be of the same data type.

Instruction blocks in function block diagram programs

The FactoryTalk Design Workbench instruction set includes instruction blocks in compliance with IEC 61131-3. Instruction blocks collectively include function blocks, functions, and operators.

Connect instruction block inputs and outputs to variables or other instruction block inputs and outputs.

Instruction block conventions

The IEC61131-3 programming language specification addresses numerous aspects of programmable controllers, including operating system execution, data definitions, programming languages, and instruction sets. The IEC61131-3 specification provides a minimum set of functionalities that can be extended to meet end user applications.

Instruction block names

Functions and function blocks are represented by a box that displays the name of the instruction and the short version of the parameter names. For function blocks with an instance, the instance name is displayed below the function block name.

Instruction block return parameters

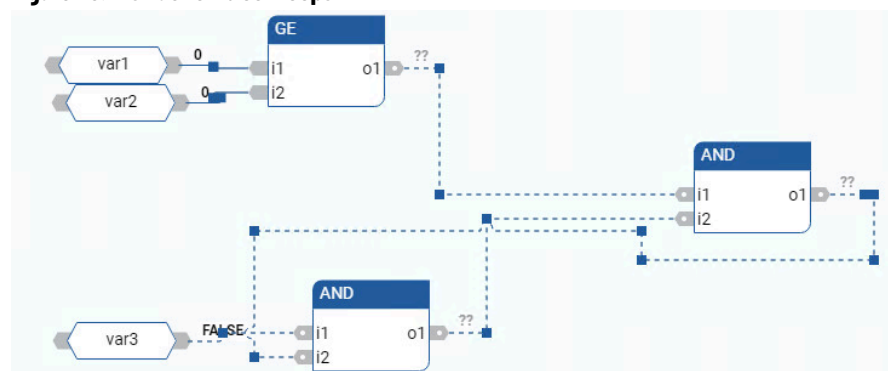
- The return parameter of a function has the same name as the function. The return parameter is the only output.
- The return parameters of a function block can have any name. Multiple return parameters can provide multiple outputs.
- You can define the parameters of programs for each controller by navigating the tabs for individual controllers displayed in the **Parameter** view.

Instruction block loops

For loops in instruction blocks, because they are initialized with a value, you must use local variables. The following FBD program shows a valid loop where:

- The local variables initialize both inputs of the greater than or equal operator (\geq) and one input of the first AND operator.
- The output of the second AND operator initializes the second input of the first AND operator.

Figure 25. Instruction block loops



Execution order of function block diagram programs

During program execution, an instruction block is any element in the diagram, a network is a group of instruction blocks linked together, and the position of an instruction block is based on its top-left corner.

The following rules apply to the execution order of an FBD program:

- Networks are executed from left to right and top to bottom.
- All inputs must be resolved before executing the instruction block. When the inputs of two or more instruction blocks are resolved at the same time, the decision for the execution is based on the position of the instruction block (left to right and top to bottom).
- The outputs of an instruction block are executed recursively from left to right and top to bottom.

Re-order elements in function block diagram programs

In the function block diagram (FBD) language editor, move elements to change the execution order.

To re-order elements in FBD programs

- Do one of the following:
 - Drag an individual element to the new location. Connections will re-organize when the mouse is released.
 - Use the **CTRL** or **Shift** key, or drag the cursor to select several elements, and then drag the selected elements or the selection area to the new location. Connections will re-organize when the mouse is released.

Monitoring function block diagram programs

When monitoring function block diagram (FBD) programs, you are monitoring the value for the input reference, variable, function, and function block at the output of an FBD.

The values appear in color, number, or text values according to their data type:

- Boolean data type output values are displayed in either black or gray, regardless of whether the value is True or False. A black output value indicates that it can be modified, and a gray output value indicates that it cannot be changed.
- Output values of the SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, LINT, ULINT, LWORD, REAL, LREAL, TIME, DATE, and STRING data types are displayed as a number or text value in the element.
- When the output value for a number or text value is unavailable, question marks (??) appear in the output label. Values also appear in the corresponding **Dictionary** instance.

Structured text (ST) language

Structured text (ST) is a high-level structured programming language designed for automation processes. It handles complex procedures that graphic languages like ladder diagrams or function block diagrams cannot easily express.

FactoryTalk Design Workbench includes an ST editor and supports the elements and instructions that are supplied with the FactoryTalk Design Workbench software only.

Structured text program development environment

This section introduces the structured text (ST) program language editor.

Figure 26. Structured text program development environment

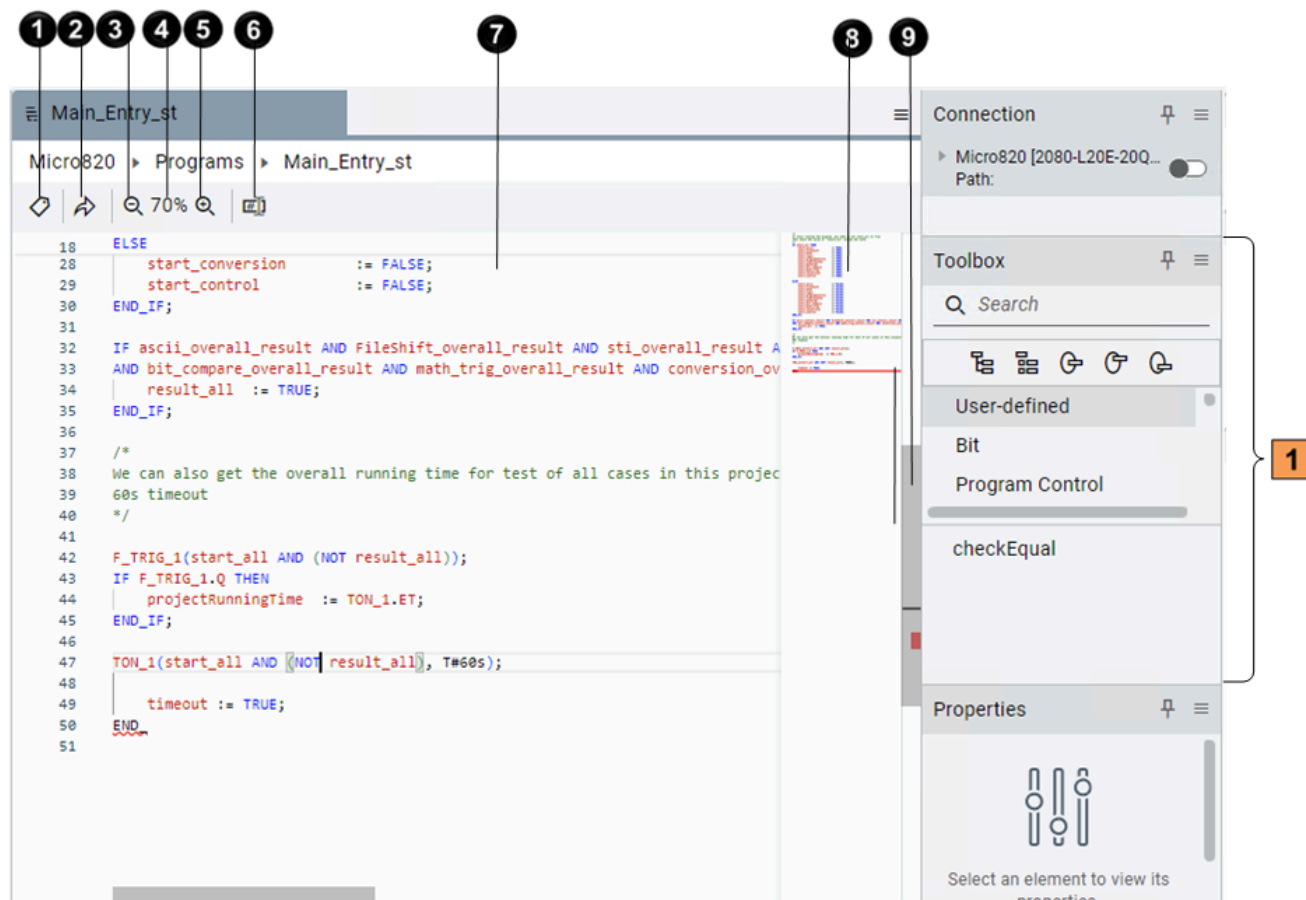


Table 47. ST program development environment

Item	Description
1	Opens the local variable grid to view the variables that are used by the current program.
2	Opens a box to enter a line number to go to. The box shows: <ul style="list-style-type: none"> The current line number where your cursor is located. The number of characters in the current line. The range of line numbers that you can enter in the box.

Table 47. ST program development environment (continued)

Item	Description
	When a line number is entered, the line is highlighted. Selecting Enter goes to the line.
3	Zooms out the editor. It resets when the tab is closed and does not affect other tabs.
4	Resets the editor to 100% zoom.
5	Zooms in the editor. It resets when the tab is closed and does not affect other tabs.
6	Displays or hides the values of the variables, through which you can read the value and check the project logic. It does not affect other tabs.
7	Makes up the primary portion of the structured text editor. In this area, you add all of the text and make any edits to your ST program. When you are entering instructions, a prompt list displays for possible instructions, and then a tooltip shows the instruction definition and expected parameters of the selected instruction. If there are any errors in your code, tooltips will provide error explanations.
8	Displays a scaled-down overview of the entire content. The slider indicates the portion of code displayed in the editor, with errors highlighted in red. Select or drag the shaded area to quickly jump to different sections in your code. Use the ST editor commands on page 119 to configure the mini-map.
9	Scrolls the page up and down. The short line on the scroll bar indicates your current position.
1	Adds elements to the ST editor.

Structured text Toolbox

Use the structured text (ST) **Toolbox** to add program elements to the ST diagram.

Toolbox is displayed by default. If it is hidden, do one of the following:


- From the pane bar, select .
- From the **View** menu, select **Toolbox**.

Figure 27. ST Toolbox

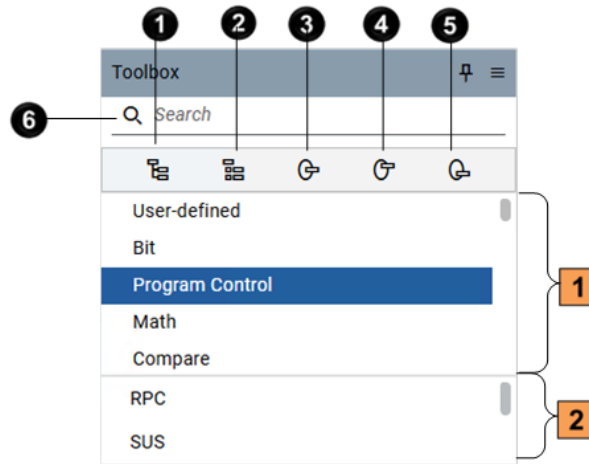


Table 48. ST editor Toolbox

Item	Description
1	Adds the code snippet IF-THEN-ELSEIF-ELSE-END_IF statement to the editor.
2	Adds the code snippet Case statement to the editor.
3	Adds the code snippet FOR statement to the editor.
4	Adds the code snippet WHILE statement to the editor.
5	Adds the code snippet REPEAT statement to the editor.
1	Displays the categories of the program elements. Selecting a category filters the displayed program elements.
2	Displays the program elements of the selected category.
6	Contains the entered element name for quick finding. As you enter, elements that contain the search text appear in Toolbox .

Commands in structured text editor

This section introduces the commands when you right-click the language editor or the scroll bar.

Table 49. Commands in structured text editor

Command	Description
Cut	Moves the selected text or the current line to the clipboard.
Copy	Copies the selected text or the current line to the clipboard.
Paste	Inserts the clipboard text to the current cursor position.

Table 49. Commands in structured text editor (continued)

Command	Description
Add element	Opens Toolbox if hidden, highlighting your last position.
Create new variable	Opens New variable to define a new variable within the code.
Find all references	Opens the cross-reference browser to view reference information of the selected variable.
Select variable	Opens the variable selector to replace the selected variable with a new one.
Open instruction parameters	Opens the local variable grid to view the variables that are used in the selected instruction.
Open instruction logic	Opens the definition of the selected instruction.
Open monitor	Opens the corresponding variable grid to monitor the variable in real time. The menu is available when the controller is online.

Code snippets

Code snippets are predefined blocks of code for inserting into a structured text program. After inserting a snippet, fill in the parameters.

FactoryTalk Design Workbench supports the following predefined code snippets:

- IF-THEN-ELSIF-ELSE-END_IF
- CASE
- FOR
- WHILE
- REPEAT

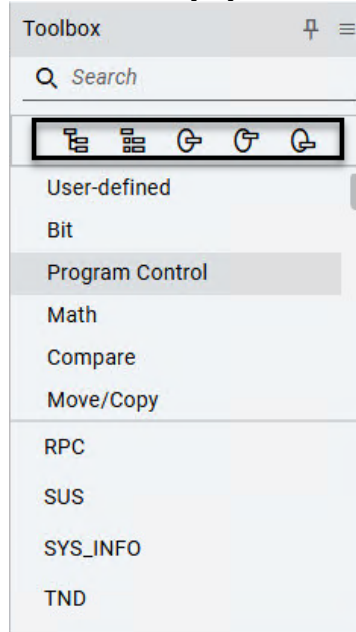
Insert a code snippet

Code snippets are predefined blocks of code for inserting into a structured text program.

To insert a code snippet

1. In the structured text language editor, put the cursor where you want to insert a snippet.
2. Enter the keyword for the snippet, and then select the snippet from the prompt list or select **Tab**.

You can also drag a snippet from **Toolbox** or double-click it to insert it at the cursor's location in the language editor.



3. Fill in the parameters for the snippets.

Structured text main syntax

A structured text (ST) program is a list of ST statements.

The following rules apply to the ST program statements:

- Each statement ends with a semi-colon (";") separator.
- Names used in the source code, such as variables, identifiers, constants, or language keywords, are separated with:
 - An inactive separator, such as a space
 - An active separator, such as ">", which indicates a "greater than" comparison
- Comments are non-executed information and can be included anywhere in an ST program. Insert comments at the beginning of a line or after code using the following syntaxes:
 - Single-line comments begin with "//" and end at the end of the line.
 - Multi-line comments begin with "/*" and end with "*"".
 - Multi-line comments begin with "/*" and end with "*/".



Indirect bit addressing is supported when a DINT variable or a defined word is used to specify the bit number.

ST statement types

Assignment statement

```
variable := expression;
```

Function call

- (*Non-formal syntax with all inputs separated by commas in set order and output as a separate statement.*)
 Output1 := FUNCTION_NAME(Input1, Input2);

- (*Formal syntax with inputs and outputs separated by comma optionally listed and in any order.*)

```
Output1 := FUNCTION_NAME(InputParameter1 := Input1,
InputParameter2 := Input2);
```

Function block call

- (*Non-formal syntax with all inputs separated by commas in set order and output as a separate statement.*)

```
FUNCTION_BLOCK_INSTANCE(Input1, Input2, ...);

Output1 := FUNCTION_BLOCK_INSTANCE.OutputParameter1;

Output2 := FUNCTION_BLOCK_INSTANCE.OutputParameter2;
```
- (*Formal syntax with inputs and outputs separated by comma optionally listed and in any order.*)

```
FUNCTION_BLOCK_INSTANCE(InputParameter1 := Input1,
InputParameter2 := Input2, OutputParameter1 => Output1,
OutputParameter2 => Output2);
```

Selection statements

IF, THEN, ELSE, CASE...

Iteration statements

FOR, WHILE, REPEAT...

Control statements

RETURN, EXIT...

Special statements

Use special statements to link with other languages.

- In the ST language editor, items displayed in different colors:
 - Black: Basic code, numbers, and strings of text
 - Blue: Keywords and functions
 - Green: Comments
 - Red: Variables and function block instances
- The inclusion of inactive separators between active separators, literals, and identifiers increases ST program legibility. ST inactive separators include:
 - Space - blank
 - Tabs
 - End of line - can be placed anywhere in a program
- Guidelines for using inactive separators:
 - Write no more than one statement for each line.
 - Use tabs to indent complex statements.
 - Insert comments to increase legibility of lines or paragraphs.

Call functions

The [structured text programming language on page 117](#) can call functions. Function calls can be used in any expressions, and contain properties explained in the following table.

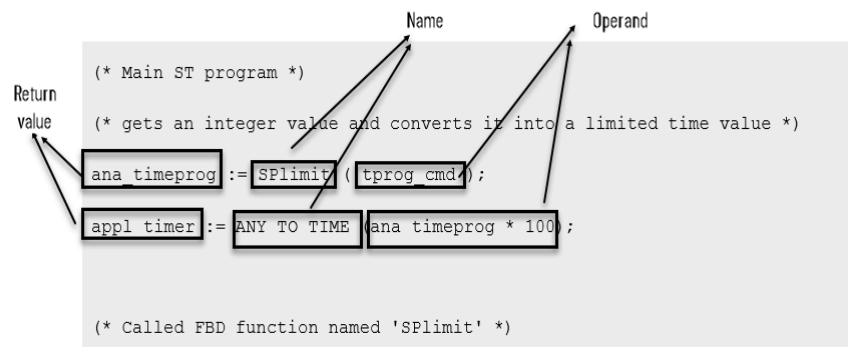
Function call properties

A function is an instruction block that has input parameters and one output parameter. It can be written in structured text (ST), ladder diagram (LD), or function block diagram (FBD) languages.

Table 50. Properties of function calls

Property	Description
Name	The name of the called function written in an IEC 61131-3 language.
Meaning	Calls an ST, LD, or FBD function, and gets its return value.
Syntax	<code><variable> := <funct> (<par1>, ... <parN>);</code>
Operands	The type of return value and calling parameters must follow the interface defined for the function.
Return value	Value returned by the function.

Figure 28. Function call example



When setting the value of the return parameter in the body of a function, you can assign the return parameter using the same name as the function:

```
FunctionName := <expression>;
```

Example: IEC 61131-3 function call

```

(* Main ST program *)
(* gets an integer value and converts it into a limited time value *)
appl_timer := ANY_TO_TIME (ana_timeprog * 100);

(* Called FBD function named 'SPlimit' *)

```

Call function blocks

The [structured text programming language on page 117](#) can call function blocks. Function block calls can be used in any expression, and contain properties explained in the following table.

Function block call properties

Table 51. Properties of function block calls

Property	Description
Name	Name of the function block instance.
Meaning	Calls a function block from the standard library, or from a user-defined library, and accesses its return parameters.
Syntax	<pre>(* call of the function block *) <blockname> (<p1>, 2 : <p2> ...); (* gets its return parameters *) <result> := <blockname>. <ret_param1>; ... <result> := <blockname>. <ret_paramN>;</pre>
Operands	Parameters are expressions that match the type of the parameters specified for that function block.
Return value	See syntax to get the return value.

Figure 29. Function block call example

```
(* ST program calling a function block *)

(* declare the instance of the block in the variable editor: *)

(* trigb1 : block R_TRIG - rising edge detection *)

(* Function block activation from ST language *)
trigb1 (b1);

(* return parameters access *)
If trigb1.Q Then nb_edge := nb_edge + 1; End_if;
```

Diagram annotations:

- Operand:** Points to the function block name `trigb1` in the call `trigb1 (b1);`
- Name:** Points to the parameter `b1` in the call `trigb1 (b1);`
- Return value:** Points to the return parameter `Q` in the assignment `trigb1.Q`.

When setting the value of the return parameter in the body of a function block, you can assign the return parameter using its name concatenated with the function block name:

```
FunctionBlockName.OutputParaName := <expression>;
```

Example

```
(* ST program calling a function block *)

(* declare the instance of the block in the variable editor: *)
(* trigb1 : block R_TRIG - rising edge detection *)

(* Function block activation from ST language *)
trigb1 (b1);
```

```
(* return parameters access *)
If (trigbl.Q) Then nb_edge := nb_edge + 1; End_if;
```

Structured text keyboard shortcuts

The following keyboard shortcuts are available for use with the structured text language.

Table 52. Structured text keyboard shortcuts

Shortcut	Description
Ctrl+G	Opens a box to enter a line number to go to.
Ctrl+M	Displays or hides the values of variables.
Ctrl++	Zooms in the language editor.
Ctrl+-	Zooms out the language editor.
Ctrl+Space	Displays possible completions for your code.
Ctrl+Shift+Space	Displays different parameter options as you enter the code.
F1	Opens the context-sensitive help.
Ctrl+F	Searches for the entered text in the language editor.

FactoryTalk Design Workbench settings

Use the **Settings** page from the **File** menu to customize the FactoryTalk Design Workbench application settings.

Switch FactoryTalk Design Workbench language

Use the **Settings** page from the **File** menu to switch the language of the FactoryTalk Design Workbench software. The language change takes effect after restating FactoryTalk Design Workbench.

To switch FactoryTalk Design Workbench language

1. From the **File** menu, select **Settings**.
2. In **Settings**, on the **General** page, select a language from the **Language** list.
3. Select **OK**.
4. Restart FactoryTalk Design Workbench for the language change to take effect.

Product lifecycle information

Rockwell Automation supports tracking long product lifecycles, helping to innovate and evolve by identifying current lifecycle stages and managing transitions to modern technologies.

FactoryTalk Design Workbench shows the product lifecycle status of Micro800 controllers in the **Selection** pane of the **Add device** dialog. The lifecycle status also recommends replacement products if the status of the selected device is **End of Life**.

When installed, FactoryTalk Design Workbench updates the product lifecycle information if the last update was over 90 days, or the device can be updated manually.

For the latest product lifecycle information, go to the [product lifecycle status page](#).

Table 53. Product lifecycle status

Lifecycle Status	Description
Active	Most current offering within a product category.
Active Mature	Product is fully supported, but a newer product or family exists. Gain value by migrating.
End of Life	Discontinued date announced-actively executes migrations and last time buys. Product generally orderable until the discontinued date. Outages on specific items might occur prior to the Discontinued date.
Discontinued	New product no longer manufactured or procured. Limited stock might be available in run-out mode, regionally. Repair or exchange services might be available.

Update the product lifecycle information

Use the **Settings** menu to update the lifecycle status information for your device.

Prerequisites

- Ensure connection to the network.
- Rockwell Automation's website is accessible.

To update the product lifecycle information

1. From the menu bar, select **File > Settings**
2. In **Settings**, select **Product lifecycle**.
3. Select **Update device lifecycle information**.
Select **Automatically update every 90 days** as needed.

If the update fails, the lifecycle status remains not changed.

Configure general controller settings

Use the **Controller - General** page to configure the following settings:

- **Communication - Timeout:** Sets the communication timeout period between 300 milliseconds and 15000 milliseconds. The default value is 3000 milliseconds.
- **Data Log - ZIP file folder hierarchy:** Specifies whether to reserve the folder structure of the data log files or combine data log files to a TXT file when exporting data logs. The default setting is **Preserve source folder structure**.
- **Recipe - Destination:** Defines the destination path on your computer for uploading recipes from the SD card. The default path is the Downloads folder.

To configure general controller settings

1. From the **File** menu, select **Settings**.
2. In **Settings**, select **Controller > General**.
3. Customize the settings as needed, and then select **OK**.
Select **Reset** to restore the default settings.

Rockwell Automation Support

Use these resources to access support information.

Technical Support Center	Find help with how-to videos, FAQs, chat, user forums, and product notification updates.	rok.auto/support
Local Technical Support Phone Numbers	Locate the telephone number for your country.	rok.auto/phonesupport
Technical Documentation Center	Quickly access and download technical specifications, installation instructions, and user manuals.	rok.auto/techdocs
Literature Library	Find installation instructions, manuals, brochures, and technical data publications.	rok.auto/literature
Product Compatibility and Download Center (PCDC)	Get help determining how products interact, check features and capabilities, and find associated firmware.	rok.auto/pcdc

Documentation Feedback

Your comments help us serve your documentation needs better. If you have any suggestions on how to improve our content, complete the form at rok.auto/docfeedback.

Waste Electrical and Electronic Equipment (WEEE)







At the end of life, this equipment should be collected separately from any unsorted municipal waste.

Rockwell Automation maintains current product environmental information on its website at rok.auto/pec.

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

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

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

Connect with us.    

rockwellautomation.com — expanding **human possibility**[®]

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

EUROPE/MIDDLE EAST/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2663 0600

ASIA PACIFIC: Rockwell Automation SEA Pte Ltd, 2 Corporation Road, #04-05, Main Lobby, Corporation Place, Singapore 618494, Tel: (65) 6510 6608

UNITED KINGDOM: Rockwell Automation Ltd., Pitfield, Kiln Farm, Milton Keynes, MK11 3DR, United Kingdom, Tel: (44)(1908)838-800