



Hardy Device Library

Release v1.01



Allen-Bradley

by ROCKWELL AUTOMATION

Reference Manual

Original Instructions

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.

	Table of Contents	3
	Preface	7
	Device Object Libraries Overview	7
	Application Code Manager	7
	Other Application Code Libraries	8
	Software and Firmware Upgrades	9
	Rockwell Automation® Services and Support	9
	Chapter 1	
Rockwell Automation® Hardy Device Library	Compatibility	11
	Compatible Software	11
	Compatible Hardware	11
	Summary of Changes	12
	Footprint	12
	Additional Resources	12
	Chapter 2	
Library Components	Hardy Device Instructions	13
	Library Folders and Files	14
	Visualization Files	16
	Studio 5000 View Designer® Visualization Files	16
	Basic Faceplate Attributes	17
	Common Status Banner	17
	Faceplate Navigation	18
	Faceplate Revision Notes	18
	Faceplate Help	19
	Launch Buttons	20
	Library Versions	21
	State Model	23
	Interfaces	24
	Application Code Manager	25
	Architectural Overview	25
HMI Tags	25	
	Chapter 3	
Using the Library	Install the Library	27
	Download the Library	27
	Download & Install Studio 5000® Application Code Manager	27
	Register Libraries in Studio 5000® Application Code Manager	28
	Importing Logic into Studio 5000® Projects	30
	Import Library Objects Wizard	30
	Import Rung Logic	33
	Using Studio 5000 View Designer®	36
Using View Designer Project Files	36	

- Configuring View Designer Objects 37
- Using FactoryTalk® View Studio 37
 - Import HMI Tags 37
 - Import FactoryTalk® View Visualization Files 40
 - Configuring FactoryTalk® View Objects 40
- Using FactoryTalk® Optix 41
 - Register FactoryTalk Optix Library 42
 - Import Library Objects in FactoryTalk Optix Project 42
 - Using Style Sheets 42
 - Configuring FactoryTalk Optix Objects 45
- Library Upgrades 46
 - Add-On Instruction Upgrades 46
 - FactoryTalk® View Upgrades 51
 - Studio 5000 View Designer® Upgrades 51

Chapter 4

Using Application Code Manager

- Overview of Application Code Manager 53
- Creating a New Project 53
- Adding & Configuring Device Objects 56
 - Adding Hardy Device Objects 56
- Configuring Displays 59
- Generating Displays 61
- Importing Displays into FactoryTalk® View Studio 61
- Generating Controller Files 63
- Exporting Attachments 64

Chapter 5

Using the Hardy Device Library with Other Application Code Libraries

- Application Code Libraries 67
- Using Hardy Device Objects with PlantPax® Process Objects Library 67
- Using Hardy Device Objects with Machine Builder Library 67

Chapter 6

HI1756 - ControlLogix® Weigh Scale Module (raC_Dvc_HI_1756xWS)

- Overview 69
- Functional Description 69
- Required Files 69
 - Controller Files 69
 - FactoryTalk® View HMI Files 70
 - Studio 5000 View Designer® HMI Files 70
 - Studio 5000® Application Code Manager Files 71
- Device Definition 71
 - HARDWARE Definition 72
- Operations 73
 - Faults & Warnings 73
 - Execution 74
- Add-On Instruction I/O Data 74
 - Add-On Instruction Ladder Implementation 74
 - InOut Data 75
 - Input Data 75

Output Data. 77

Data Types. 79

 raC_UDT_ItfAD_Hardy_CtrlSet. 79

 raC_UDT_ItfAD_Hardy_CtrlCmd 80

 raC_UDT_ItfAD_Hardy_CtrlSts 81

 raC_UDT_Event 82

 raC_UDT_HI_1756xWS_Cfg 82

 raC_UDT_LookupMember_STRoo82. 83

Programming Example. 84

Graphic Symbols 85

 FactoryTalk® View ME/SE Graphic Symbols 85

 Studio 5000 View Designer® Graphic Symbols. 85

Faceplates 89

 Home Tab 90

 Trend Tab 91

 Configure Tab 91

 Diagnostics Tab. 95

 Fault Warning Tab 96

Application Code Manager 97

 Definition Object: raC_Dvc_HI_1756xWS. 97

 Implementation Object: raC_LD_Dvc_HI_1756xWS 97

 Linked Libraries 98

 Configured HMI Content 98

 Attachments 98

Chapter 7

**HI5069 - CompactLogix® Weigh
Scale Module
(raC_Dvc_HI_5069xWS)**

Overview 101

Functional Description 101

Required Files 101

 Controller Files 102

 FactoryTalk® View HMI Files 102

 Studio 5000® Application Code Manager Files 102

Device Definition 103

 HARDWARE Definition 103

Operations. 104

 Faults & Warnings 104

 Execution 105

Add-On Instruction I/O Data 105

 Add-On Instruction Ladder Implementation. 105

 InOut Data 106

 Input Data. 106

 Output Data. 108

Data Types. 111

 raC_UDT_ItfAD_Hardy_CtrlSet. 111

 raC_UDT_ItfAD_Hardy_CtrlCmd 112

 raC_UDT_ItfAD_Hardy_CtrlSts 113

 raC_UDT_Event 114

 raC_UDT_HI_5069xWS_Cfg. 114

 raC_UDT_HI_5069xWS_Inp. 115

raC_UDT_Dropdown	116
.....	117
raC_UDT_LookupMember_STRo082.....	117
Programming Example.....	118
Graphic Symbols	119
FactoryTalk® View ME/SE Graphic Symbols	119
Faceplates	119
Home Tab	120
Trend Tab	121
Configure Tab	121
Diagnostics Tab.....	125
Integrated Technician Tab	125
Fault Warning Tab	126
Application Code Manager	128
Definition Object: raC_Dvc_HI_5069xWS	128
Implementation Object: raC_LD_Dvc_HI_5069xWS	128
Linked Libraries	129
Configured HMI Content	129
Attachments	129

Device Object Libraries Overview

Our Device Object Libraries enable you to easily interface with Rockwell Automation® intelligent devices like drives, motion, network switches, sensors, IO and more. The libraries contain tested, documented, and life-cycle managed objects which can be used with machine builder, process, and packaged libraries or as standalone components. Device objects include HMI faceplates for FactoryTalk® View ME/SE, FactoryTalk® Optix and Studio 5000 View Designer® software and provide a user interface that seamlessly integrates with the products.

HMI faceplates are standard display files that provide a common user interface. These are HMI pop-up screens used to display detailed information related to a specific instruction or device. In systems that follow ISA 101.1 design guidelines, faceplates are often referred to as Level 4 displays.

Pre-configured Device Objects include an Add-On Instruction Rung and an HMI Faceplate providing the following benefits:

- Collect, Process and Deliver Data between Smart Devices and Application Logic
- Detailed Device Data Collection and Delivery
- Enhanced Device Status and Diagnostics
- Common Control Interfaces maximizing Flexible Automation Device Selection & Application Code Reuse

Device Object Use Cases:

- Basic Device Maintenance and Diagnostics
- Virtual Device Operations for Startup and Commissioning
- Operator and Program Control for Velocity Machine and Process Applications



Device Object Libraries may be downloaded from the [Product Compatibility and Download Center](#). Search for "Library".

Application Code Manager

Studio 5000® Application Code Manager is a tool that can be used with Device Object Libraries to streamline project and machine development. This bulk coding tool allows you to easily design and standardize functionality with reusable application code.

Enable more efficient project development with reusable libraries of code:

- Quickly create and deploy projects through our Application Content Libraries
- Import Rockwell provided application content libraries to expedite system development

- Build your own reusable code that can be managed and deployed across your entire enterprise
- Easily configure objects in bulk with reusable code to increase application development, no additional programming is necessary
- Consolidate content for Studio 5000® Logix Designer, FactoryTalk® View Studio, FactoryTalk® Alarms & Events, FactoryTalk® Historian to configure an object a single time and generate content for each of those software packages.

See the section on [Using the Library with Application Code Manager](#) for more details.

Other Application Code Libraries

This Device Object Library may be used in harmony with other Application Code Libraries including other Device Object Libraries (Network, IO, IO-Link, Safety Device Libraries) or Application Libraries (PlantPAX® Process Objects library, Machine Builder Libraries). All libraries are intended to follow similar design philosophies to provide a consistent experience for operators and maintenance staff.

A complete list of Application Code Libraries from Rockwell Automation® follows.

Item	Description
PlantPAX® Process Library	Rockwell Automation® Library of Process Objects provides application templates, Endress + Hauser library objects, Application Code Manager library objects, and tools and utilities for PlantPAX® DCS applications. Includes the following: <ul style="list-style-type: none"> • Graphics for built-in instructions • HMI images and Help files • Logix diagnostic objects • Process objects • Control strategies • Sequencer objects • PlantPAX® Configuration Tools for Tags, Alarms and Historian • Color Change • Historian -- Asset Framework template and objects
Machine Builder Libraries	Tested, documented and life-cycle managed library objects and faceplates for use with Studio 5000® Application Code Manager for use primarily with OEM and discrete machine applications.
Common Application Libraries	Commonly used application library objects and faceplates for use with Studio 5000® Application Code Manager including basic functions like unit conversion and data collection.
Independent Cart Technology Libraries	ICT Libraries for iTRAK® and MagneMotion® including MagneMover LITE® and QuickStick® for Studio 5000® Application Code Manager
I/O Device Library	Provides objects for Rockwell Automation® 1756, 1769, 1734, 1794, 1738, 1732E, 1719, 5069, 5094 I/O modules including pre-configured status and diagnostic faceplates
IO-Link Device Library	Provides IO-Link master and sensor objects including pre-configured status and diagnostic faceplates
Network Device Library	Provides objects for Stratix® switch and Device Level Ring network objects
Power Device Library	Provides objects for discrete, velocity, motion, and PowerMonitor™ devices
Safety Device Library	Provides safety objects to interface with safety I/O
Condition Monitoring Device Library	Provides Dynamix™ -1444 module and machinery Condition Monitoring applications such as motors and pumps. This includes FactoryTalk View® SE HMI faceplates and Studio 5000® Application Code Manager implementations.
Electrical Protection Device Library	Provides a standard to represent protection devices within your electrical distribution system
Mettler Toledo Device Library	Provides Mettler Toledo Weighing Terminal Objects including pre-configured status and diagnostic faceplates
Hardy Device Library	Provides Hardy Weighing scale module including pre-configured status and diagnostic faceplates

Libraries can be accessed from the [Product Compatibility and Download Center](#).

Software and Firmware Upgrades

When you update software or firmware revisions, we recommend that you verify the impact on performance and memory utilization before implementing the upgrade on the production system. For FactoryTalk® View or ControlLogix® platforms, we recommend that you review the release notes and verify the impact of the upgrade on performance and memory utilization.

You can also verify the compatibility of the upgrade with the installed software and operating systems in use on your system. See the [Product Compatibility and Download Center](#).

Rockwell Automation® Services and Support

System Support offers technical assistance that is tailored for control systems. Some of the features include the following:

- Highly experienced team of engineers with training and systems experience
- Use of online remote diagnostic tools
- Access to otherwise restricted TechConnectSM Knowledgebase content
- 24-hour, 7 days per week, 365 days per year of phone-support coverage upgrade option

For more information, contact your local distributor or Rockwell Automation® representative or see <http://www.rockwellautomation.com/support>.

You can view or download publications at <http://www.rockwellautomation.com/literature>. To order paper copies of technical documentation, contact your local Allen-Bradley® distributor or Rockwell Automation® sales representative.

Rockwell Automation® Hardy Device Library

The Hardy Device Library is a tested, documented, and life cycle managed object library. The Device Library provides pre-configured status and diagnostic faceplates and AOI sets for HI1756-ControlLogix® & HI5069-CompactLogix® Weigh Scale Module. The Hardy Device Objects may be used with Machine Builder, Process, and Packaged Libraries or as standalone components. Hardy Device Library add-on instructions objects collect, process, and deliver data between hardware devices and application logic.

The Hardy Device Library includes Add-On Instructions (AOIs) and HMI Faceplates for HI 1756 & HI 5069 Devices.



The Hardy Device Library may be downloaded from the [Product Compatibility and Download Center](#). Search for Hardy Device Library.

Compatibility

Compatible Software

- HI1756-ControlLogix requires Studio 5000 Logix Designer® v31.02 or later for PAC Application Development
- HI5069-CompactLogix requires Studio 5000 Logix Designer® v33.00 or later for PAC Application Development
- Studio 5000® Application Code Manager v4.01 and later for bulk code configuration
- Studio 5000 View Designer® v8.00 and later for PanelView™ 5000 Application Development - “Only compatible with 1756 HI-WS device objects”
- FactoryTalk® View Studio v10 and later for PanelView™ Plus or FactoryTalk® View SE Application Development
- FactoryTalk® Optix 1.5 or later - “Only compatible with 1756 HI-WS device objects”

Compatible Hardware

- PanelView™ 5500 with v8 or later firmware
- PanelView™ Plus with v10 or later firmware
- FactoryTalk® Optix Panel
 - a. OptixPanel Standard 10.1” or larger
- ControlLogix® 5570/5580 controller or CompactLogix™ 5370/5380 Controller with v3.01 or later firmware
- HI 1756-WS & HI 1756-2WS with Series A or later firmware

- HI 5069-WS & HI 5069-2WS with Series A or later firmware

IMPORTANT FactoryTalk View HMI faceplates are not compatible with FactoryTalk® ViewPoint

Summary of Changes

This publication contains the following new or updated information. This list includes substantive updates only and is not intended to reflect all changes.

Topic	Page
Reference manual entirely reformatted	all
Added raC_Dvc_HI_5069xWS chapter	97
Added Import HMI Tags section	35

Footprint

Each instruction requires memory footprint within the Logix controller. The following characteristics apply:

- **Definition:** Estimated memory required to store the object definition, including all dependents
- **Instance:** Estimated memory required per object instantiated.
- **Execution (L85):** Estimated execution time / scan footprint evaluated in 1756-L85 PAC

Device Object Footprint

Device Object	Defintion (kB)	Instance (kB)	Execution (µs)
raC_Dvc_HI_1756xWS	87.068	10.2	103
raC_Dvc_HI_5069xWS	125.552	11.973	108

Additional Resources

These documents contain additional information concerning related products from Rockwell Automation.

Resource	Description
Rockwell Automation Library of Process Objects Reference Manual PROCES-RM200	Describes the Add-On Instructions, PlantPAx instructions, and associated faceplates that are available to develop applications.
Application Code Manager User Manual LOGIX-UM003	Studio 5000® Application Code Manager user manual.
EtherNet/IP Network Devices User Manual, publication ENET-UM006	Describes how to configure and use EtherNet/IP devices to communicate on the EtherNet/IP network.
Hardy 1756-xWS User Manual HI 1756-Weight Scale Module	Provides installation instructions, wiring diagram, configuration, and specifications.
Hardy 5069-xWS User Manual HI 5069-Weight Scale Module	Provides installation instructions, wiring diagram, configuration, and specifications.

Library Components

The Hardy Device Library is a tested, documented, and life cycle managed object library. The Device Library provides pre-configured status and diagnostic faceplates and AOI sets for HI1756-ControlLogix® & HI5069-CompactLogix® Weigh Scale Module. The Hardy Device Objects may be used with Machine Builder, Process, and Packaged Libraries or as standalone components. Hardy Device Library add-on instructions objects collect, process, and deliver data between hardware devices and application logic.

Hardy Device Instructions

The Hardy Device Library includes instructions to interface with Weigh Scale Modules. The HI1756-ControlLogix® and HI5069-CompactLogix® Weigh Scale Module Series A is configured for both single-channel operation and dual-channel operation. Both modules can be used for a wide variety of process weighing applications such as batching, blending, filling/dispensing, check weighing, force measurement, level by weight and weight rate monitoring. The analog-to-digital converter in the weigh module controller updates fifty times per second and is capable of 8,388,608 counts of display resolution. This is enough to provide accurate weight measurement and control and to tolerate large “dead” loads or over sizing of load cells. The weigh module analyzes the performance of each individual load cell and determines the total number of load cells in the system. C2 calibration can be initiated from the C2 calibration screen.

The instruction included is as follows:

- [HI1756 - ControlLogix Weigh Scale Module](#)
- [HI5069 - CompactLogix Weigh Scale Module](#)

There are two type of instruction in this library:

- Device (Dvc): instruction used for devices (e.g. HI1756-ControlLogix® Weigh Scale Module)
- Device (Dvc): instruction used for devices (e.g. HI5069-CompactLogix® Weigh Scale Module)

Hardy Device Instructions

Instruction	Version	Category	Instruction Description
raC_Dvc_HI_1756xWS	1.01	Weigh Scale Module	HI1756-ControlLogix® Weigh Scale Module
raC_Dvc_HI_5069xWS	1.01	Weigh Scale Module	HI5069-CompactLogix® Weigh Scale Module

Library Folders and Files

When you extract the library from the downloaded .zip folder, you will find the following folder and file structure. Note that some items are generalized with *TYPE* (e.g. Dvc, Opr, Tec) and *OBJECT* (e.g. HI1756-ControlLogix® Weigh Scale Module). The major and minor versions are represented by X and Y respectively.

Level 1	Level 2	Level 3	File Type	Description
Application Example			Folder	Application Example Files
	HardyApplication_ACM_v1_01.xlsx		XLSX	Application Code Manager Project
	HardyApplication_v1_01.ACD		ACD	Logix Designer Example Project
	HardyApplication_SE_v1_01.apa		APA	FT View SE Project Archive
	HardyApplication_ME_v1_01.apa		APA	FT View ME Project Archive
	HardyApplication_VD_1_01.vpd		VPD	View Designer Project File
	HardyApplication.Optix.Z		Z	FT Optix Project File
ApplicationCodeManagerLibraries			Folder	Application Code Manager files
	Attachments (.HZ1 and .txt files)		Folder	ACM Object Attachments
	(RA-LIB)_Device_Asset-Control_GROUP_raC_Dvc_OBJECT_(X.Y).HSL4		HSL4	ACM Asset-Control Object
	(RA-LIB)_Device_Device_GROUP_raC_Dvc_OBJECT_(X.Y).HSL4		HSL4	ACM Device Object
HMI - FactoryTalk View ME			Folder	FactoryTalk® View ME files
	Displays - gfx		Folder	FT View ME display files
		(raC-X_YY-ME) raC_TYPE_OBJECT-faceplate.gfx	GFX	Object Faceplate display
	Global Objects - ggfx		Folder	FT View ME Global Object files
		(raC-X-ME) Graphic Symbols - Hardy Device.ggfx	GGFX	Graphic Symbol/Launch Button global objects
(raC-X-ME) Toolbox - Hardy Device.ggfx		GGFX	Toolbox global objects	
HMI - FactoryTalk View SE			Folder	FactoryTalk® View SE Files
	Displays - gfx		Folder	FT View SE display files
		(raC-X_YY-SE) raC_TYPE_OBJECT-faceplate.gfx	GFX	Object Faceplate display
	Global Objects - ggfx		Folder	FT View SE Global Object files
		(raC-X-SE) Graphic Symbols - Hardy Device.ggfx	GGFX	Graphic Symbol/Launch Button global objects
(raC-X-SE) Toolbox - Hardy Device.ggfx		GGFX	Toolbox global objects	
HMI - ViewDesigner - vpd			Folder	Studio 5000 View Designer® Files
	(raC-X_YY-VD) raC_Dvc_Hardy.vpd		VPD	Object faceplate and graphic symbol/launch buttons
FactoryTalkOptixLibraries				FactoryTalk® Optix Library Files
	HardyDevice_vXR			Library Folder
		HardyDevice_vXR.Optix		FT Optix Library Application
HMI FactoryTalk View Images - png			Folder	FT View ME/SE image files
	images.png		PNG	FTView ME/SE images
Reference Manuals			Folder	Manuals
	DEVICE-RM915A-EN-P.pdf		PDF	Reference manual
Studio 5000 Logix Designer Files - L5X			Folder	Studio 5000® AOI and RUNG import files
		raC_TYPE_OBJECT_X.YY_RUNG.L5X	L5X	Object rung import
		raC_TYPE_OBJECT_X.YY_AOI.L5X	L5X	Object AOI import

Help Files - pdf		Folder	Documents
	Help raC-1 Hardy.pdf	PDF	Help File
FTViewStudio_HardyLibrary_Tags_1_01.csv		CSV	FTView ME/SE HMI Tags
ReadMe.txt		TXT	Explanation of setup.cmd
Setup.cmd		CMD	Application Code Manager setup script to register library



See the files in the *Application Example* folder to see a functional application that uses all of the Hardy Device Library instructions. These files are referenced in the Programming Examples for each instruction. The files include a Studio 5000 Logix Designer® controller file, a Studio 5000® Application Code Manager project back-up, and an HMI projects for FactoryTalk® View SE Local Station, FactoryTalk® View SE Local Station, and Studio 5000 View Designer®.

Visualization Files

Each Add-On Instruction has associated visualization files that provide a common user interface. The Hardy Device Library supports three HMI options each with their own files supplied:

- FactoryTalk® View ME (Machine Edition)
- FactoryTalk® View SE (Site Edition)
- Studio 5000 View Designer®
- FactoryTalk® Optix

FactoryTalk® View Visualization Files

You must import these files in the following order:

- Images (.png files)
- Global Objects(.ggfx file type)
- HMI faceplates (.gfx file type)

File Type Abbreviations	FactoryTalk® View SE	FactoryTalk® View ME	Description
Images (.png)	All .png files in the <i>HMI FactoryTalk® View Images - png</i> folder. IMPORTANT: FactoryTalk® View application renames PNG files when they are imported with a .bmp file extension, but the files retain a .png format.		Common icons that are used in the Global Objects and standard displays for all objects.
Global objects (.ggfx)	(raC-1-SE) Graphic Symbols - Hardy Device.ggfx	(raC-1-ME) Graphic Symbols - Hardy Device.ggfx	Graphic symbols or launch buttons used to open faceplate displays from other displays.
	(raC-1-SE) Toolbox - Hardy Device.ggfx	(raC-1-ME) Toolbox - Hardy Device.ggfx	Common objects used across multiple device faceplates.
Standard displays (.gfx)	(raC-1.00-SE) precedes name of the display.	(raC-1.00-ME) precedes name of the display.	e.g. (raC-1.00-SE) raC_Dvc_HI_1756xWS-Faceplate.gfx

Global object files contain Graphic Symbols that are created once and referenced multiple times on multiple displays in an application. When changes are made to a global object, all instances in the application are automatically updated.

Global objects serve two purposes:

- Toolbox files contain common elements that are used to build faceplate displays.
- Graphic Symbols files contain device symbols or launch buttons that you can use to build your application displays. Select the symbol to open the corresponding faceplate display.

Standard display files, commonly called faceplates, provide a common user interface.

Studio 5000 View Designer® Visualization Files

Studio 5000 View Designer® project files are supplied that contain faceplates and launch buttons for the Hardy Device Library. The devices are distributed over multiple Studio 5000 View Designer® Project files. These files are found in the *HMI - ViewDesigner - vpd* folder. Inside of the VPD file you will find a the required display files inside of the *User-Defined Screens* folder.

Display Type	View Designer Screen	Description
Screen	Toolbox	Graphic symbols or launch buttons used to open faceplate/pop-up displays from other displays.
Pop-Up	raC_Dvc_ precedes name of the pop-up.	Faceplate display for specific device. e.g. raC_Dvc_HI_1756xWS_FP

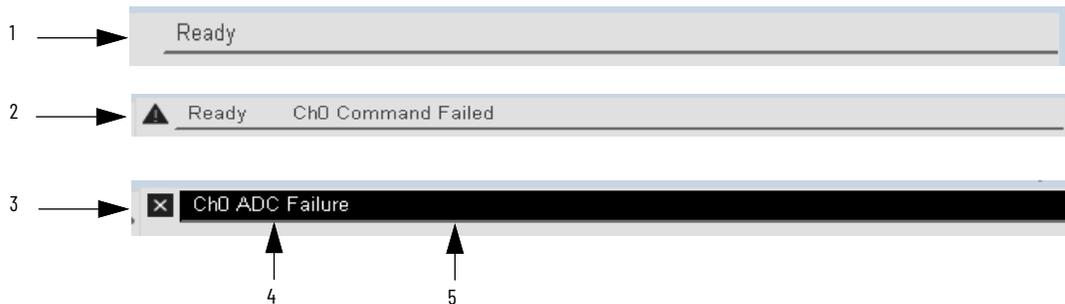
Basic Faceplate Attributes

Faceplates consist of tabs, and each tab consists of one or more pages. The Home tab is displayed when the faceplate is initially opened. The faceplate provides the means for operators, maintenance personnel, engineers, and others to interact with a device or instruction instance, which includes a view of its status and values. Faceplates may also manipulate an instruction through its commands and settings. Select the appropriate icon on the left of the faceplate to access a specific tab. This section provides an overview of the faceplate attributes that are common across the objects. More details are supplied in the individual section for each object.

Common Status Banner

At the top of all device object faceplates there is a common status banner which provides the following information:

- Ready status. Shown if the device is connected and ready in physical mode with no faults.
- Faulted (banner will show Not Ready with fault message)
- Warning status and reason.



Item	Description
1	Ready state displays Ready Status.
2	Warning state shows the reason with warning icon.
3	Faulted state shows Fault and fault reason with icon.
4	Fault message for latest fault present.
5	Faulted state shows Black background.

Faceplate Navigation

All device object faceplates have navigation tabs on the left side of the faceplate. Navigation tabs may vary based on device type. The active tab will show as a light grey, while an inactive tab will show as a dark grey.



Active Tab



Inactive Tab

The common tabs are shown below.



Home Tab



Diagnostics Tab



Trend Tab



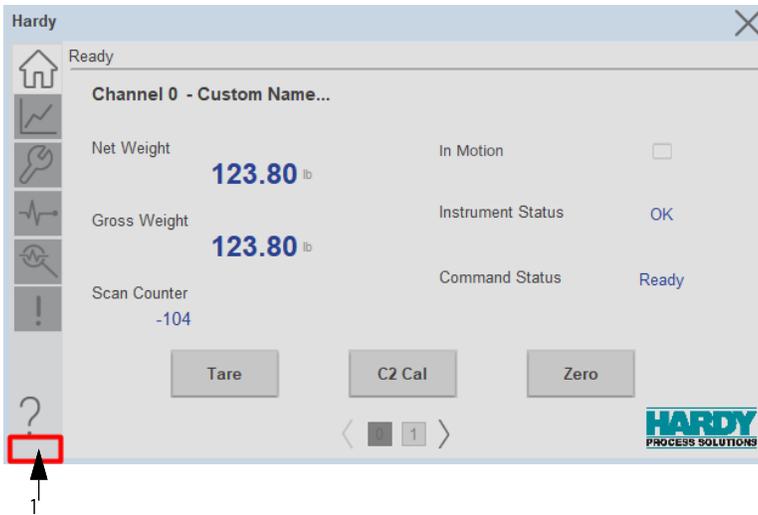
Fault Tab



Configuration Tab

Faceplate Revision Notes

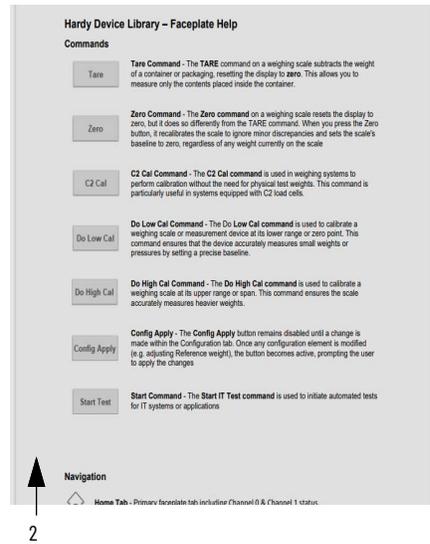
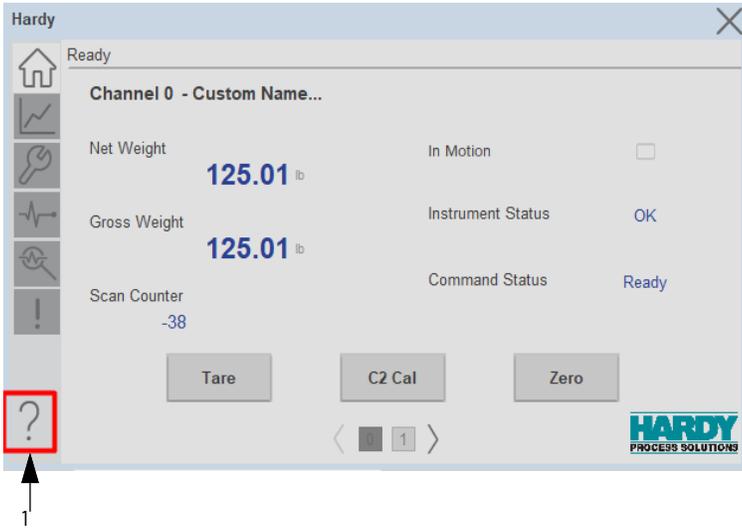
By clicking on the open space near the bottom left corner of the faceplate you can momentarily view revision notes and details of the active faceplate. This may be useful in troubleshooting or when communicating with Rockwell Automation® Tech Support.



Item	Description
1	Click the bottom right corner to temporarily open up the revision notes dialogue
2	Revision number
3	Faceplate display name

Faceplate Help

By clicking on the “?” icon in the bottom left of the faceplate you can open the Help file document of the faceplate. This may be useful in understanding the functionality of commands, navigation, status indicators.

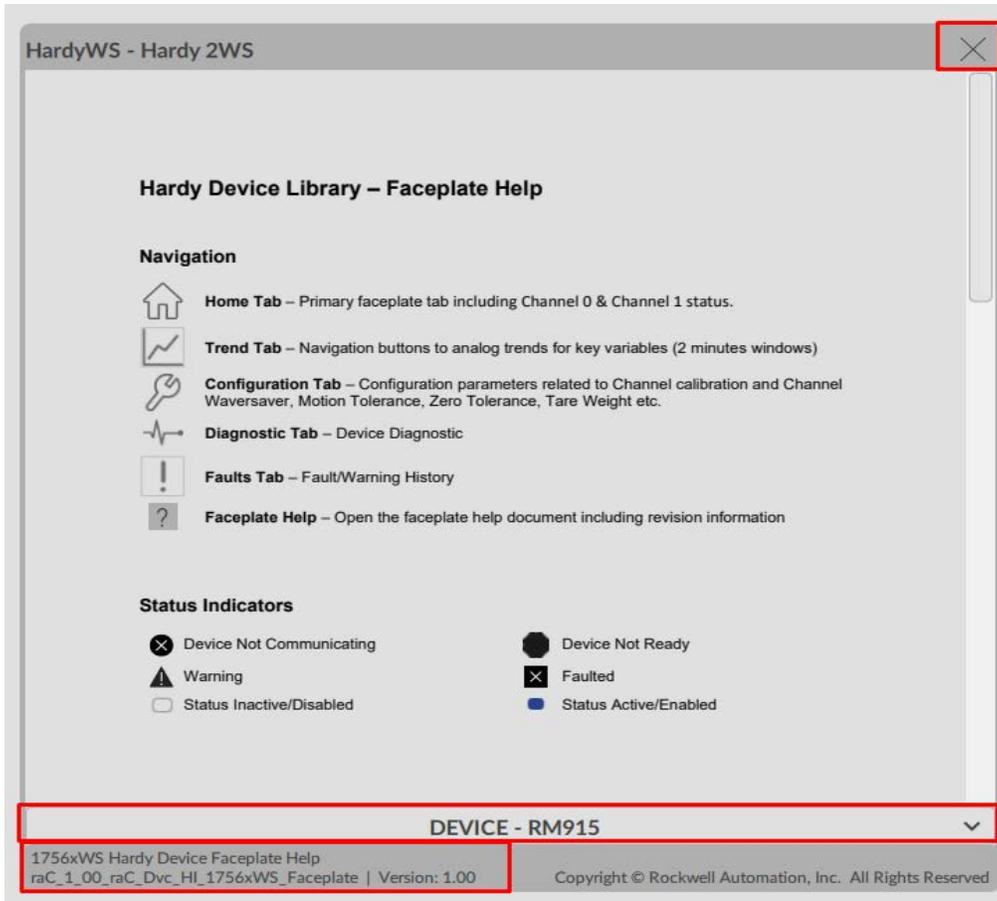


Item	Description
1	Click on the “?” icon to open Help File.
2	Help File pdf

Additionally, The Optix Faceplates are provided with the Help Button for each faceplates. Help Button is located on the upper right corner of Faceplate frame as shown in image below,



When user click on the Help Button it will open a pop-up display which includes a Help Document and Accordion which contains web browser to access Reference manual from web. See in Image provided below.



Item	Description
1	Help Pop-Up window with Close button
2	Help Document
3	Accordion which contains Reference Manual web browser (Status: - Not Expanded)
4	Revision Information
5	Accordion which contains Reference Manual web browser (Status: - Expanded) shown in image below. Note: Active Internet connection is required to see Reference Manual.

Launch Buttons

Launch buttons are provided in Global Display (GGFX) files for FactoryTalk View® ME/SE as well as in Studio 5000 View Designer® projects. These are used to open HMI faceplate displays or pop-ups. Two types of launch buttons are provided:

Launch Button Style	Image Examples	Usage
Basic Text Button		Simple launch button with diagnostic information.

Diagnostic Icons

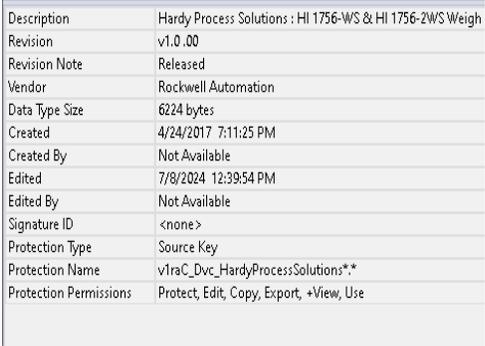
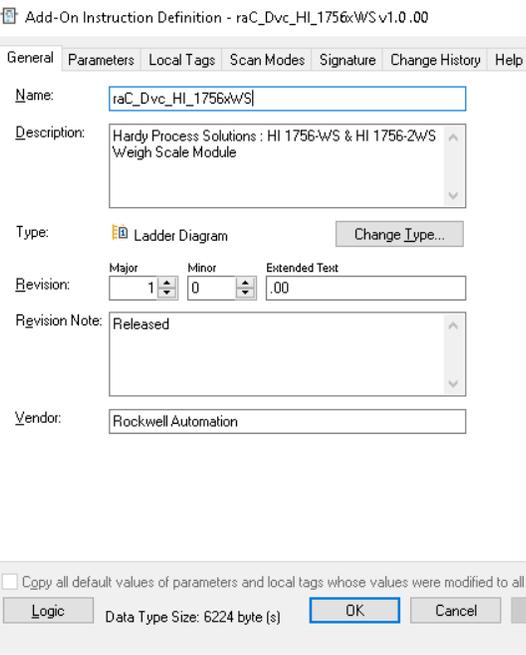
Diagnostic icons may be displayed on the graphic buttons for compatible modules. Safety modules are designated with a small guard icon.

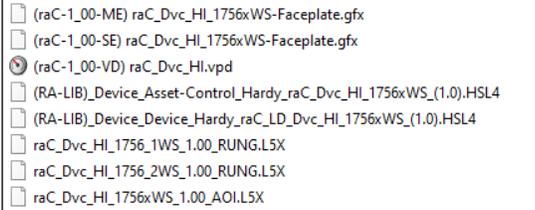
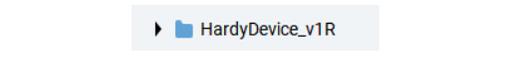
Icon	Image	Visible Condition
Communications Failure		Connection Faulted
Fault		Any device fault active (module hardware issue)
Warning		Any device warning active (maintenance required)
Not Ready		Device Not Ready

Library Versions

Each library object has a revision x.yy where: x is the Major Revision number and yy is the Minor Revision number. Each release of the library comes with release notes that describe the changes that were made since the last release.

You can find the revision number of the object in a number of locations as shown below.

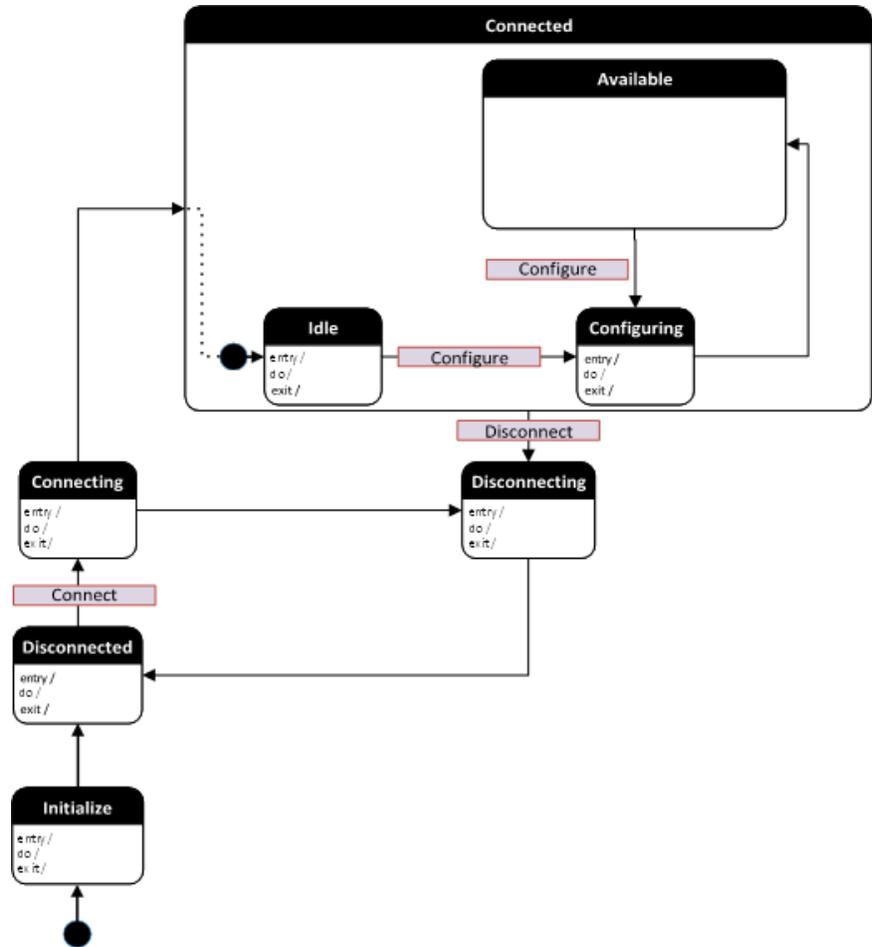
Component	Example
<p>The Add-On Instruction in Logix Designer application has revision information visible when the instruction is selected in the Controller Organizer.</p>	
<p>The Add-On Instruction Definition General tab shows the revision number along with basic revision notes. Refer to the release notes for complete revision notes.</p>	
<p>The faceplate in FactoryTalk® View software has revision information visible when the pointer is clicked just inside the lower left corner of the faceplate.</p>	

Component	Example
<p>The revision number is shown in the file names for GFX, VPD, ACM.HSL4, AOI.L5X, and RUNG.L5X files.</p>	 <ul style="list-style-type: none">  (raC-1_00-ME) raC_Dvc_HI_1756xWS-Faceplate.gfx  (raC-1_00-SE) raC_Dvc_HI_1756xWS-Faceplate.gfx  (raC-1_00-VD) raC_Dvc_HI.vpd  (RA-LIB)_Device_Asset-Control_Hardy_raC_Dvc_HI_1756xWS_(1.0).HSL4  (RA-LIB)_Device_Device_Hardy_raC_LD_Dvc_HI_1756xWS_(1.0).HSL4  raC_Dvc_HI_1756_1WS_1.00_RUNG.L5X  raC_Dvc_HI_1756_2WS_1.00_RUNG.L5X  raC_Dvc_HI_1756xWS_1.00_AOI.L5X
<p>Library object folder where the major (x) and minor (y) versions are used in the folder name e.g. raC_x_yy_raC_Dvc_DeviceName_UI</p>	
<p>The major (x) revision number is used in the Library folder and file name e.g. HardyDevice_vxR Note that only major versions are used at the library level and a library may contain multiple minor versions of different objects.</p>	

State Model

The following section will discuss the state model for Device Object. The figure below shows the core logic states.

By default, each state is active for a minimum of 256us to allow for evaluation of state outside of the ADO instance in the user program



Interfaces

Device object interfaces are intended to provide the application programmer a class based harmonized interface for interacting with the device object from user code. Standard control interfaces are used for passing Information (Inf), Settings (Set), Commands (Cmd) and Status (Sts).

These interfaces are commonly used when interfacing with other Rockwell Automation® application code libraries such as the PlantPAX® Process Objects Library or the Machine Builder Library.

For detailed information on specific interfaces, please refer to the appropriate section in this manual. A list of interface UDTs used in this library follows. Note that OBJECT used in the Inp interfaces is replaced with the specific Hardy device object (e.g. 1756xWS).

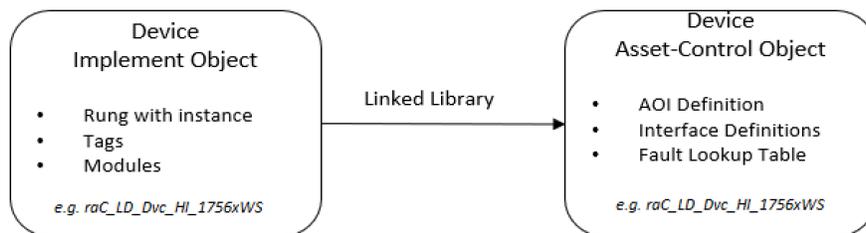
Interface Class	Object Class	Object Sub-Class	Interface Type	Interface Name (UDT)
Control	Hardy	HI-1756xWS & HI-5069xWS	Command	raC_UDT_ItfAD_Hardy_CtrlCmd
			Setting	raC_UDT_ItfAD_Hardy_CtrlSet
			Status	raC_UDT_ItfAD_Hardy_CtrlSts

Application Code Manager Architectural Overview

Device libraries, as with most Application Code Libraries are divided into 2 logical groups: either Asset-Control Object or Device Implement Object.

Asset-Control Objects contain the asset definition of an object and any associated content which belongs to the asset. This includes controller tags, add-on instructions, data types, and attachments such as HMI content and documentation. These are found under the (RA-LIB) Device > Asset-Control folder and have names like *raC_Dvc_xxxx* where *xxxx* is the device name.

Device Implement Objects contain an instance of an asset-control object and provide all related configuration of the asset. The Device implement type is the application code (e.g. programming rung). This includes the required controller tags, programs, modules, and FactoryTalk® View ME/SE symbols. These are found under the (RA-LIB) Device > Device folder and have names like *raC_LD_Dvc_xxxx* where *xxxx* is the device name. LD stands for ladder logic.



HMI Tags

The Hardy Device Library contains "FTViewStudio_HardyLibrary_Tags_1_01.CSV", which is used by FactoryTalk® View Site Edition and FactoryTalk® View Machine Edition applications to operate the Help File from screens. The CSV import file includes the following tags:

Tag Name	Type	Description
RALibrary\HelpFilePath	String	Path name to folder where the help files are stored

Using the Library

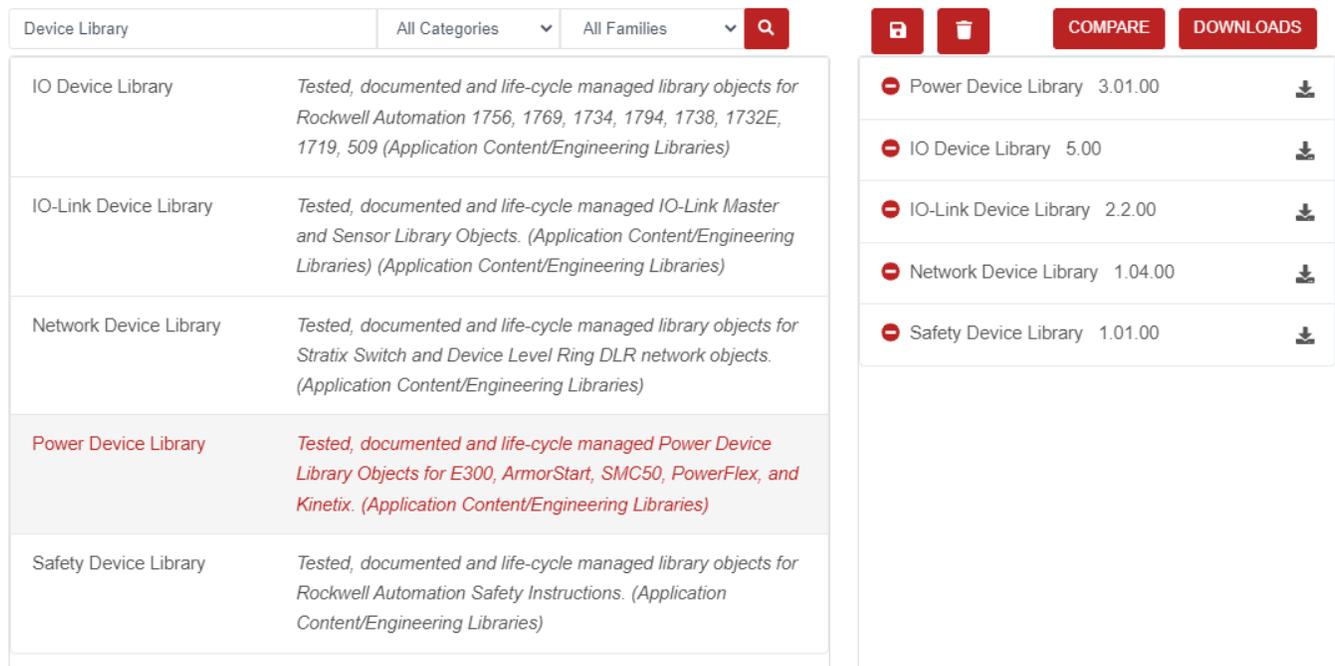
Install the Library

Download the Library

For the latest compatible software information and to download the Rockwell Automation® Library, see the [Product Compatibility and Download Center](#).

Search “Device Library” or filter on Application Content to quickly find the library.

FIND DOWNLOADS 



The screenshot shows a search interface with the following elements:

- Search bar: "Device Library"
- Filters: "All Categories" and "All Families"
- Buttons: "COMPARE" and "DOWNLOADS"
- Results Table (Left):

IO Device Library	<i>Tested, documented and life-cycle managed library objects for Rockwell Automation 1756, 1769, 1734, 1794, 1738, 1732E, 1719, 509 (Application Content/Engineering Libraries)</i>
IO-Link Device Library	<i>Tested, documented and life-cycle managed IO-Link Master and Sensor Library Objects. (Application Content/Engineering Libraries) (Application Content/Engineering Libraries)</i>
Network Device Library	<i>Tested, documented and life-cycle managed library objects for Stratix Switch and Device Level Ring DLR network objects. (Application Content/Engineering Libraries)</i>
Power Device Library	<i>Tested, documented and life-cycle managed Power Device Library Objects for E300, ArmorStart, SMC50, PowerFlex, and Kinetix. (Application Content/Engineering Libraries)</i>
Safety Device Library	<i>Tested, documented and life-cycle managed library objects for Rockwell Automation Safety Instructions. (Application Content/Engineering Libraries)</i>
- Results List (Right):
 - Power Device Library 3.01.00 
 - IO Device Library 5.00 
 - IO-Link Device Library 2.2.00 
 - Network Device Library 1.04.00 
 - Safety Device Library 1.01.00 

Download & Install Studio 5000® Application Code Manager

Studio 5000® Application Code Manager is free to install from Rockwell Automation’s [Product Compatibility and Download Center](#).

Search “Application Code Manager” and select the item to download.

FIND DOWNLOADS ?

The screenshot shows a search interface for 'Application Code Manager'. The search results are as follows:

Library Name	Description
Independent Cart Technology Libraries	ICT Libraries for iTRAK and MagneMotion including MagneMover LITE, QuickStick for Application Code Manager (ACM) (Application Content/Engineering Libraries)
Machine Builder Libraries	Tested, documented and life-cycle managed library objects and faceplates for use with Studio 5000 Application Code Manager (ACM) (Application Content/Engineering Libraries)
Process Library	RA Library of Process Objects, Application Templates, Application Code Manager Library, Tools & Utilities, and Integration with Endress+Hauser Devices (pre-5.00) (Process Solutions/PlantPAx)
Studio 5000 Application Code Manager	Engineering design productivity tool focused on rapid automation application development leveraging (ACM) (Software/Software)

The right-hand pane shows the selected item: Studio 5000 Application Code Manager 4.02.00, with a download icon highlighted.

Extract the downloaded .zip file by running the 4.xx.00-Studio5000_ACM-DVD.exe executable file. This will extract a new folder containing a Setup.exe file which can be run to begin product installation.

Follow the prompts from the splash screen until installation is complete. Note that a SQL server is required for Application Code Manager. SQL Server Express is offered for free and is included in the Application Code Manager installer.

Register Libraries in Studio 5000® Application Code Manager

It is recommended that you use Studio 5000® Application Code Manager or the Studio 5000® “Import Library Objects” Plug-In Wizard to import device library objects into a Logix 5000 controller project. To use the library in Application Code Manager you must first register the libraries.



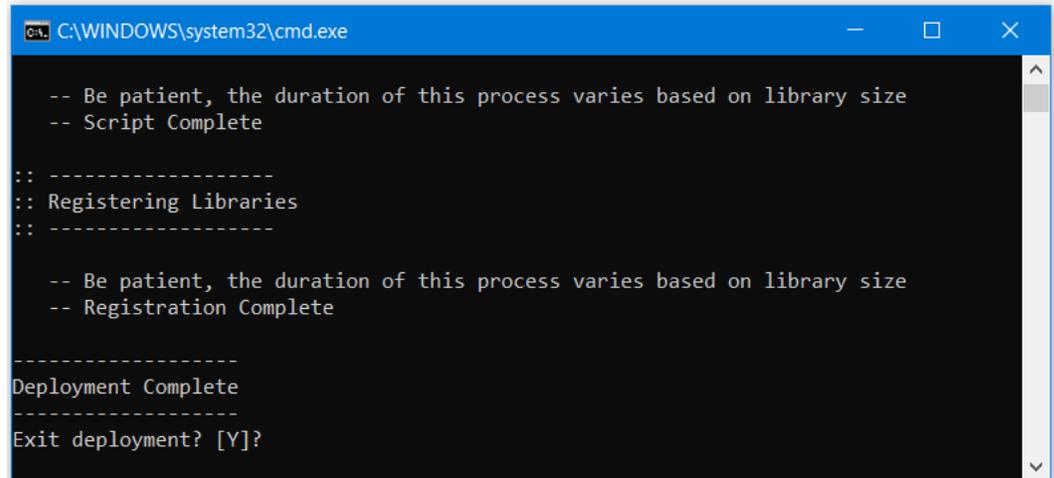
Using Studio 5000® Application Code Manager is not mandatory although it is highly recommended to reduce the likelihood of configuration errors and simplify the workflow. Alternatively, you can import the RUNG.L5X files directly into a Studio 5000® project.



The Lite version of Studio 5000® Application Code Manager is free of charge and can be downloaded from the Product Compatibility and Download Centre. None of the features included in the Standard (paid) version are required to use Device Object Libraries.

Register Complete Library Automatically

To automatically register the entire library, find and run the *setup.cmd* file in the root folder of the library files. You will see a windows console appear as the script runs. When it is complete it will display “Deployment Complete”. Enter “Y” to exist the console.



```

C:\WINDOWS\system32\cmd.exe

-- Be patient, the duration of this process varies based on library size
-- Script Complete

:: -----
:: Registering Libraries
:: -----

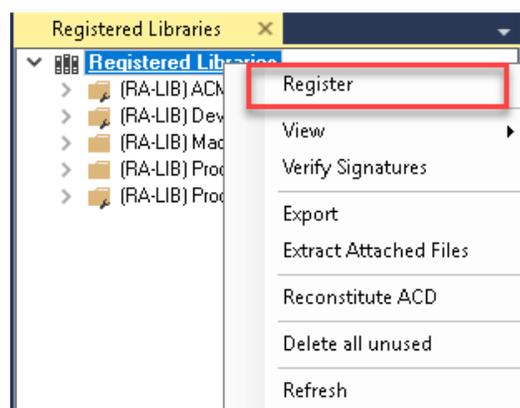
-- Be patient, the duration of this process varies based on library size
-- Registration Complete

-----
Deployment Complete
-----
Exit deployment? [Y]?

```

Register Individual Library Objects Manually

As an alternative to registering the entire library using the *setup.cmd* script, you can manually register one or multiple library objects in Studio 5000® Application Code Manager. Open up Application Code Manager and view the Registered Libraries panel on the right. Right-click on *Registered Libraries* and select *Register*. Browse to the *ApplicationCodeManagerLibraries* folder within the library files and select any HSL4 files that you would like to register. Note you may select more than one at a time. Once you complete registering the desired objects they will be shown under the *(RA-LIB) Device* solution folder.



Importing Logic into Studio 5000® Projects

There are multiple methods to using the logic in a Studio 5000® application. For projects that are being developed from scratch using Application Code Manager along with other Application Code Libraries such as the PlantPAX® Process Objects Library or the Machine Builder Library, you can continue to use the Device Object Libraries in Application Code Manager. For existing applications where devices are being added, it is recommended to use the Studio 5000® Plug-In “Import Library Objects” Wizard. Alternatively you can import the RUNG.L5X files into your program and configure them manually.



It is not recommended to simply import the AOI.L5X files and attempt to build your own logic rung. Doing so will increase the likelihood of configuration errors and likely miss logic that is required outside of the Add-On Instruction.

AOI files should only be imported when updating an existing application from a previous version of a Device Object Library to a newer one.

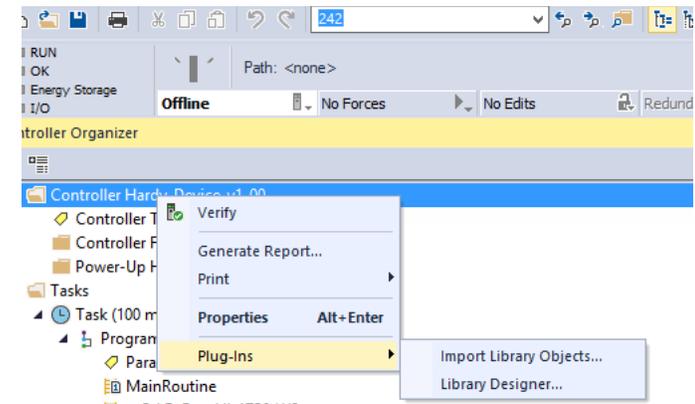
Below is a table to capture recommendations on when to use which tool or workflow when importing and configuring device objects.

Tool/Workflow	Description of when to use	Software Requirements
Application Code Manager (full application)	Project is developed from scratch using Application Code Manager along with PlantPAX® or Machine Builder libraries.	Studio 5000 Logix Designer® Studio 5000® Application Code Manager (Lite)
Studio 5000® Plug-In “Import Library Objects” Wizard	Application Code Manager is installed but not required for the entire project. Application has already been developed but some Device Objects need to be added.	Studio 5000 Logix Designer® Studio 5000® Application Code Manager (Lite)
Import RUNG.L5X File	Application Code Manager is not installed. Application has already been developed but some Device Objects need to be added. Familiar with rung import workflow.	Studio 5000 Logix Designer®
Import AOI.L5X File	Updating existing application that contains an older version of a Device Object AOI.	Studio 5000 Logix Designer®

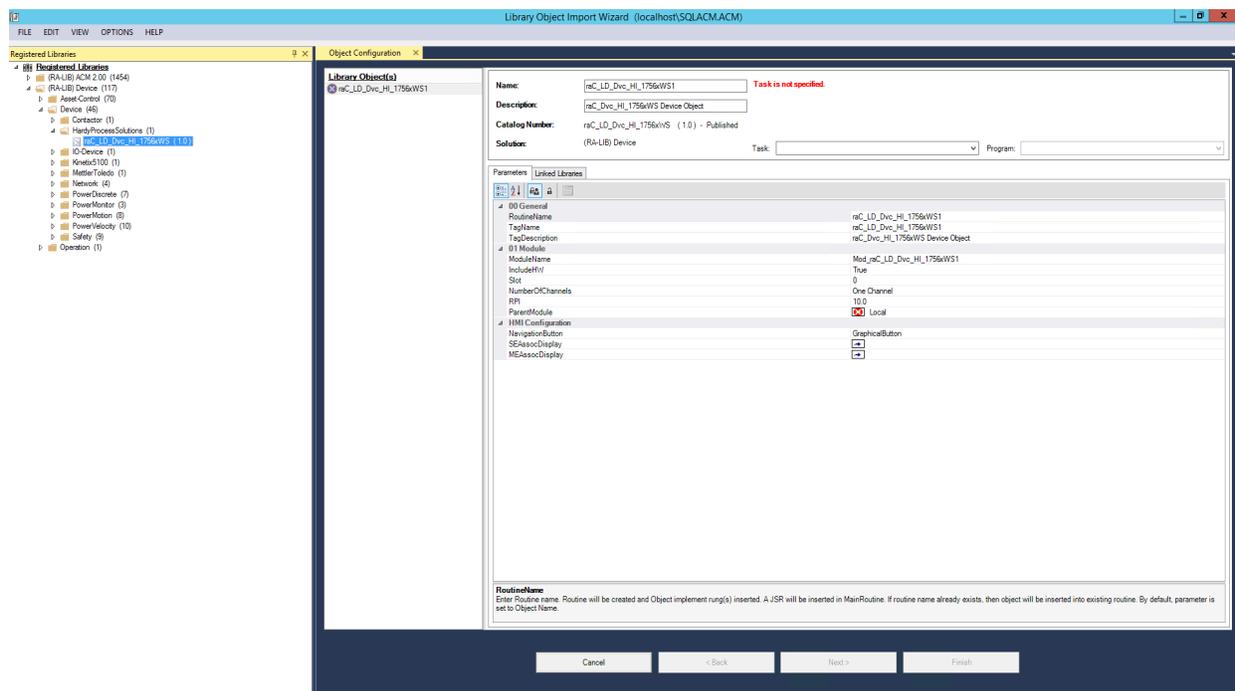
Import Library Objects Wizard

The most simple way to import a Device Object into an existing application is to use the Studio 5000® Plug-In “Import Library Objects” wizard. This plug-in requires Application Code Manager to be installed but does not require it to be open or have a project created.

Right click on an item (e.g. Controller, Task, Program, etc) in the Controller Organizer and select *Plug-Ins > Import Library Objects...*



This will launch a small wizard version of Application Code Manager inside of your Studio 5000 Logix Designer® Project. In the Registered Libraries panel on the left, find your desired object under *Registered Libraries > (RA-LIB) Device > Device* and drag it into the Library Object(s) list in the Object Configuration Tab.



Perform the following configuration:

- Enter a **name** and **description**. Maximum name length can be 22 characters. Note that other parameters such as the RoutineName, TagName, etc will auto-complete based on these fields.
- Assign the **Task** and **Program**.
- Assign the Module (associated hardware e.g. 1756xWS)
 - Set **IncludeHW** to True if you would like the wizard to add a new module (e.g. 1756xWS) to your hardware tree. Set this to False if you already have the module pre-existing in your hardware tree.

- Set the **ModuleName**. If IncludeHW is false, set this to the name of the existing module. If IncludeHW is true, set this to the desired name of the module that will be created.
- Assign a **Slot Number** to the device.
- Select the **NumberOfChannels** and **RPI** Parameters. The *NumberOfChannel* will default to One Channel and *RPI* will default to 10.Oms.
- Set the **ParentModule** to name of the network card that the device is connected to. If using the embedded Ethernet port of the processor module, leave as Local.
- The HMI Configuration options are not used in the Plug-In Wizard and can be ignored.

Name:	HI_1756xWS
Description:	Hardy Device Object
Catalog Number:	raC_LD_Dvc_HI_1756xWS (1.0) - Published
Solution:	(RA-LIB) Device
Task:	Task
Program:	Program

Parameters		Linked Libraries	
00 General	RoutineName	HI_1756xWS	
	TagName	HI_1756xWS	
	TagDescription	Hardy Device Object	
01 Module	ModuleName	Mod_HI_1756xWS	
	IncludeHW	True	
	Slot	0	
	NumberOfChannels	One Channel	
	RPI	10.0	
	ParentModule	Local	
02 HMI Configuration	NavigationButton	GraphicalButton	
	SEAssocDisplay		
	MEAssocDisplay		

NumberOfChannels
Choose the Number of Channels that need to configure on the device.

- Click next or click on the *Linked Libraries* tab. Click the *Auto Create* button to automatically create all of the required linked libraries.
-  You can manually create new linked libraries or point to existing linked libraries if necessary. You may need to do this if you would like to use an older version of library objects when multiple versions are installed in Application Code Manager.
- On the following screen you can select the desired Merge Actions. Generally these can be left with the default actions.
 - Add: used when AOIs don't previously exist in application
 - Overwrite: usually preferred. Used when AOIs previously exist but may or may not be the same revision.
 - Use Existing: used when AOIs previously exist in the application and you do not wish to overwrite the existing items.
 - Click next and you can now see any new logic and modules that will be created.
 - Click Finish to complete the import.

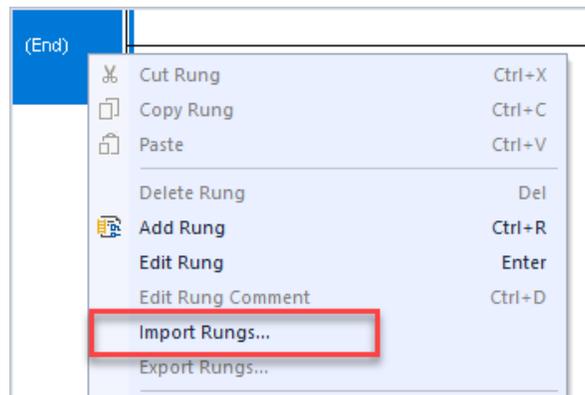
Import Rung Logic

An Add-On Instruction is defined once in each controller project, and can be instantiated multiple times in your application code. To use pre-engineered logic, import each desired RUNG.L5X file into a controller project.

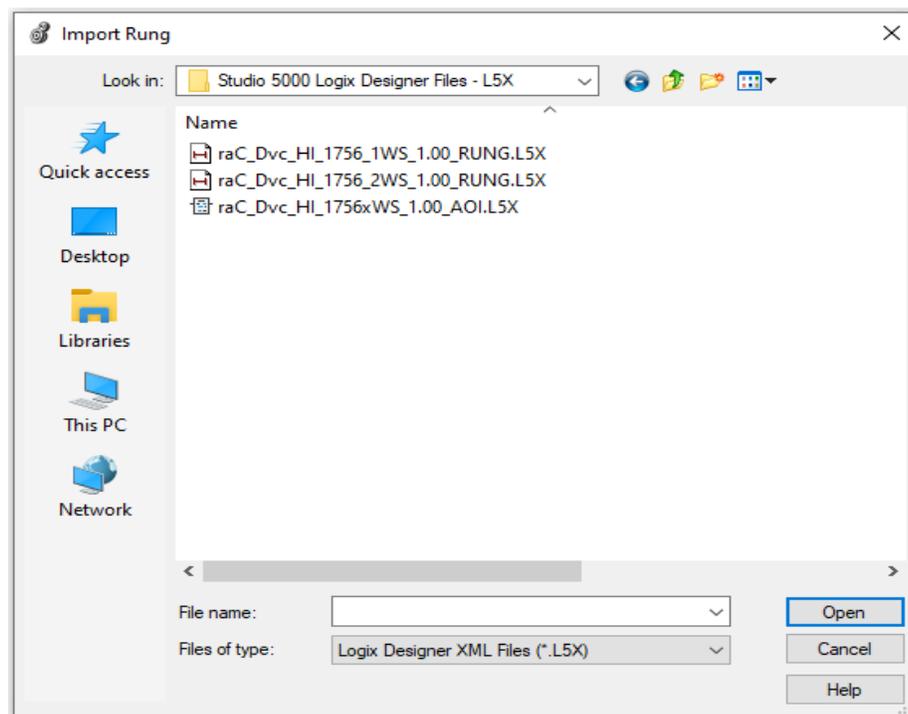
1. In the Studio 5000 Logix Designer® application, open a new or existing project.

IMPORTANT Add-On Instruction definitions can be imported, but not updated, online.

2. Choose or create a new ladder routine to open. Right-click in the routine ladder and choose Import Rungs...



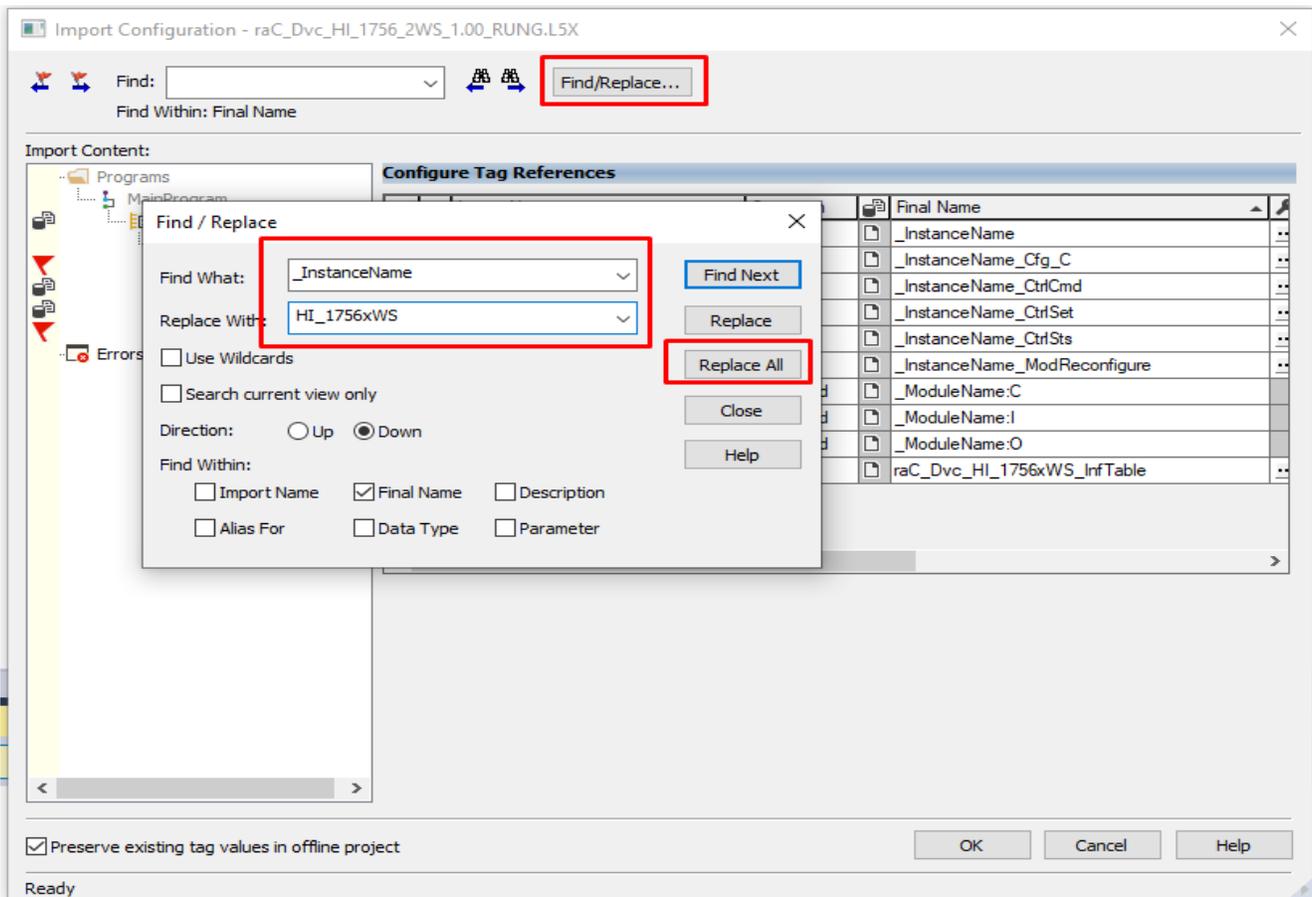
3. Select the desired RUNG and Select Import. The file will have a name like *raC_Dvc_Object_x.yy_RUNG.L5X*.



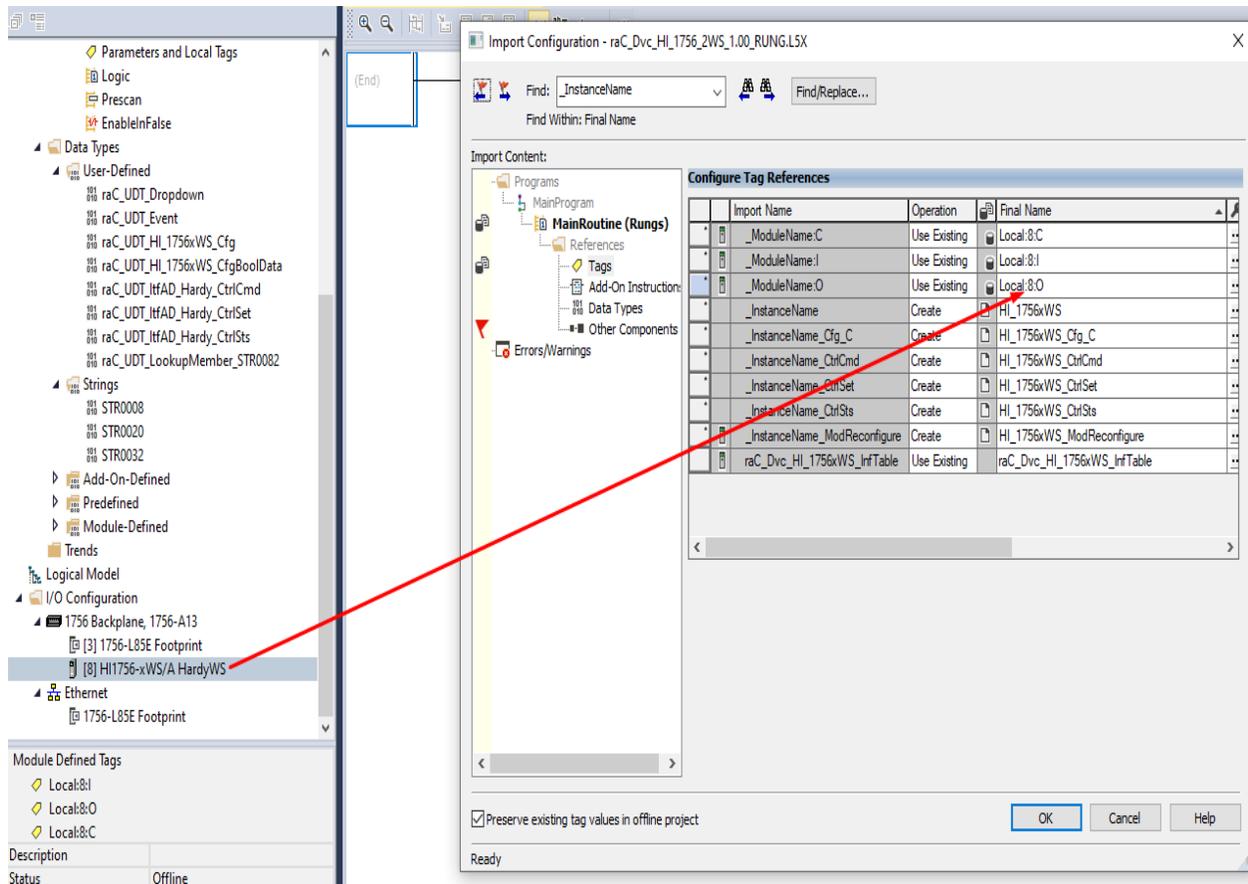
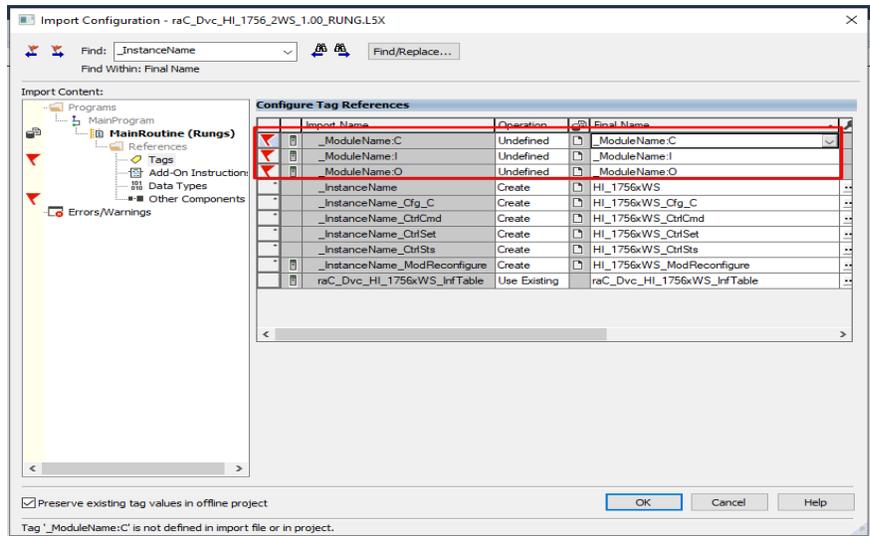


Both “RUNG” and “AOI” .L5X files are provided. Import the RUNG file to get all required additional tags, data types, and message configurations.

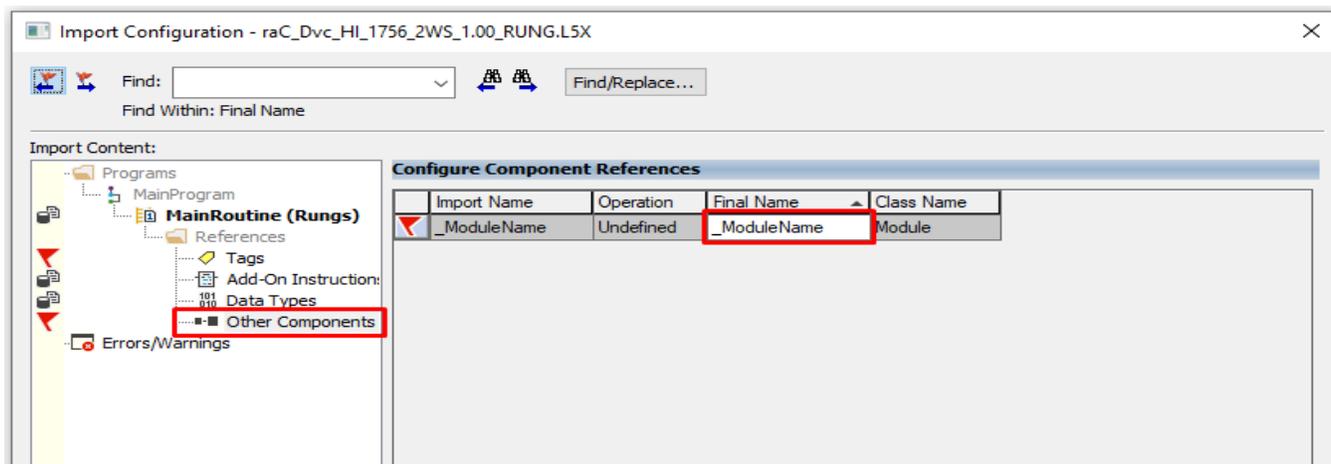
- An *Import Configuration* dialogue window will open and display generic Import names which include “_InstanceName”. Click the *Find/Replace...* button and replace all instances of “_InstanceName” with your desired device name (e.g. “1756xWS_HI1001”).



- You will need to point the new object to the correct module in your project. You can type in or browse for the correct Configuration (:C), input (:I) and Output (:O) tags in your project. In this example our module is called `_ModuleName`.



- Click on the *Other Components* section and type or browse to the name of the module. In this example our module is called Weighing Module.



- The rung will now be imported into your ladder routine.

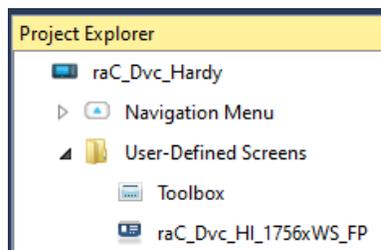
Using Studio 5000 View Designer®

Using View Designer Project Files

Studio 5000 View Designer® may be used for HMI development for PanelView™ 5000 applications. Open up your Studio 5000 View Designer® project alongside a second application instance running the required VPD file in the library folder *HMI - ViewDesigner - vpd*.

You will notice there are two screens available under the *User-Defined Screens* folder:

- Toolbox: This has the graphic symbol launch buttons for the faceplate.
- raC_Dvc_XXXXX_FP: This is a faceplate pop-up screen.



To include these files in your project, perform the following steps:

- Copy the entire faceplate _FP screen from the supplied VPD project to your project application.
- Open the Toolbox screen and copy the desired graphic symbol and paste it into a screen in your project application.

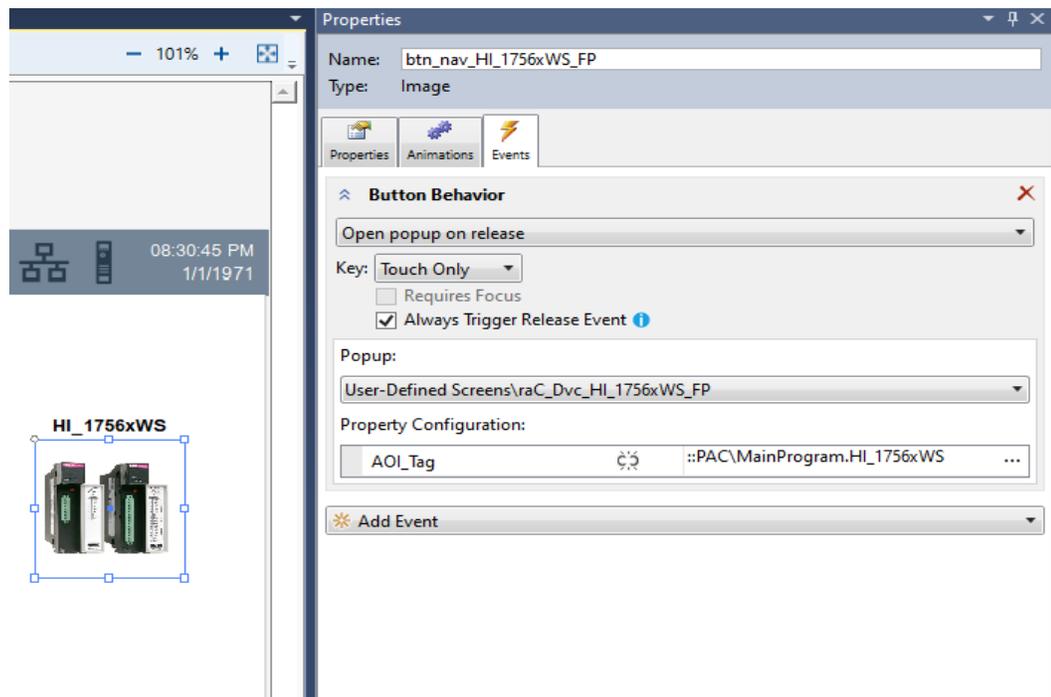
HI_1756xWS



Configuring View Designer Objects

To link the launch button to the faceplate, highlight the button and view the *Events* tab of within the *Properties* pane. Set an Event to *Open popup on release* with the following settings:

- Key: Touch Only
- Popup: Select desired faceplate screen
- AOI_Tag: Browse to AOI backing tag for the device object in your controller file



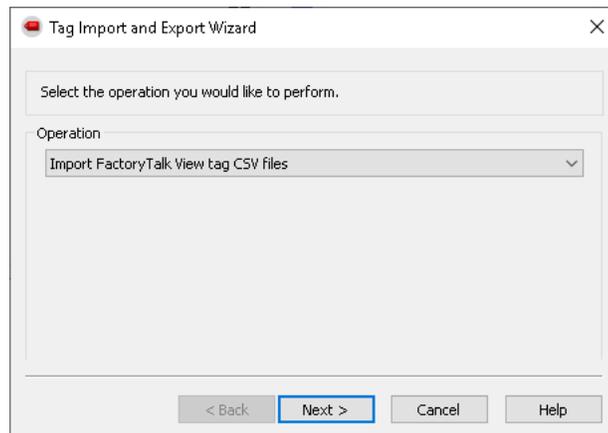
Using FactoryTalk® View Studio

Import HMI Tags

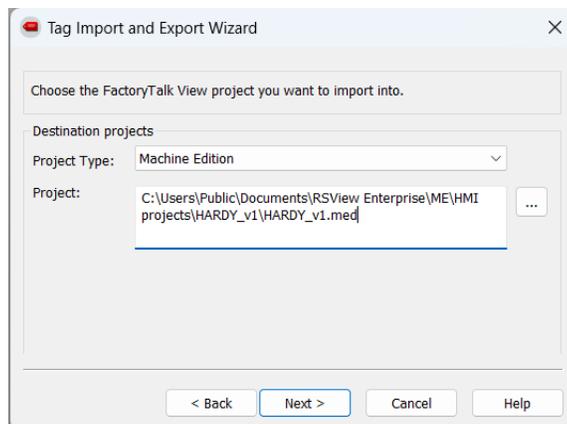
An HMI Tag file *FTViewStudio_HardyLibrary_Tags_1_01.CSV* is provided in the root of the library folder for use with FactoryTalk® View ME and SE. This is required to support opening of Help File.

To import the tag file, in FactoryTalk® View Studio go to *Tools > Tag Import and Export Wizard...*

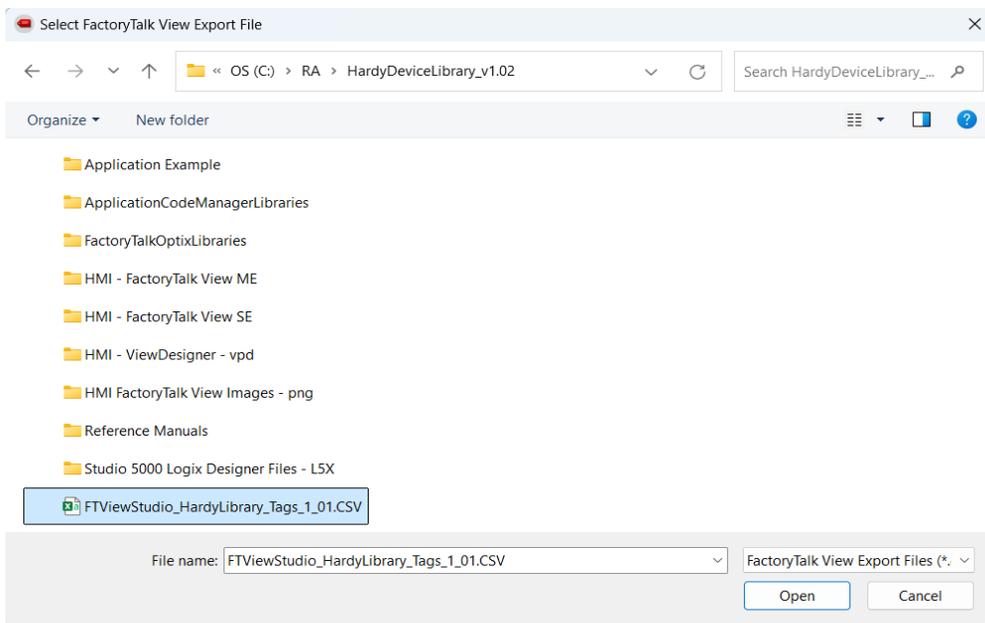
Set the *Operation* to *Import FactoryTalk® View tag CSV files*.



Choose the appropriate *Project Type* (Machine Edition or Site Edition). Browse for the desired project *.MED* or *.SED* file.



Browse for the *.CSV* file in the root of the *HardyDeviceLibrary_v1.xx* library folder.



You may choose to Skip existing or Update existing tags. If this has not been done previously the choice should not make a difference.

The following tags are now imported:

Tag Name	Type	Description
RALibrary\HelpFilePath	String	Path name to folder where the help files are stored

Help File Setup for Machine Edition

The PDF file located in the Hardy Device Library download (HardyDeviceLibrary_v1.xx) must be accessible to the PanelView Plus. They can reside directly on the PanelView Plus or an attached USB drive or SD card. If you decide to move these files directly to the PanelView Plus, you will need direct access to the desktop, review [QA10515 - PanelView Plus 6 and PanelView Plus 7: Enabling Desktop Access](#). After gaining access to the Desktop by selecting Exit on the FactoryTalk View ME Station display, you will use the file manager (similar to Windows 7) to copy the files.

Enter the Help file path location in the RALibrary\HelpFilePath tag located in the FactoryTalk View ME HMI tag database. The following are examples of the possible locations where a folder called Help_Files_Folder was created and the PDF help files were stored:

USB Drive..... \USB Storage\Help_Files_Folder

SD Card..... \Storage Card2\Help_Files_Folder

PanelView Plus..... \My Documents\Help_Files_Folder

PC Runtime..... C:\Users\Public\Documents\RSView Enterprise\Help_Files_Folder

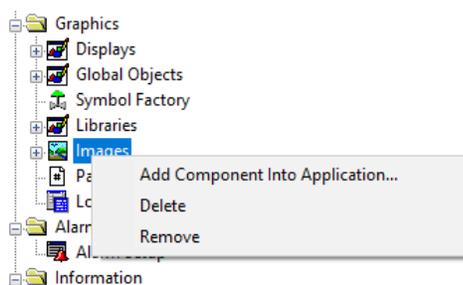
Help File Setup for Site Edition

For FactoryTalk View SE, Run the setup.cmd file located in the Hardy Device Library download (HardyDeviceLibrary_v1.xx) or place the pdf in the location C:\Users\Public\Documents\RSView Enterprise\Help_Files_Folder.

Import FactoryTalk® View Visualization Files

There are several components to import for the visualization files. You import files from the downloaded Rockwell Automation® library files via FactoryTalk® View SE/ME. The workflow is the same for both FactoryTalk® View ME and SE.

All image and display items can be imported either by right-clicking in FactoryTalk® View on the Graphic sub-folder (e.g. Displays, Global Objects, Images) or simply dragging and dropping the files into the application.



Import files in this order:

1. Import HMI Images files.

Select all the images in the \HMI FactoryTalk® View Images - png folder and Open.

2. Import Global Object files

Select the global object (.ggfx) files from the \HMI - FactoryTalk® View ME\Global Objects - ggfx or \HMI - FactoryTalk® View SE\Global Objects - ggfx folder

3. Import HMI Faceplates

Select the faceplate (.gfx) files from the \HMI - FactoryTalk® View ME\Displays - gfx or \HMI - FactoryTalk® View SE\Displays - gfx folder

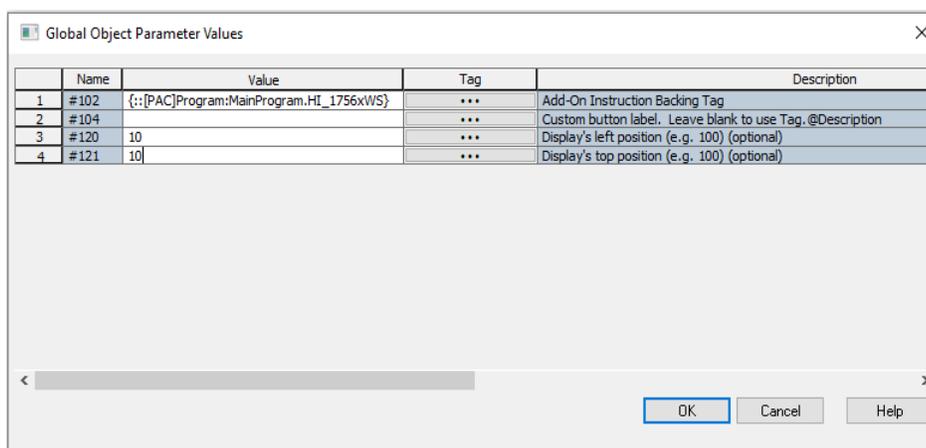
Configuring FactoryTalk® View Objects

Once the files have been imported into the FactoryTalk® View Studio project, you can begin using them in your application. Open the *Global Display (raC-X-ME) Graphic Symbols - Hardy Device* or *(raC-X-SE) Graphic Symbols - Hardy Device*. Copy the desired launch button style and paste it into a display in your application where you would like to open the faceplate. For more information

on graphic symbols, refer to the Graphic Symbols section of the specific device type chapter in this manual.



To configure the graphic symbol launch button, right-click and select *Global Object Parameter Values*. The Global Object Parameter values for the Backing Tag (#102) and Navigation Button Label (#104) are mandatory while the display position values (#120, #121) are optional. You can browse for the tag in your controller project by clicking ‘...’ or manually type them in. These parameters may vary depending on the graphic symbol used, please refer to the Graphic Symbols section of the device type for detailed information.



These Global Object Parameter Values are automatically configured when you use Studio 5000® Application Code Manager to design and configure your project. Refer to [Using Studio 5000® Application Code Manager](#) for more information.

Using FactoryTalk® Optix

Download the Library

For the latest compatible software information and to download the Rockwell Automation® Library, see the [Product Compatibility and Download Center](#).

Search “Hardy Device Library” or filter on Application Content to quickly find the library.



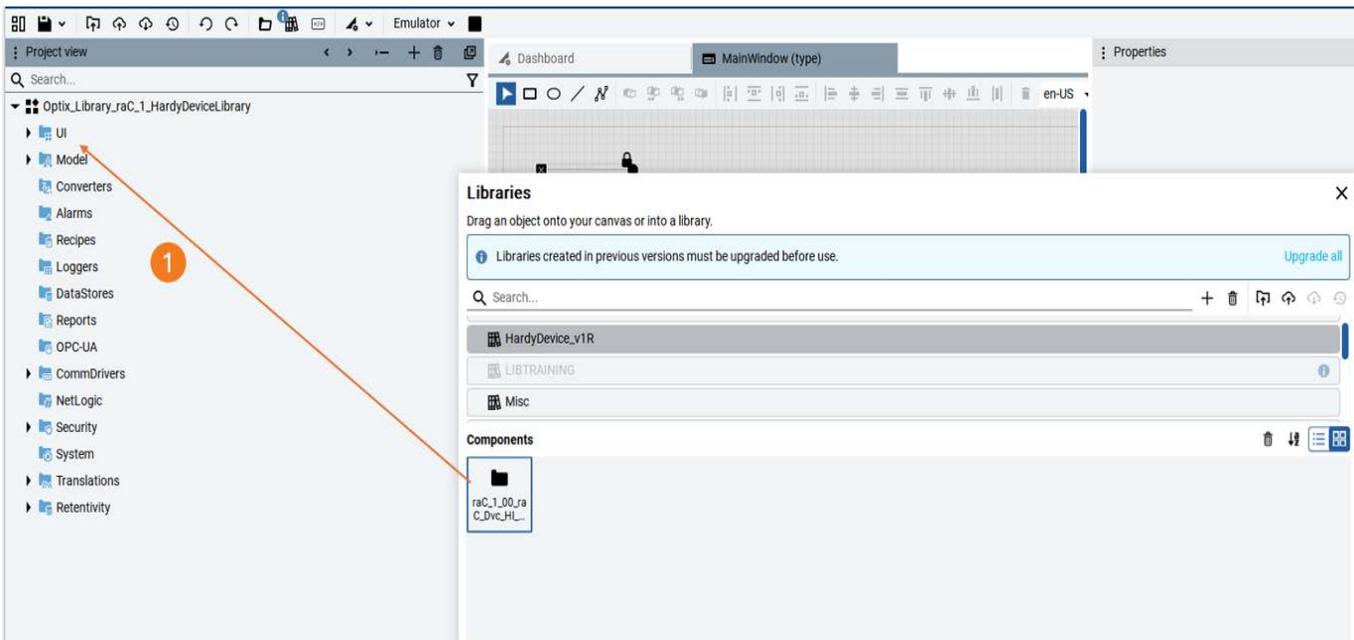
Currently, only HI 1756-WS objects are available in FactoryTalk Optix.

Register FactoryTalk Optix Library

1. Run the provided Setup.cmd script or extract the downloaded zip file to the Windows user's FactoryTalk Optix library directory, which by default is C:\Users\- 2. If you have FactoryTalk Optix Studio open on your computer, please close the application and reopen to ensure the library will be visible.

Import Library Objects in FactoryTalk Optix Project

1. Drag and drop the “raC_1_xx_raC_Dvc_ObjectName_UI” object to the UI folder of your project. *Shown as (1) below.*

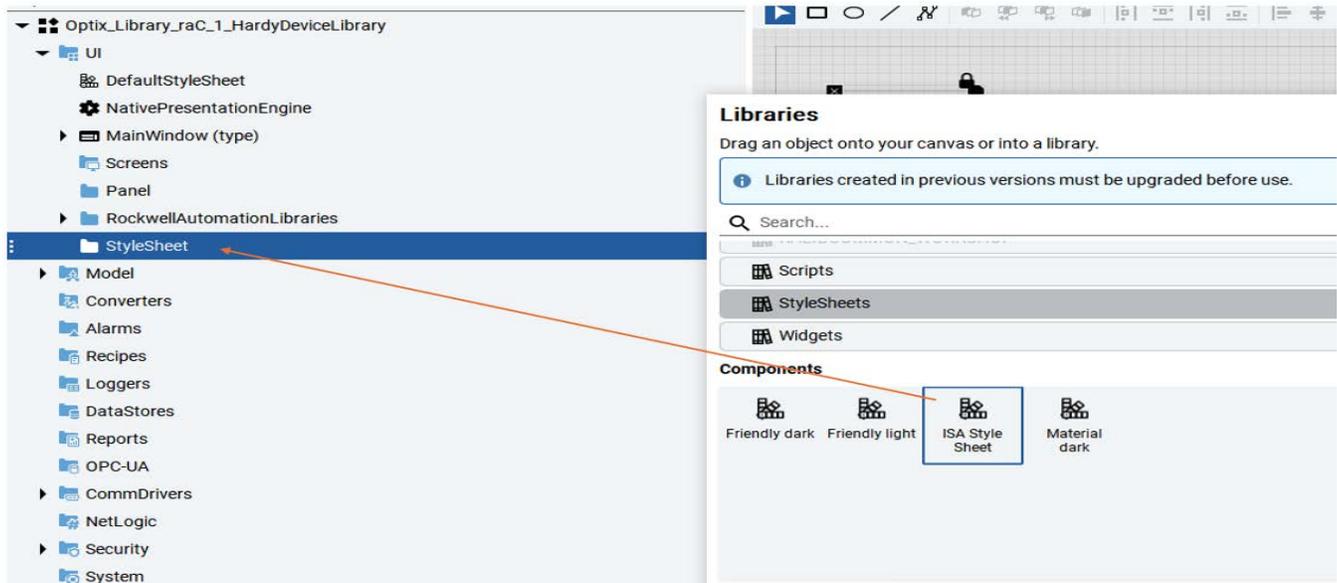


Using Style Sheets

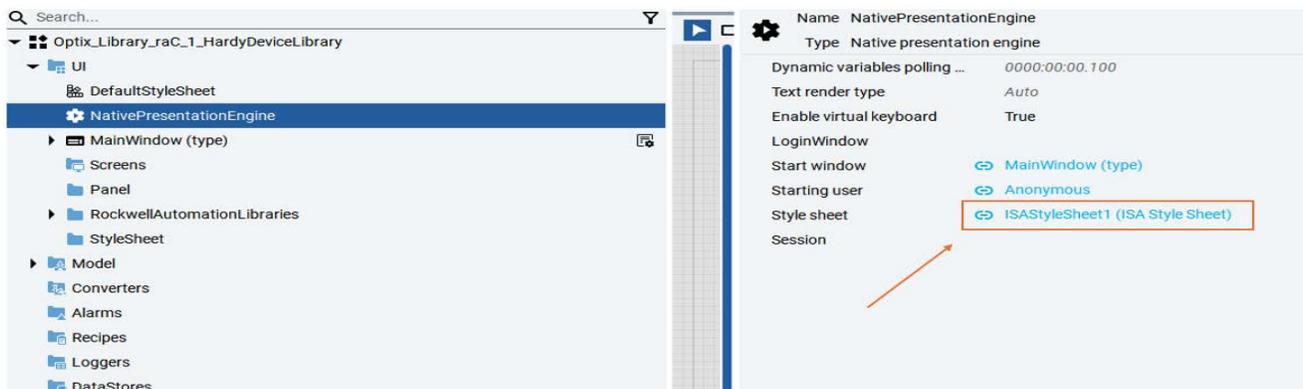
FactoryTalk Optix allows users to define custom style sheets or use ones provided in the FactoryTalk Optix Libraries. It is recommended to use the provided “ISA Style Sheet” for a consistent look and feel across all FactoryTalk Optix libraries and displays. All screenshots of faceplates in FactoryTalk Optix contained in this manual use the ISA Style Sheet. Other style sheets may be applied; however, this will cause faceplates to appear differently and usability can be hindered in some cases for example where low-contrast colors or different sizes are chosen.

Use the following steps to import and set the ISA Style Sheet which comes with FactoryTalk Optix Studio in the StyleSheets library.

1. In FactoryTalk Optix Studio, drag and drop the “ISA Style Sheet” component from StyleSheets library to a location in your project.



2. In the PresentationEngine used in the Optix project (e.g. NativePresentationEngine), point the Style sheet property to the ISAStyleSheet just imported. Update the Style Sheet's Font size and other settings accordingly.

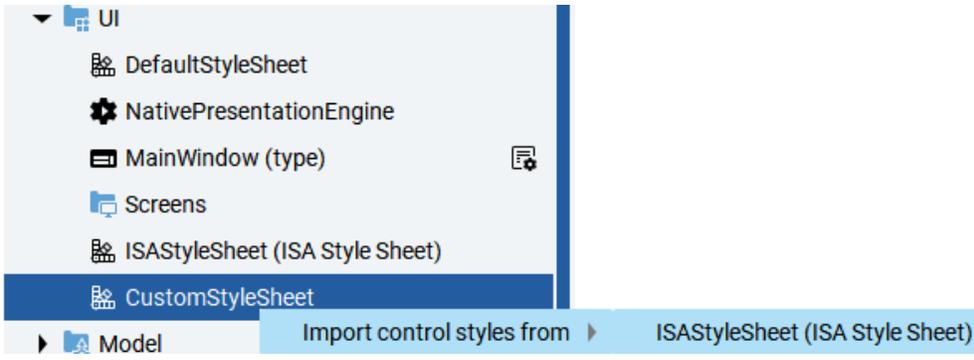


Custom Style sheet

You can now merge new style types from one stylesheet to another. Here's how to use this feature with the ISA Style Sheet:

- a. If you prefer a custom style sheet over the ISA Style Sheet, start by importing the ISA Style Sheet from the template library.
- b. Right-click on your custom style sheet and select Import control styles from > ISA Style Sheet.

Any styles in the ISA Style Sheet that are not in your custom style sheet will be added. You can then modify these new styles to fit your needs.

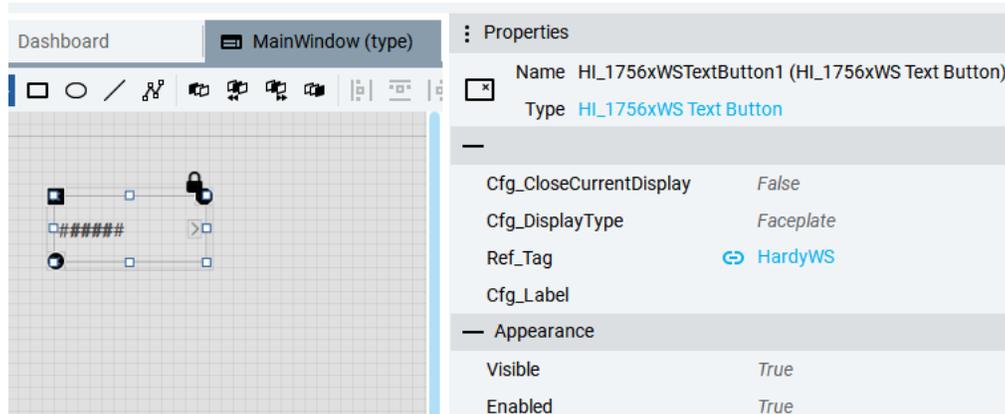


Configuring FactoryTalk Optix Objects

Once the Objects have been imported into the FactoryTalk® Optix Studio project, you can begin using them in your application. To add a new Launch Button to a Main window, navigate to raC_1_xx_raC_Dvc_ObjectName_UI > Graphic Symbols > raC_1_xx_raC_Dvc_ObjectName_GS_NavText Button to insert a navigation launch button with a text label.

After placing the graphic symbol on a UI panel, link the “Ref_Tag” property to the targeted Asset under Asset tag.

Text label shown on button can be configured using “cfg_Label” property, If it is not configured then description of the asset will be shown on the button face.



This is the only step needed to link the UI to the asset data model. For more information on graphic symbols, refer to the Graphic Symbols section of the respective device type in this manual.

Library Upgrades

Add-On Instruction Upgrades

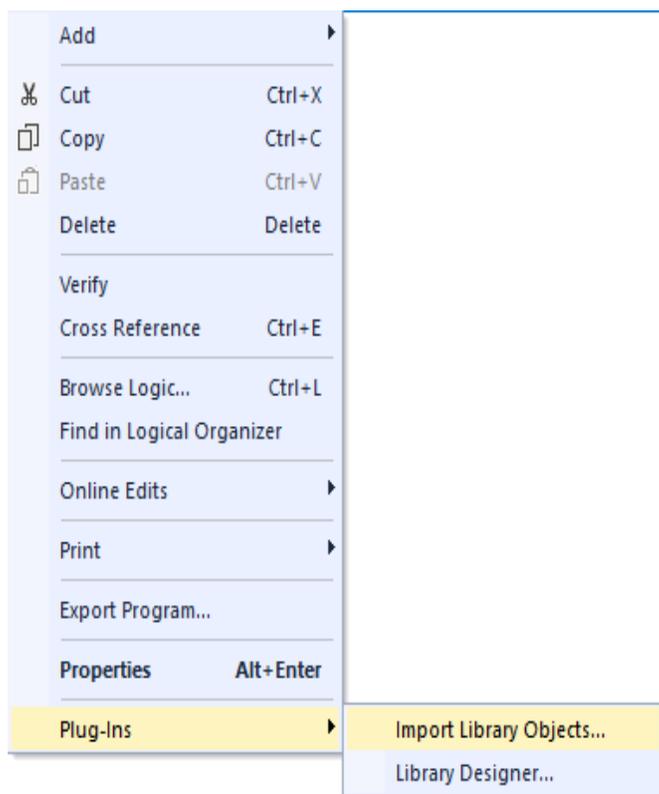
There are two methods to upgrading existing device object add-on instructions in a project. You can do this either by using the Studio 5000® Plug-In *Import Library Objects Wizard* or by importing individual add-on instruction AOI.L5X files. Using the Studio 5000® Plug-In *Import Library Objects Wizard* is the preferred method to reduce the risk of errors or compatibility issues. Both methods are described in the following sections.

Note that all updates to Add-On Instructions must be done with Studio 5000 Logix Designer® in OFFLINE mode and a download to the controller is required.

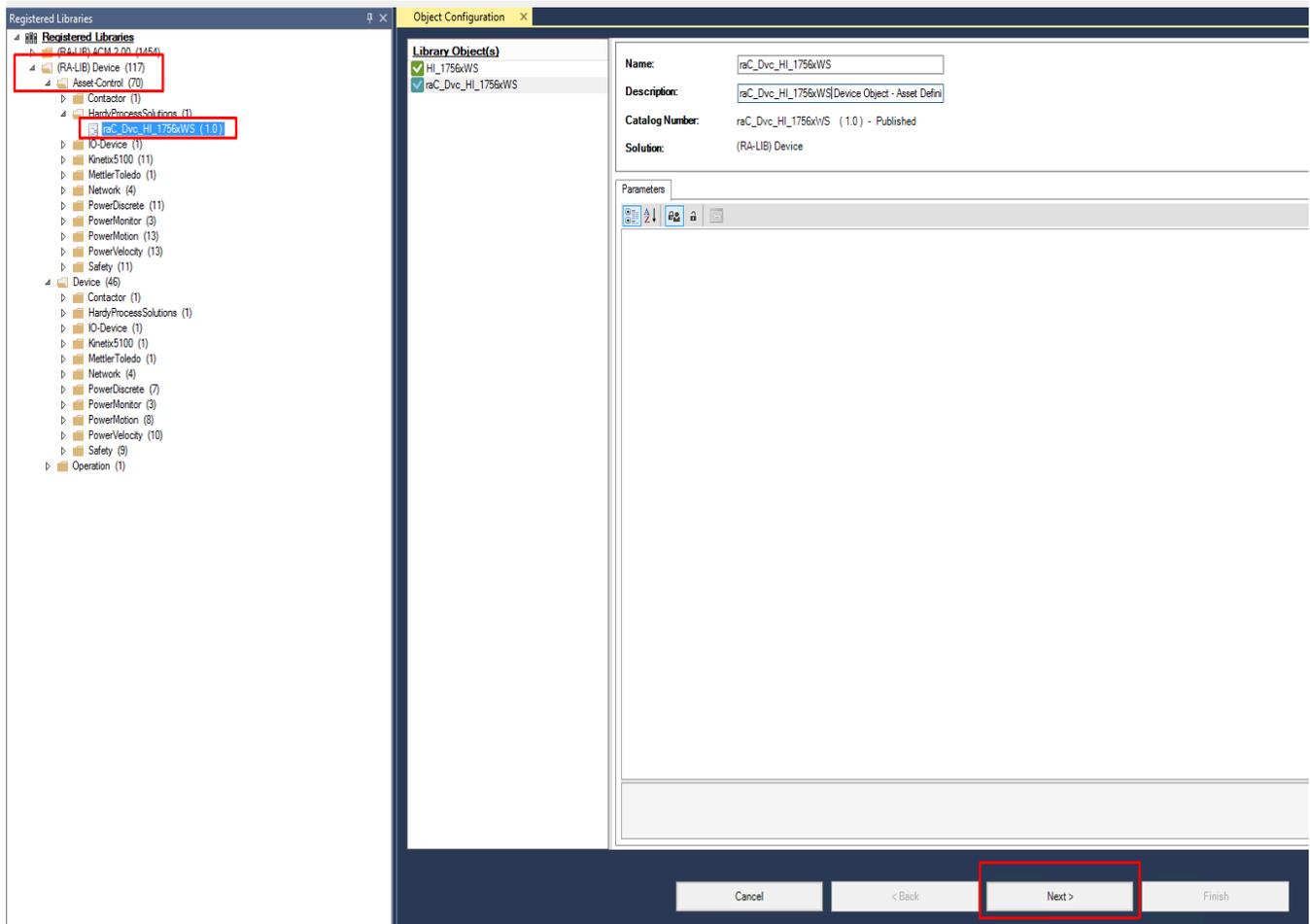
Upgrades Using Studio 5000® Plug-In to Import Library Objects

If Studio 5000® Application Code Manager is installed, you can use the Studio 5000® Plug-In *Import Library Objects Wizard* to update existing Add-On Instructions. For complete information on Studio 5000® Application Code Manager, refer to the section [Using Application Code Manager](#).

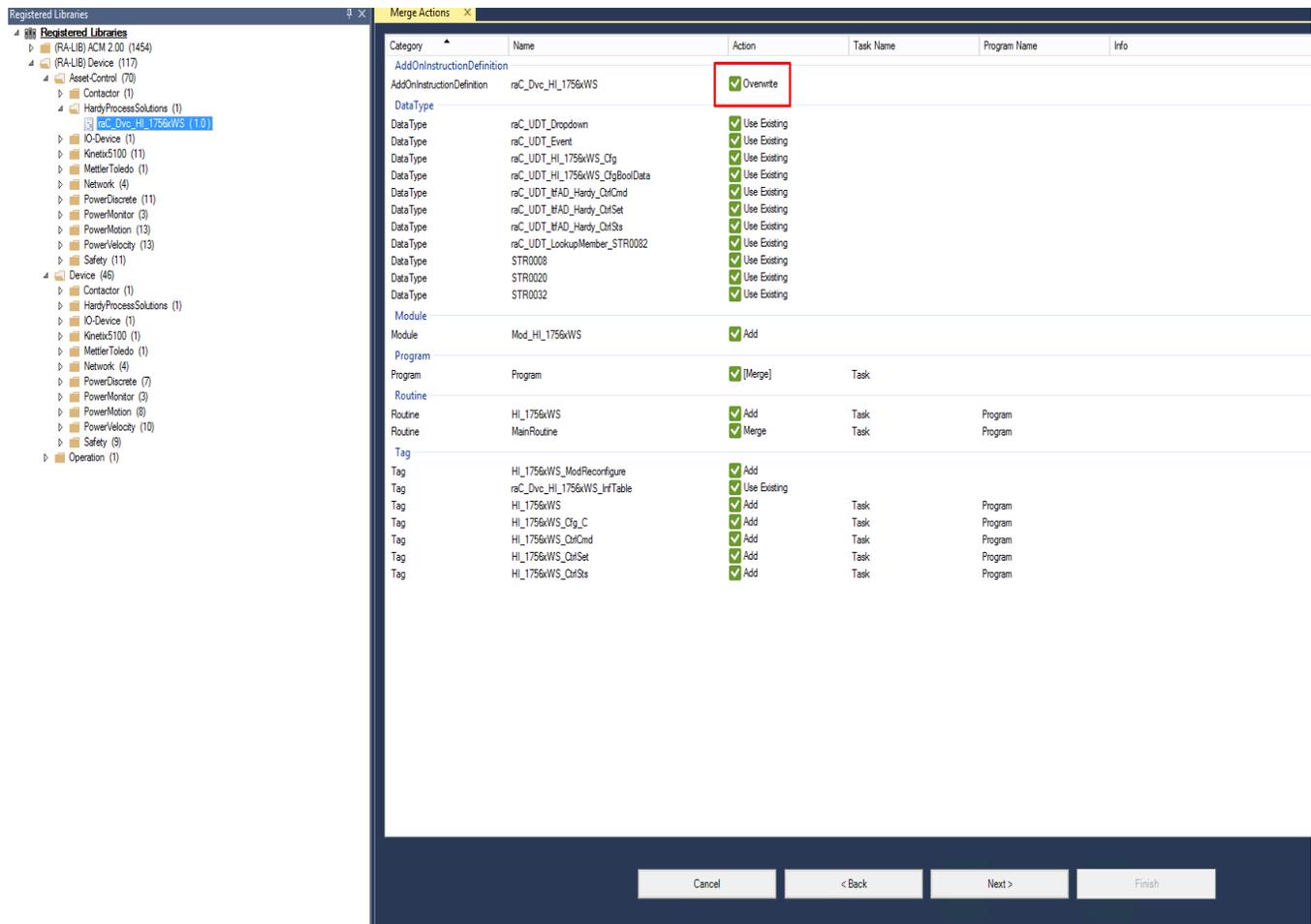
Right-click in your controller organizer or within a routine to access *Plug-Ins > Import Library Objects...*



The *Library Object Import Wizard* dialogue window will open. Under *Registered Libraries expand (RA-LIB) Device > Asset-Control* and find the desired object and version. Drag the object into the *Object Configuration* window on the right.



In the *Merge Actions* window, select the *Action* for the *AddOnIntructionDefinition* to *Overwrite*. This will update any existing instance of the object to the newer version. You may also choose to overwrite any other *DataTypes* or *Tags*. Review the release notes of the latest library release to understand what may be impacted. Click next and finish to complete the process.



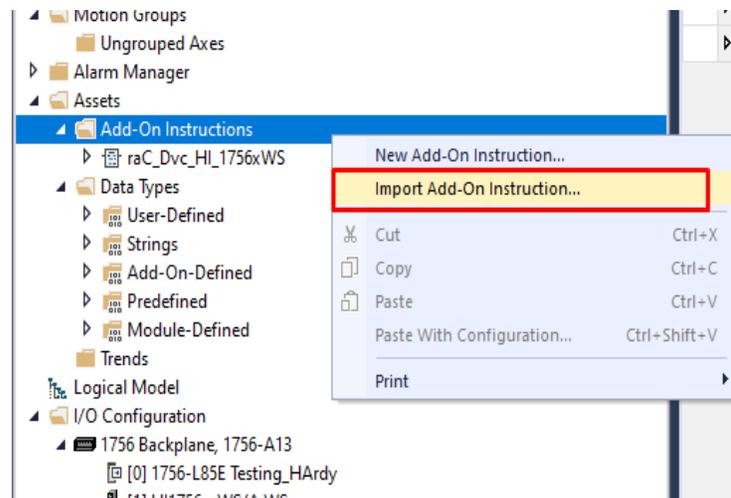
Upgrades by Importing AOI.L5X Files

To upgrade or migrate a project that uses a previous library version to a newer one, the add-on instruction L5X files are supplied.

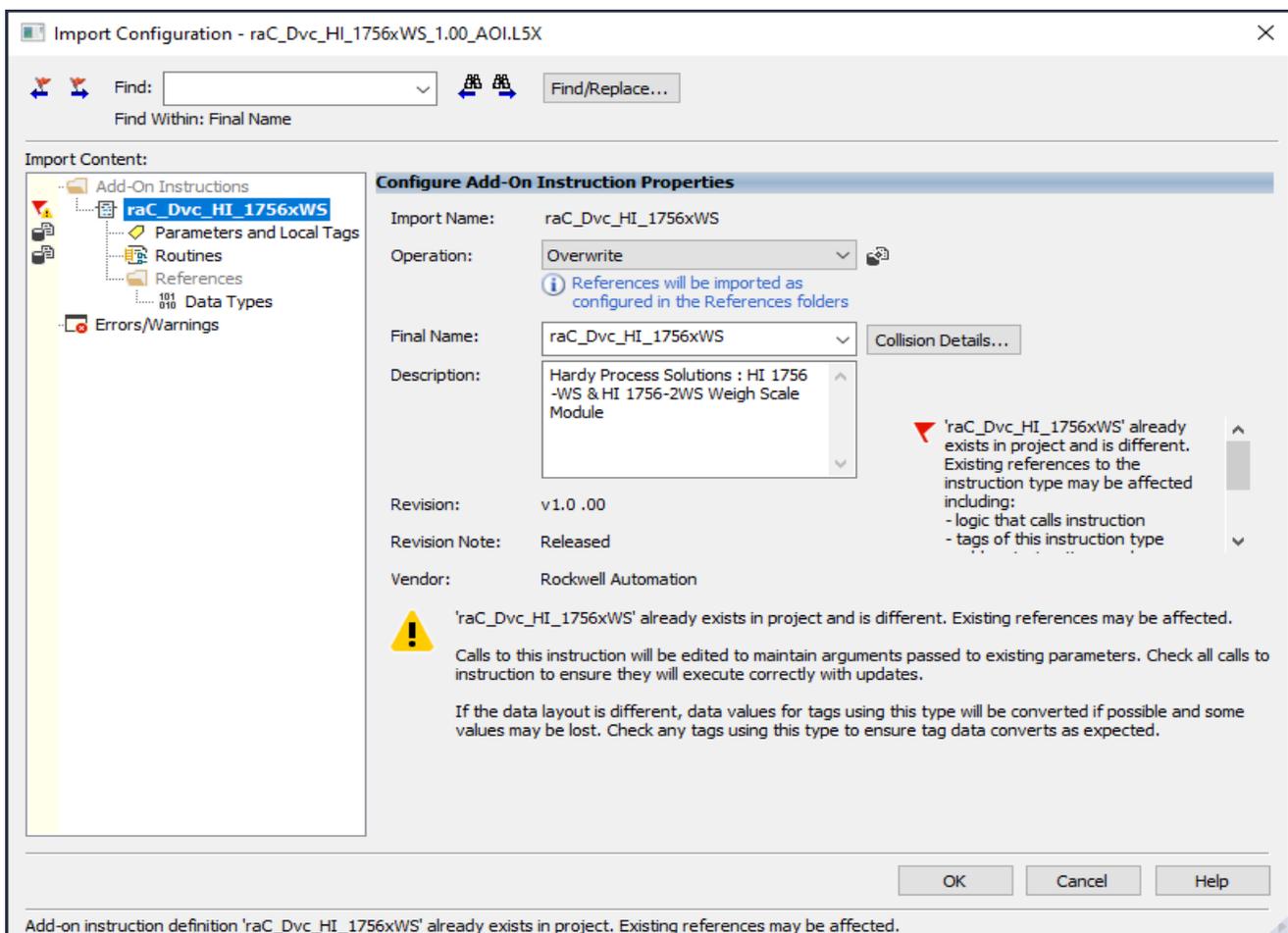
IMPORTANT In newly released versions of this library, it is possible that modifications or improvements have been made to items such as tags, faults tables, etc that are outside of the Add-On Instruction. To ensure all items are updated use the Studio 5000® Import Library Objects plug-in method or import RUNG.L5X files. Read the library release notes to understand what has been updated in the latest version of the library.

To perform an upgrade to an object perform the following steps:

- Open the controller file. Note changes must be done offline.
- In the *Controller Organizer* pane right-click on *Assets > Add-On Instructions* and select *Import Add-On Instruction*. Navigate to the AOI.L5X file in the *Studio 5000 Logix Designer Files - L5X* and Open.

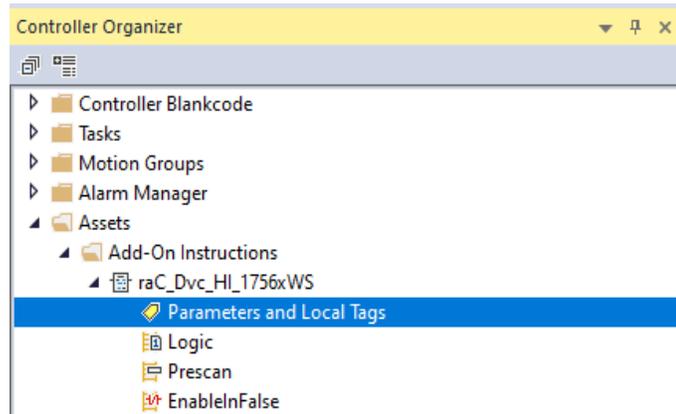


- You will be prompted that there is an existing version of the instruction that is different. Choose *Override* as the operation and select OK once you have read and understood the warnings. Your existing logic will be updated with the new add-on instruction. Verify that your code compiles and test adequately.

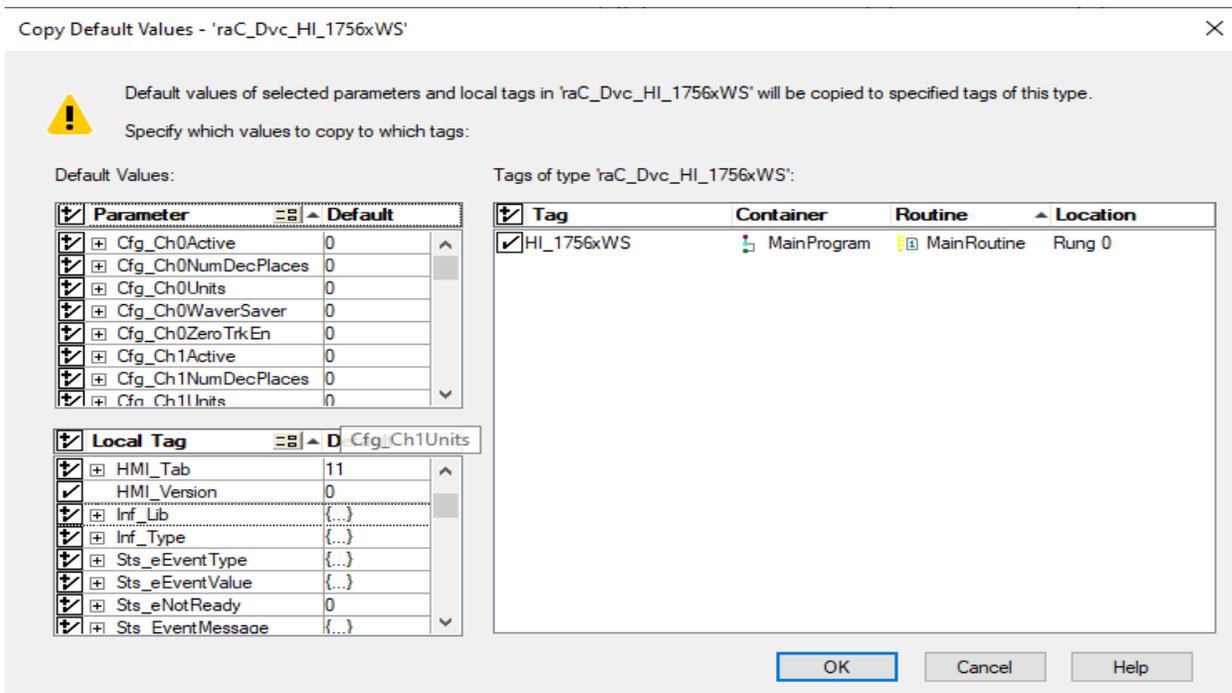
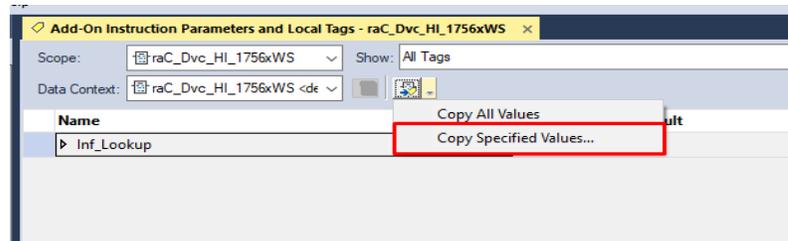


- In order to ensure the HMI faceplate still works properly you will need to update the object's library information stored in the Inf_Lib tag. In

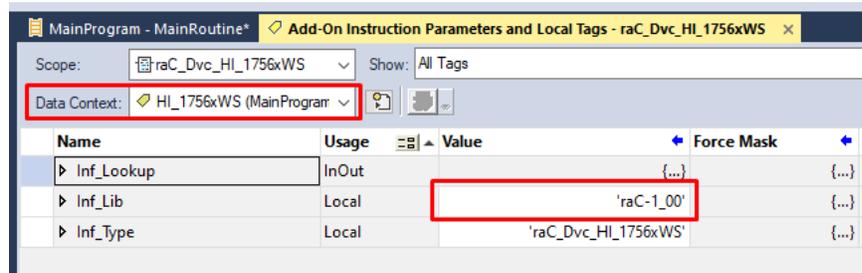
the *Controller Organizer* pane under *Assets > Add-On Instructions* expand the device object that was updated. Double-click on *Parameters and Local Tags* to open up the instructions tags.



- In the *Add-On Instruction Parameters and Local Tags* window, you may notice that the *Inf_Lib* tag in the add-on instruction definition matches the new library revision number. Click on the down-arrow to the right of the copy button and select *Copy Specified Values...*



- In the *Copy Default Values* window, be sure to **first uncheck all Parameters and Local Tags** by clicking the +/- box in the top right. Failure to do so may result in overwriting settings in the existing objects.
- Check only *Inf_Lib* in the *Local Tag* area. On the right, all affected objects should be selected. Click OK.
- You can now confirm that the *Inf_Lib* tag has been updated to the current library (e.g. 'raC-1_00') by changing the *Data Context* dropdown to a specific device object.



FactoryTalk® View Upgrades

To upgrade a device object in a FactoryTalk® View ME/SE application, simply import the new faceplate .gfx display file into the application. If any global objects or images have been added or modified, you may need to import these as well. Any unused displays from previous versions may be removed or deleted from the application.

Note that the reference to the faceplate version is set in the Add-On Instruction Local Tag *Inf_Lib* so there does not need to be other modifications to the HMI application.

Studio 5000 View Designer® Upgrades

To upgrade a device object in a Studio 5000 View Designer® application, simply import the open the new View Designer .vpd file and copy the raC_Dvc_XXXX_FP pop-up screen into the existing application. Find any graphic symbol launch buttons in the application that open the faceplate, and update the Action to open the new pop-up screen. Any unused pop-up screens from previous versions may be removed or deleted from the application.

Using Application Code Manager

Overview of Application Code Manager

Studio 5000® Application Code Manager is a tool that enables more efficient project development with libraries of reusable code. Application Code Manager creates modular objects with customizable configuration parameters using the reusable content. Application Code Manager can also create the associated visualization, historical and alarming elements for a project.

Studio 5000® Application Code Manager can be easily used along with Rockwell Automation® application code libraries such as the PlantPAX® Process Objects Library, Machine Builder Library, and Device Object Libraries. For more information on Studio 5000® Application Code Manager, refer to the [Application Code Manager User Manual](#).

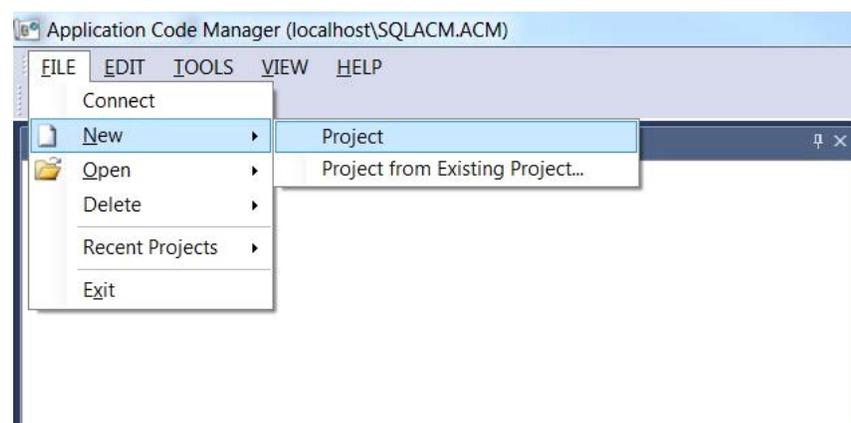
Creating a New Project

Begin by opening Application Code Manager.



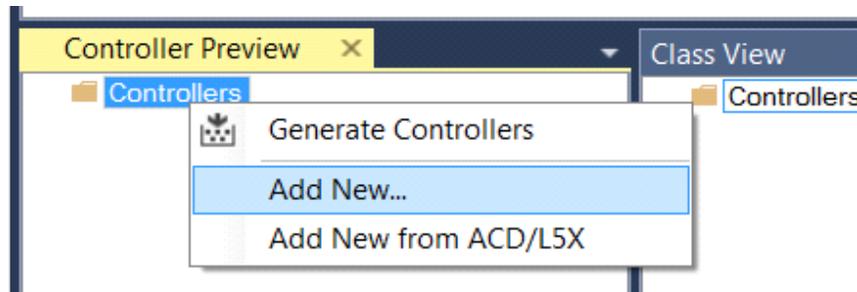
Note: the last project (if any) is opened by default; otherwise a blank screen is displayed.

Create a New Project or open an existing project. Navigate to *File > New > Project*.



Select the desired project type (e.g. *(RA-LIB) ACM 2.00 Project - Basic_Project*) and fill in the *Name* and *Description*.

To add a new controller to a project, in the *Controller Preview* window, right-click on *Controllers* and select *Add New...*



Select the desired controller type (e.g. *ControlLogix_Controller*, *GuardLogix_Controller*, *CompactLogix_Controller*, etc). Enter a *Name* and *Description* for the controller. Select the appropriate Chassis and Processor configurations.

You can also configure the HMI *AreaPath* and *AreaPathME* parameters which will be referenced if you use Application Code Manager to generate FactoryTalk® View SE/ME displays with graphic symbol launch buttons.

Object Configuration Wizard

Name: Hardy_Device_V1_00

Description: ControlLogix Controller

Catalog Number: ControlLogix_Controller (2.1) - Published

Solution: (RA-LIB) ACM 2.00

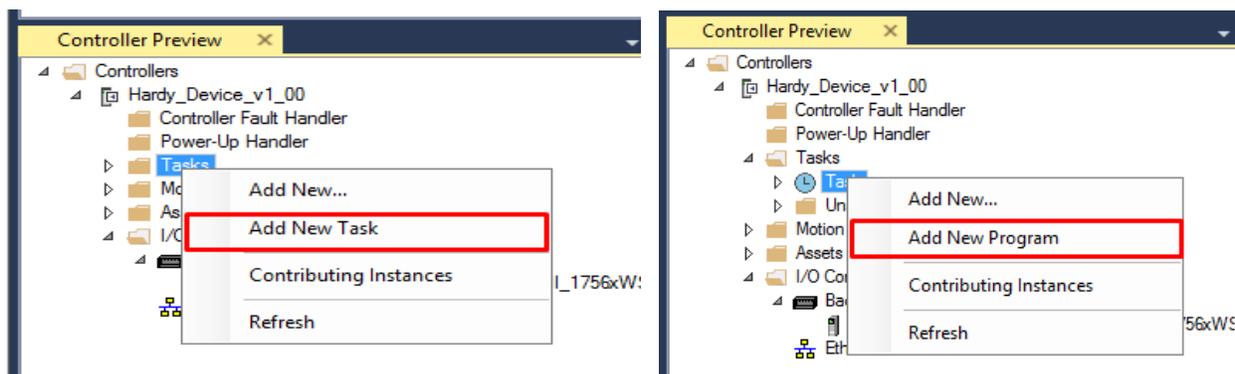
Parameters

01 - Controller	
ChassisName	Local
Slot	0
Size	17
SoftwareRevision	33
ProcessorType	1756-L85E
02 - HMI	
AreaPath	[PAC_SE]
AreaPathME	[PAC_ME]
03 - Historian	
HistorianPath	Application/Area.RSLinx Enterprise:[shortcut]
FTLDInterfaceNo	1
Motion	
ConfigureMotion	False
Port Configuration	
EthernetPort1_Enabled	True
Time Synchronization	
TimeSync_Priority1	128
TimeSync_Priority2	128
TimeSync_PTPEnable	True

EthernetPort1_Enabled
Enable\Disable the embedded Ethernet Port 1 of the controller

Cancel << Previous Next >> Finish

You can now add in any desired tasks and programs to your controller. Right-Click on the *Tasks* folder underneath your controller in the *Controller Preview* and *Add New Task*. Similarly, right-click on any Task and select *Add New Program*. Complete the desired parameters for Tasks and Programs such as name, type, period, etc.

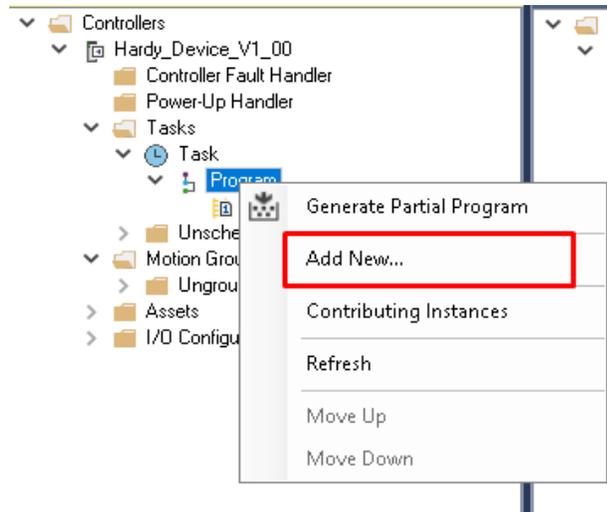


Adding & Configuring Device Objects

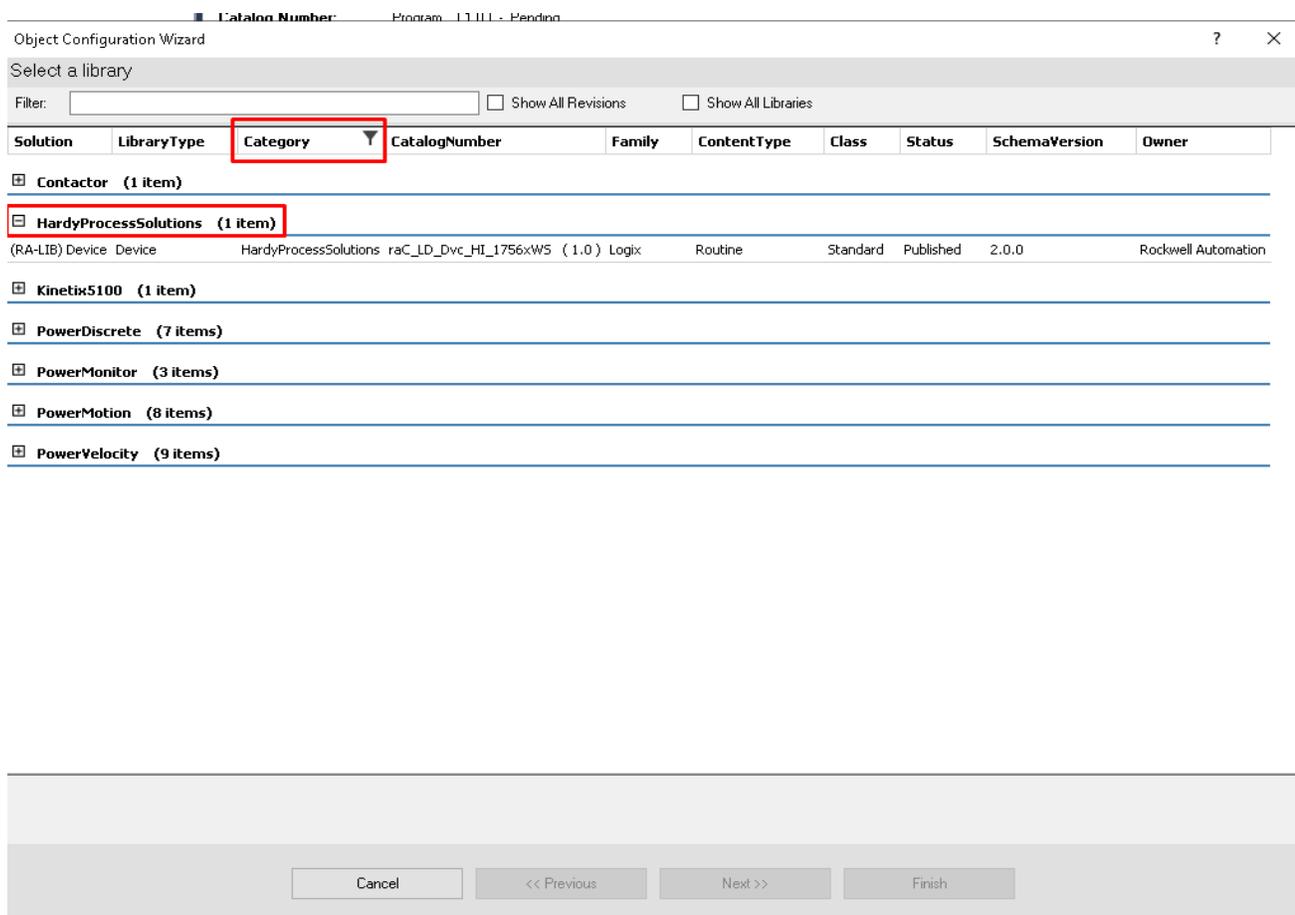
Prior to adding in any Device Objects, ensure you have registered the library in Application Code Manager. Refer to [Registering Libraries in Studio 5000® Application Code Manager](#) for details.

Adding Hardy Device Objects

To add a Hardy Device Object into a project, right-click on a Program and *Add New...*



Select the Device Object that you wish to import. You can click on the *Category* heading to group items by category and easily find *Hardy* devices.



Fill in all of the required configuration parameters for the device object. The following example shows a configuration of the raC_LD_Dvc_HI_1756xWS object.

Perform the following configuration:

o0 General:

- Enter a **name** and **description**. Maximum name length can be 22 characters. Note that other parameters such as the RoutineName, TagName, etc will auto-complete based on these fields.
- Assign the **Task** and **Program**. Note these are pre-assigned if you added the object to a specific program by right-clicking in the *Controller Preview* pane.

o1 Module:

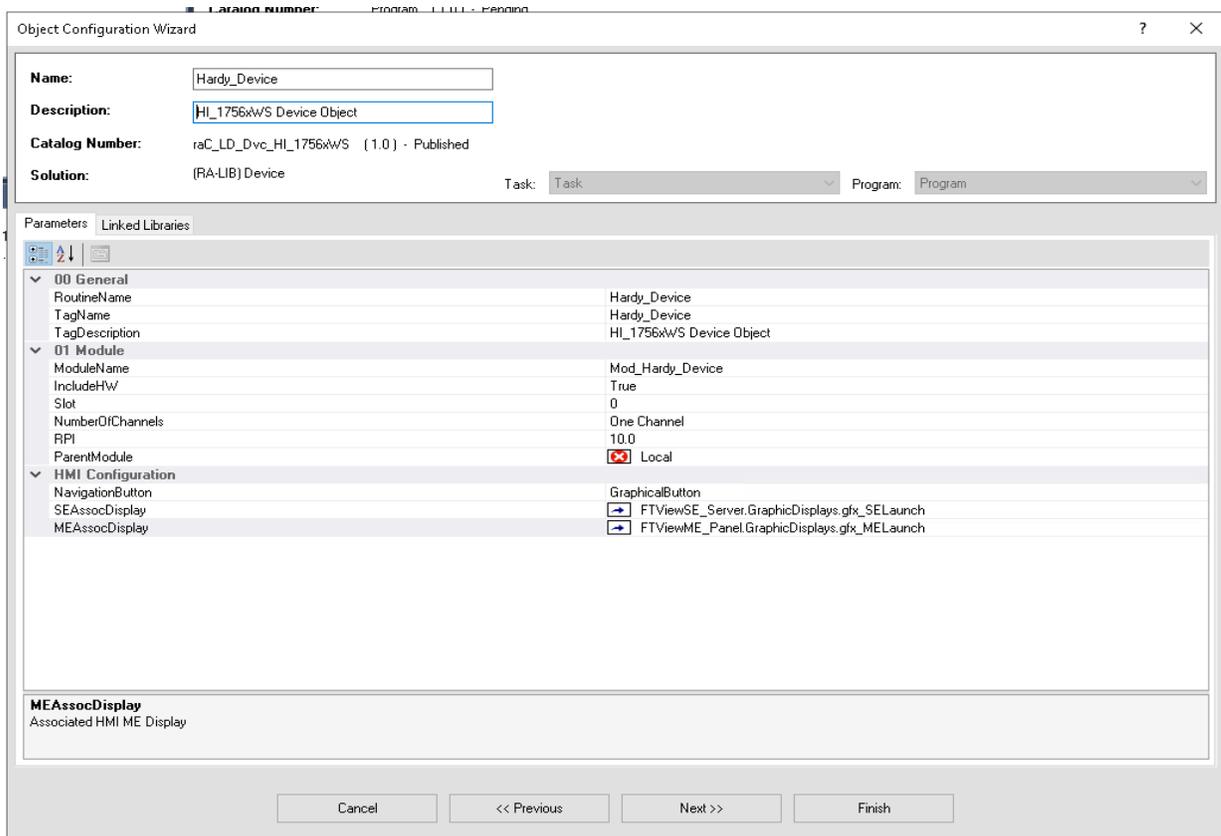
- The **ModuleName** will default to *Mod_ObjectName*
- Set **IncludeHW** to True to allow Application Code Manager to create the Hardy Device Module(1756xWS).
- Assign **Slot Number** along with the **ParentModule**. Note that *ParentModule* may show a red 'x' when using "Local" is not created in the Application Code Manager project. "Local" is used for embedded ethernet ports such as in the L8xE controller family.

- Select the **NumberOfChannel** and **RPI** Parameters. The *NumberOfChannel Format* will default to One Channel, and *RPI* will default to 10.0ms.

HMI Configuration:

- Set *SEAssocDisplay* to a FactoryTalk® View SE display configured in the project if you plan on generating displays using Studio 5000® Application Code Manager.
- Set the desired *SymbolStyle* to either “GraphicalButton” to use the graphical style launch button for schematic/system displays; or “GenericTextButton” to use a simple rectangular text button. For more information on HMI Configuration refer to [Configuring Displays](#).

Click on the *Linked Libraries* tab. Click the *Auto Create* button to automatically create all of the required linked libraries.

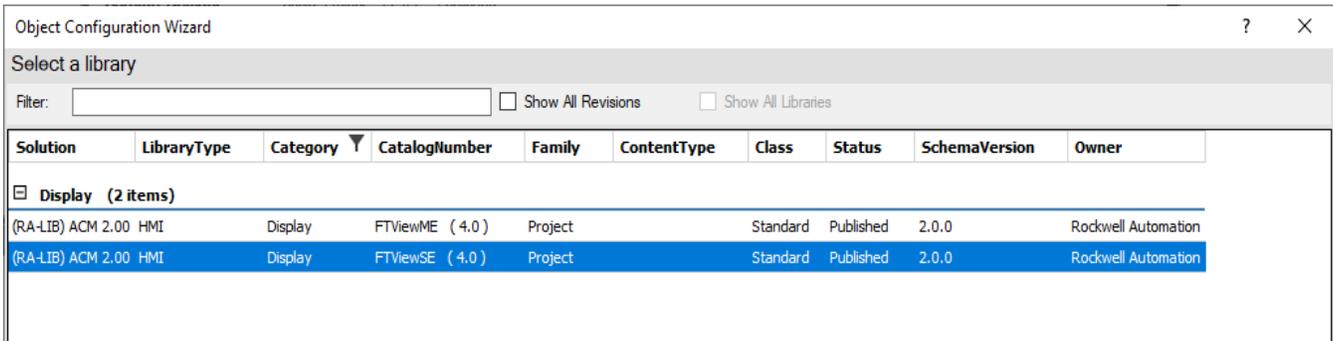
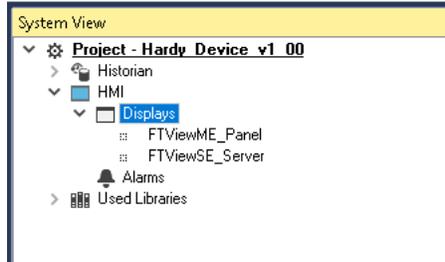


Click Finish to complete the import. For specific devices details, refer to the appropriate chapter in this manual.

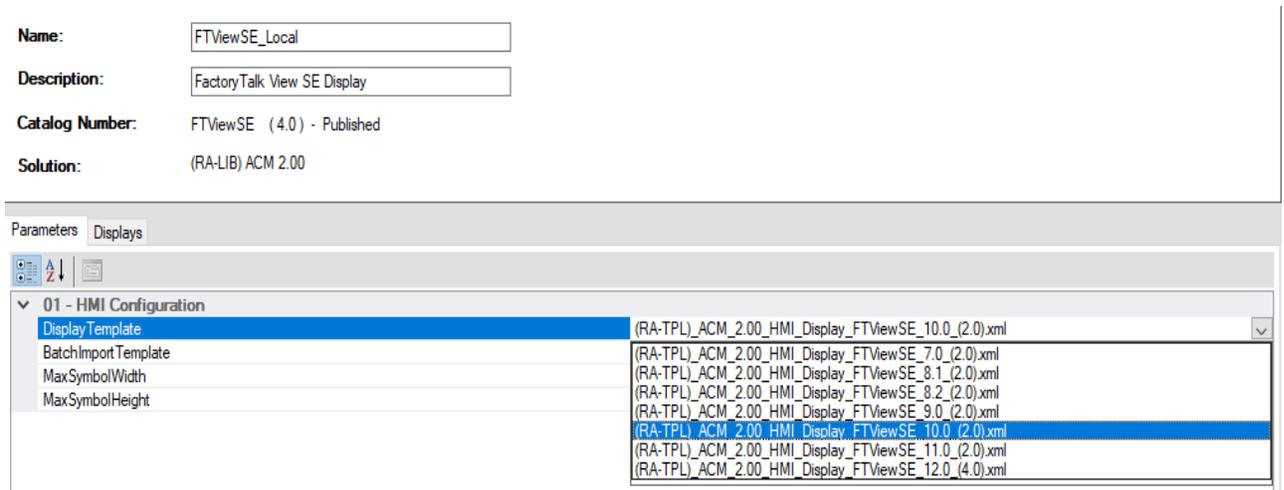
Configuring Displays

Application Code Manager can be used to automatically configure graphic symbol launch buttons for device objects In FactoryTalk® View ME or SE. Note that Application Code Manager is not compatible with Studio 5000 View Designer® applications.

First you must add Displays to your project. Under the *System View* panel expand *HMI* and right-click on *Displays* to select *Add*. Choose the type of display (e.g. *FTViewME* or *FTViewSE* depending on your project requirements).



In the display object parameter configuration, you must select the *DisplayTemplate* type to match the version of FactoryTalk® View application that you are using.



Navigate to the *Displays* tab where you can right-click and *Add New* display.

Name: FTViewSE_Local
Description:
Catalog Number: FTViewSE (4.0) - Published
Solution: (RA-LIB) ACM 2.00

Parameters		Displays
Name	SubObject Description	
	<ul style="list-style-type: none"> Add New Copy Paste Delete Reset Grouping... 	

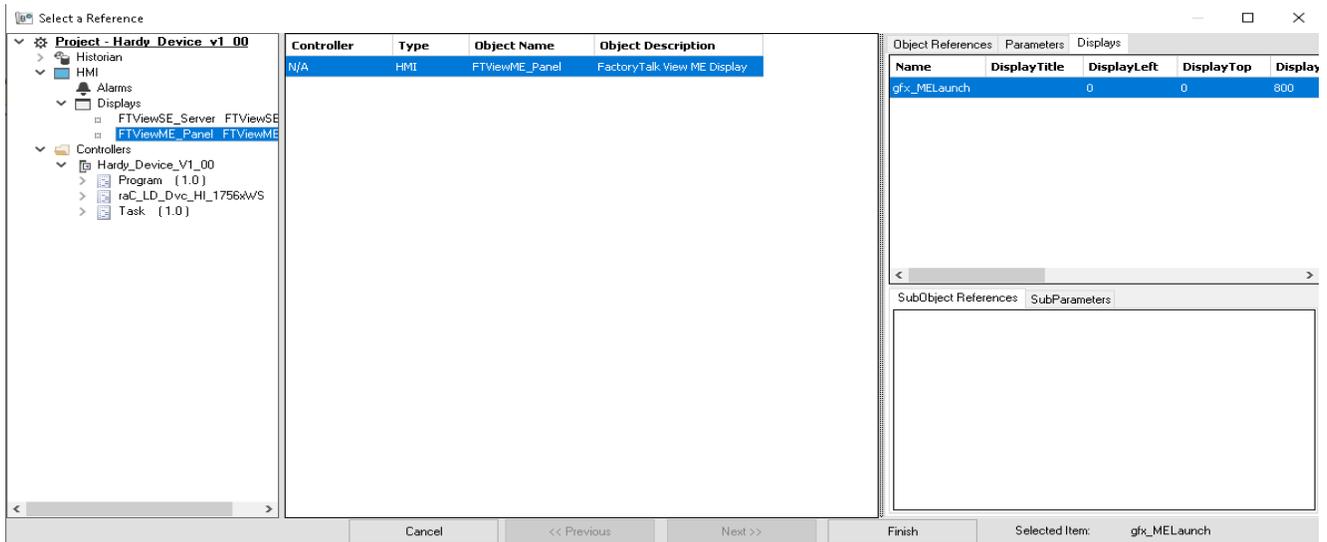
Set the desired name and display parameters. Generally all display parameters aside from *Name* can be left as default since this will often be used as a temporary display where object launch buttons are copied from.

Name: FTViewSE_Server
Description:
Catalog Number: FTViewSE (4.0) - Published
Solution: (RA-LIB) ACM 2.00

Parameters		Displays		
Name	DisplayTitle	DisplayLeft	DisplayTop	DisplayV
gfx_SELaunch		0	0	3000

Return to your device object configuration and view the *HMI Configuration* section of the parameters. You can browse or type in the *HMI_Server_Name.HMI_Display_Name*.

HMI Configuration	
NavigationButton	GraphicalButton
SEAssocDisplay	<input type="text" value="FTViewSE_Server.GraphicDisplays.gfx_SELaunch"/>
MEAssocDisplay	<input type="text" value="FTViewME_Panel.GraphicDisplays.gfx_MELaunch"/>

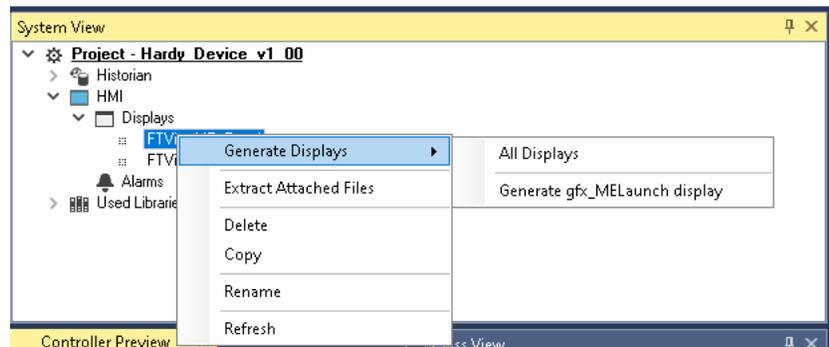


If you browse for the display, select the desired display server in the left panel, then click on the *Display* tab in the right panel and select the specific display. Click finish.

This workflow can be followed for either FactoryTalk® View ME or SE depending on the project requirements.

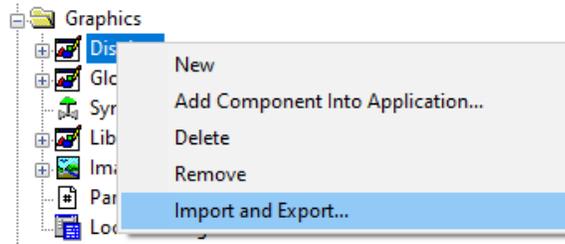
Generating Displays

Once you have assigned displays to all of the device objects, you can generate the displays. In the *System View* highlight the desired display server under *HMI > Display* and right-click to select *Generate Displays > All Displays* or select individual displays. Choose a place to save the generated files and take note of it.



Importing Displays into FactoryTalk® View Studio

To import the configured displays, open your FactoryTalk® View ME/SE project in FactoryTalk® View Studio. Right-click on *Graphic > Displays* and select *Import and Export...*



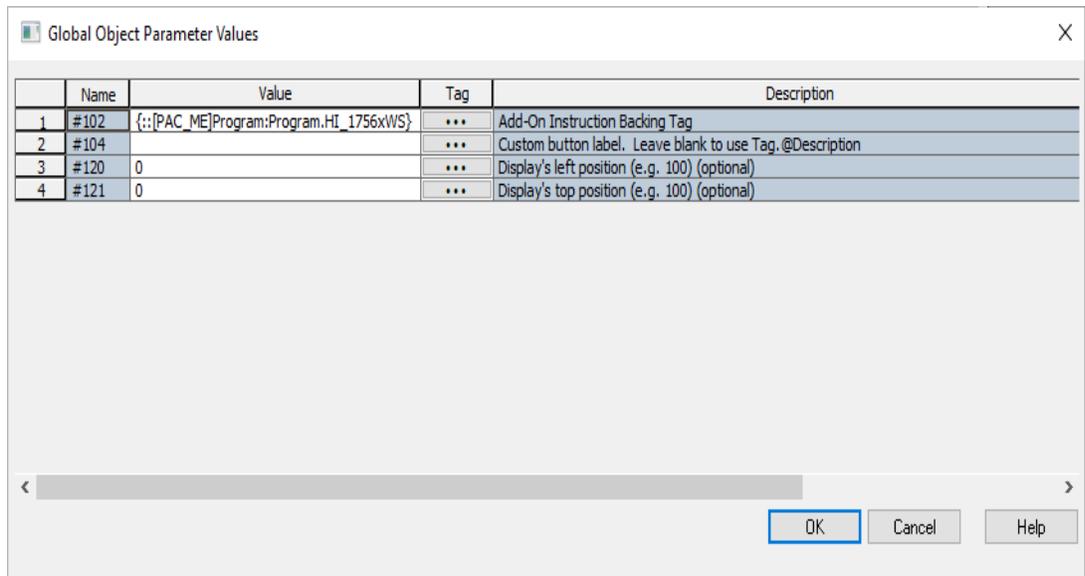
Follow the required prompts:

- Import graphic information into displays
- Choose whether or not to backup displays
- Choose either a *Single display import file* (must have an existing or blank display to import into) or *Multiple displays batch import file* if *All Displays* was used to Generate Displays.
- If this is the first time it is recommended to import *Multiple displays batch import file* and then *Create new objects on the display*.
- If you have done this before and are updating the imported display after modifying your Application Code Manager project, you can choose *Update existing objects on the display*.
- Browse for the BatchImport.xml file or individual display.xml file.

Open up the newly imported display. Notice that there are graphic symbol launch buttons labeled and configured for each item that was configured in Application Code Manager.



Right-click on the object and select *Global Object Parameters* to view that all of the parameters have been pre-configured for you.

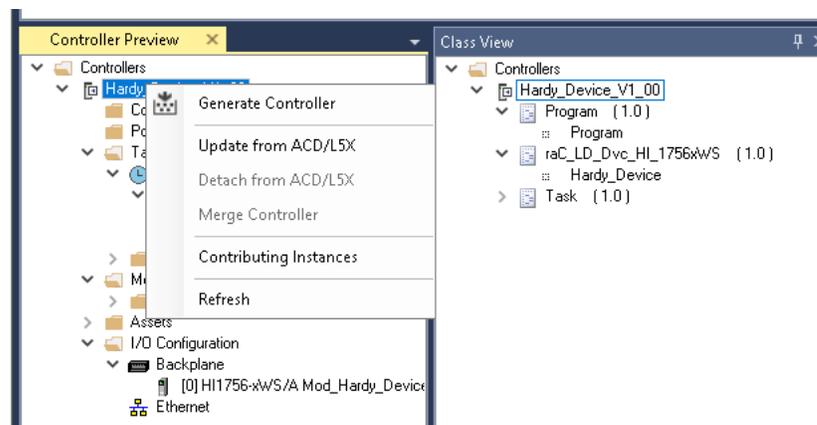


You may not copy and paste this graphic symbol onto any other display in your application.

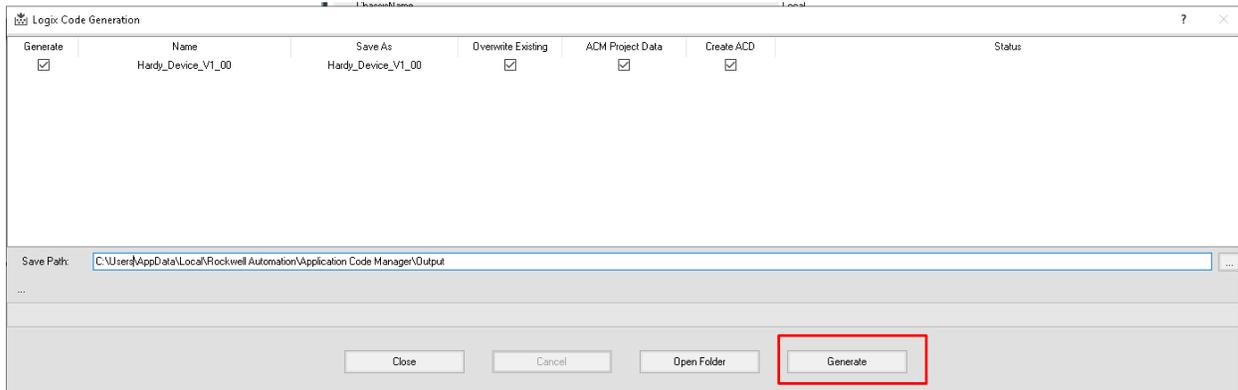
Generating Controller Files

Once you have completed configuring your project in Studio 5000® Application Code Manager, you can generate the controller file for use in Studio 5000 Logix Designer®.

In the *Controller Preview* pane right-click on the controller name within the *Controllers* folder and select *Generate Controller*.



In the *Logix Code Generation* dialogue window you will need to check *Create ACD*. You may also need to check *Overwrite Existing* if this is not the first time generating the controller code.



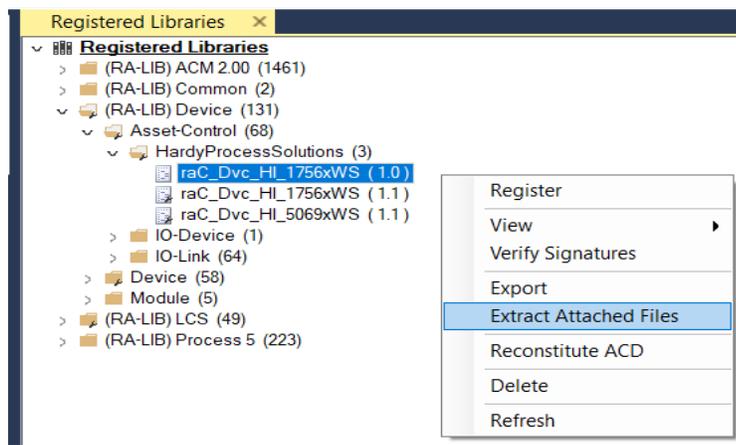
Once the controller file is generated, you can navigate to the location set in *Save Path* and open your file. Note that all of the configuration that was done in Application Code Manager is now shown in your Logix Designer ACD file.

Exporting Attachments

Application Code Libraries not only contain Logix code, but also contain Visualization collateral and associated documentation. Every Asset library contains at least a reference manual (RM). Those libraries which have associated Visualization content also have all required global objects (GO), images, static displays and View Designer applications added as attachments. In this manner the user can generate only the necessary visualization and documentation for the objects included in the project.

In Application Code Manager, all of the attachments are associated with the device objects in the *(RA-LIB) Device > Asset-Control* folder. These can be accessed both through the full Application Code Manager software, or via the Studio 5000 Plug-In “Import Library Objects”.

To access the attached files, right click on the objects (e.g. raC_Dvc_HI_1756xWS) and select *Extract Attached Files*.



Select the destination folder on your computer, and select OK. An Extract Attachments dialog will show the extraction status.

The extracted folder will contain the following:

- Reference Manual
- Required Images
- FactoryTalk® View Site Edition Display
- FactoryTalk® View Site Edition Global Objects
- FactoryTalk® View Machine Edition Display
- FactoryTalk® View Machine Edition Global Objects
- View Designer Faceplate File



Using the Hardy Device Library with Other Application Code Libraries

Application Code Libraries

The Hardy Device Library is commonly used alongside other Application Code Libraries.

Other libraries utilize the common device interface UDTs to interact with device level objects. This is covered in detail in [Interfaces](#) section of this document.

Using Hardy Device Objects with PlantPAX® Process Objects Library

The PlantPAX® Process Object Library application level library objects may be used in the same applications with the device-level objects in the Hardy Device Library. At this time there are no direct dependencies or interaction points between these libraries. They may be used independently from one another but within the same application.



PlantPAX® 5.xx only supports FactoryTalk® View SE or FactoryTalk Optix as an HMI platform. You should not be using the FactoryTalk® View ME or Studio 5000 View Designer® faceplates with a PlantPAX® application.

Using Hardy Device Objects with Machine Builder Library

The Machine Builder Library objects may be used in the same applications with the device-level objects in the Hardy Device Library. At this time there are no direct dependencies or interaction points between these libraries. They may be used independently from one another but within the same application.

HI1756 - ControlLogix® Weigh Scale Module (raC_Dvc_HI_1756xWS)

Overview

The Hardy device object includes a faceplate which displays status and configuration information of HI1756-ControlLogix® Weigh Scale Module series (raC_Dvc_HI_1756xWS). The HI1756-ControlLogix® Weigh Scale Module Series A is configured for single-channel operation while the HI 1756-2WS Weigh Scale Module Series A is configured for dual-channel operation. Both modules can be used for a wide variety of process weighing applications such as batching, blending, filling/dispensing, check weighing, force measurement, level by weight and weight rate monitoring. The analog-to-digital converter in the weigh module controller updates fifty times per second and is capable of 8,388,608 counts of display resolution. This is enough to provide accurate weight measurement and control and to tolerate large “dead” loads or oversizing of load cells/sensors. The weigh module analyzes the performance of each individual load cell and determines the total number of load cells in the system. C2 calibration can be initiated from the C2 calibration screen.

Functional Description

The HI1756-ControlLogix® Weigh Scale Module pre-configured Device Objects:

- Collect, Process and Deliver Data between Smart Devices and Application Logic
- Provide Device Status & Diagnostics Faceplates for Machine Startup, Commissioning, Operations, and Maintenance
- Include Common HMI Faceplate Components, Device States, and Control Interfaces providing Application Development and Operation Efficiency

All these features provide quick feedback, shorten recovery time, and simplify implementation.

Required Files

Device Objects include Add-On Instructions (AOIs) and HMI faceplates. The revision number (e.g. 1.00) used in file names can change as new revisions are created. If using FactoryTalk® View ME/SE you must also import the tag import file FTViewStudio_HardyLibrary_Tags_1_01.CSV to open the Help file.

Controller Files

Add-On Instructions are reusable code objects that contain encapsulated logic that can streamline implementing your system. This lets you create your own instruction set for programming logic as a supplement to the instruction set

provided natively in the ControlLogix® firmware. An Add-On Instruction is defined once in each controller project, and can be instantiated multiple times in your application code as needed.

The Add-On Instruction must be imported into the controller project to be used in the controller configuration. These can be imported as Add-On Instruction files, or as part of the Rung Import or Import Library Objects wizard.

All Add-On Instruction and Rung Import files can be found in the */Studio 5000 Logix Designer Files - L5X/* folder in the library.

Device/Item	Add-On Instruction	Rung Import
HI1756xWS	raC_Dvc_HI_1756xWS_1.01.A01.L5X	raC_Dvc_HI_1756_1WS_1.01_RUNG.L5X
		raC_Dvc_HI_1756_2WS_1.01_RUNG.L5X

FactoryTalk® View HMI Files

FactoryTalk® View ME or SE applications require importing the desired device faceplates in addition to all Global Object (ggfx) files and all images located in the */HMI FactoryTalk® View Images - png/* folder of the library. FactoryTalk® View ME files are stored in the */HMI - FactoryTalk® View ME/* library folder and FactoryTalk® View SE files are stored in the */HMI - FactoryTalk® View SE/* library folder.

Device/Item	Type	FactoryTalk® View ME Faceplate	FactoryTalk® View SE Faceplate
HI1756xWS	Display	(raC-1.01-ME) raC_Dvc_HI_1756xWS-Faceplate.gfx	(raC-1.01-SE) raC_Dvc_HI_1756xWS-Faceplate.gfx
Graphic Symbols	Global Object	(raC-1-ME) Graphic Symbols - Hardy Device.ggfx	(raC-1-SE) Graphic Symbols - Hardy Device.ggfx
Toolbox	Global Object	(raC-1-ME) Toolbox - Hardy Device.ggfx	(raC-1-SE) Toolbox - Hardy Device.ggfx

Studio 5000 View Designer® HMI Files

All Studio 5000 View Designer® Files can be found in the */HMI - ViewDesigner - vpd/* folder of the library.

Device/Item	Studio 5000 View Designer® Faceplate
HI1756xWS	(raC-1.01-VD) raC_Dvc_HI.vpd

FactoryTalk® Optix Library Files

FactoryTalk® Optix applications require importing the desired library objects located in the HardyDevice_v1R library folder.

Device/Item	Studio 5000 View Designer® Faceplate
HI1756xWS	raC_1.01.raC_Dvc_HI_1756xWS_UI

Studio 5000® Application Code Manager Files

Studio 5000® Application Code Manager (ACM) can be optionally used if it is installed. All devices can be easily registered in the ACM repositories by running the *setup.cmd* file located in the root folder of the library.

Individual HSL4 files are provided as an alternative to running the *setup.cmd* to allow users to manually register specific implementation objects. Each object has two files - an Asset Control file and a Device file. The Asset Control files include attachments of all required files for that object. The Device files are used to actually add that device into a Studio 5000® project and these reference the Asset Control files.

All Studio 5000® Application Code Manager files can be found in the / *ApplicationCodeManagerLibraries/* folder of the library. The files included are as follows:

Implementation Object	Asset Control File (.HSL4)	Device File (.HSL4)
1756xWS	(RA-LIB)_Device_Asset-Control_HardyProcessSolutions_raC_Dvc_HI_1756xWS_(1.1)	(RA-LIB)_Device_Device_HardyProcessSolutions_raC_LD_Dvc_HI_1756xWS_(1.1)

Device Definition

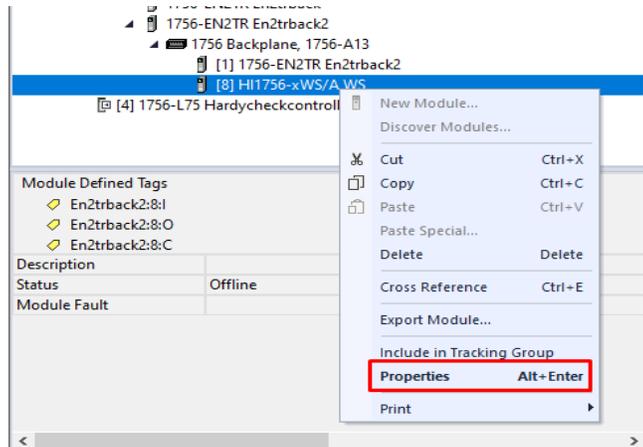
The device (ie: HI1756-ControlLogix® Weigh Scale Module) must be configured with the correct device definition. Proper device configuration enables the required cyclic device data to pass information from the device into the add-on instruction.



Note that this configuration is completed automatically when using Application Code Manager or the Studio 5000® Import Library Objects wizard plug-in.

To verify the device definition:

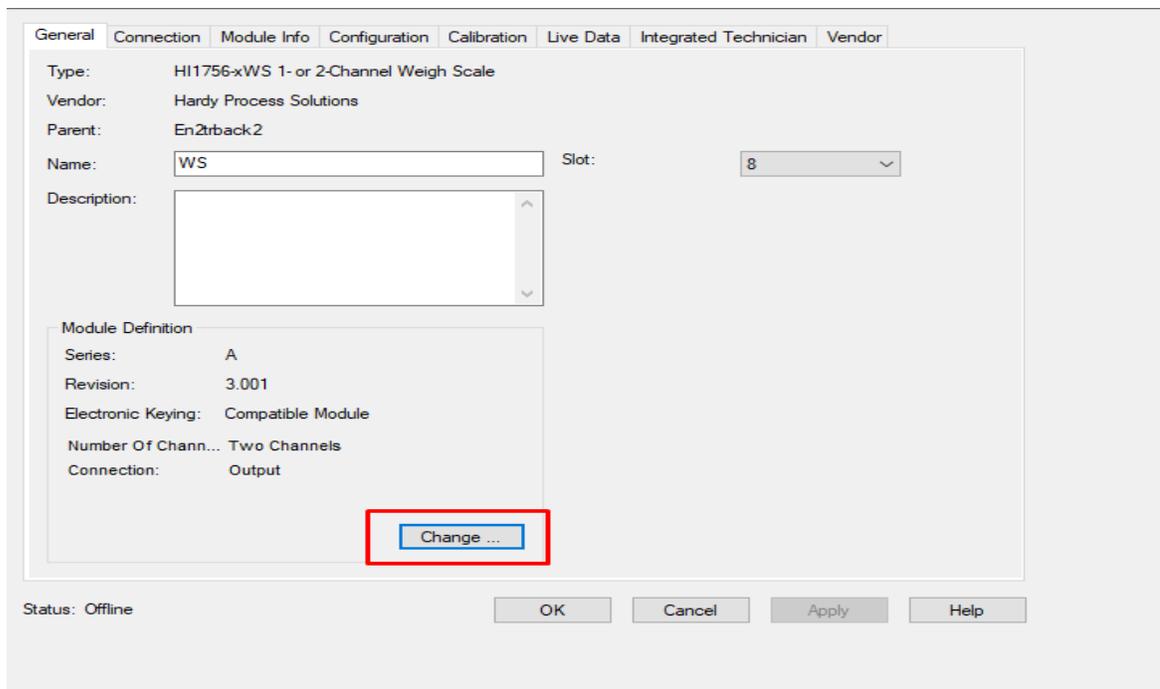
- Find the device in the *Controller Organizer* pane in Studio 5000 Logix Designer® and open the *Module Properties* by double-clicking or right-click and select *Properties*. There are two channel types: one-channel and two-channel.



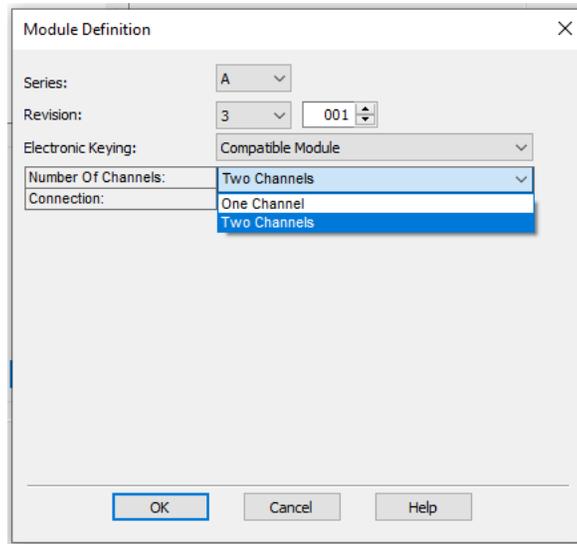
- Refer to the following sections for specific device configuration.

HARDWARE Definition

- On the *General* tab click on the *Change...* button.



- Choose "One channel" for single-channel use. Choose "Two channels" for Two-channel use.



Operations

The Hardy objects provide only physical mode of operation. There is no virtual device mode offered.

Faults & Warnings

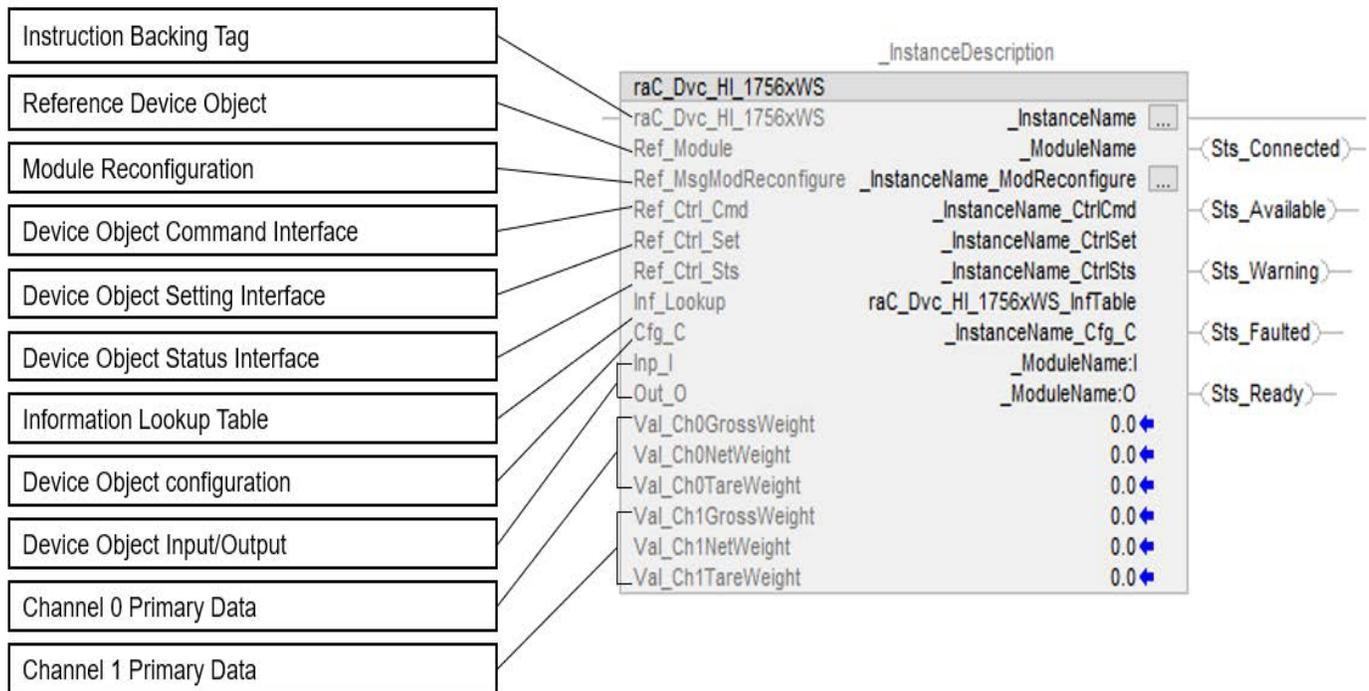
- **First Warning:** This function helps in capturing the first warning triggered in the device. Display the respective description in faceplate.
- **First Fault:** Capture the first fault from device. Display the respective description in faceplate.
- **Event log:** Log Warning and Fault the last 4 events in a log queue. The queue contains fault code, description, and time stamp. Display the same in faceplate.

Execution

The following table explains the handling of instruction execution conditions.

Condition	Description
EnableIn False (false rung)	Processing for EnableIn False (false rung) is handled the same as if the device were taken out of service by Command. The device outputs are de-energized and the device is shown as Program Out of Service on the HMI. All alarms are cleared.
Powerup (prescan, first scan)	On prescan, any commands that are received before first scan are discarded. The device is de-energized. On first scan, the device is treated as if it were returning from Hand command source: the instruction state is set based on the position feedback that is received from the device. If the feedback is valid for one position, the device is set to that position. If the device does not have position feedback or the position feedback is invalid, the device is set to the 'unknown/powerup' state. The command source is set to its default, either Operator or Program (unlocked).
Postscan	No SFC Postscan logic is provided.

Add-On Instruction I/O Data Add-On Instruction Ladder Implementation



InOut Data

InOut	Function / Description	Data Type
Ref_Module	Reference to module in I/O tree	MODULE
Ref_MsgModReconfigure	Message Module Reconfiguration Write	MESSAGE
Ref_Ctrl_Cmd	Hardy Device Command Interface	raC_UDT_ItfAD_Hardy_CtrlCmd
Ref_Ctrl_Set	Hardy Device Setting Interface	raC_UDT_ItfAD_Hardy_CtrlSet
Ref_Ctrl_Sts	Hardy Device Setting Interface	raC_UDT_ItfAD_Hardy_CtrlSts
Inf_Lookup	Code / Description List Entry	raC_UDT_LookupMember_STR0082[3 1]
Cfg_C	Device Object Inputs	raC_UDT_HI_1756xWS_Cfg
Inp_I	Device Object Inputs	HI:1756_xWS:I:0
Out_O	Device Object Output	HI:1756_xWS:O:0

Input Data

Input	Function/Description	Data Type
Cfg_Ch0Active	Channel0 Enable : 0=Disable, 1=Enable	DINT
Cfg_Ch0NumDecPlaces	Channel0 Number of Decimal Places : 0=0, 1=1, 2=2, 3=3, 4=4, 5=5, 6=6	DINT
Cfg_Ch0Units	Channel0 Units : 0=lb, 1=Kg	DINT
Cfg_Ch0WaverSaver	Channel0 WaveSaver : 0=7.5 Hz, 1=3.5 Hz, 2=1 Hz, 3=0.5 Hz, 4=0.25 Hz	DINT
Cfg_Ch0ZeroTrkEn	Channel0 Zero Trk En : 0=NotEnable, 1=Enable	DINT
Cfg_Ch1Active	Channel1 Enable : 0=Disable, 1=Enable	DINT
Cfg_Ch1NumDecPlaces	Channel1 Number of Decimal Places : 0=0, 1=1, 2=2, 3=3, 4=4, 5=5, 6=6	DINT
Cfg_Ch1Units	Channel1 Units : 0=lb, 1=Kg	DINT
Cfg_Ch1WaverSaver	Channel1 WaveSaver : 0=7.5 Hz, 1=3.5 Hz, 2=1 Hz, 3=0.5 Hz, 4=0.25 Hz	DINT
Cfg_Ch1ZeroTrkEn	Channel1 Zero Trk En : 0=NotEnable, 1=Enable	DINT
Cmd_ApplyConfiguration	Module reconfigure Command	BOOL
EnableIn	Enable Input - System Defined Parameter	BOOL
Set_Ch0AutoZeroTol	Channel0 Setpoint of Auto Zero Tolerance	REAL
Set_Ch0CalLowWt	Channel0 Setpoint of Calibration Low Weight	REAL
Set_Ch0DiscreteCmds	Channel0 Tare and Zero cmd 2=Tare cmd 1=Zero Cmd	INT
Set_Ch0MotionTol	Channel0 Setpoint of Motion Tolerance	REAL
Set_Ch0NumAvg	Channel0 Setpoint of Num Averages	DINT
Set_Ch0RocTb	Channel0 Setpoint of ROC Timebase	DINT
Set_Ch0SpanWt	Channel0 Setpoint of Span Weight	REAL
Set_Ch0TareWeight	Channel0 Setpoint of Tare Weight	REAL
Set_Ch0ZeroTol	Channel0 Setpoint of Zero Tolerance	REAL
Set_Ch1AutoZeroTol	Channel1 Setpoint of Auto Zero Tolerance	REAL
Set_Ch1CalLowWt	Channel1 Setpoint of Calibration Low Weight	REAL
Set_Ch1DiscreteCmds	Channel1 Tare and Zero cmd 2=Tare cmd 1=Zero Cmd	INT
Set_Ch1MotionTol	Channel1 Setpoint of Motion Tolerance	REAL

Input	Function/Description	Data Type
Set_Ch1NumAvg	Channel Setpoint of Num Averages	DINT
Set_Ch1RocTb	Channel Setpoint of ROC Timebase	DINT
Set_Ch1SpanWt	Channel Setpoint of Span Weight	REAL
Set_Ch1TareWeight	Channel Setpoint of Tare Weight	REAL
Set_Ch1ZeroTol	Channel Setpoint of Zero Tolerance	REAL
Set_ChTrendMargin	Channel Trend Margin Setpoint	DINT

Output Data

Output	Function/Description	Data Type
Sts_Active	Device active status: 1 = output power structure is active	BOOL
Sts_ApplyConfiguration	Module Reconfigure Request On Status	BOOL
Sts_Available	Device is available for interaction with user code	BOOL
Sts_bNotReady	Bitwise device 'not ready' reason 0 = Reserved 1 = Device not connected 2 = Device not available 3 = Device Faulted 4 - 31 = Reserved 1 = Device not connected 2 = Device not available 3 = Device Faulted 4 - 31 = Reserved	DINT
Sts_Ch0ADCCConvertError	Channel0 ADC Convert Error	BOOL
Sts_Ch0ADCErr	Channel0 ADC Error	BOOL
Sts_Ch0ADCFailure	Channel0 ADC Failure	BOOL
Sts_Ch0CommFailure	Channel0 Comm Failure	BOOL
Sts_Ch0EEPromWriteError	Channel0 EEPROM write Error	BOOL
Sts_Ch0Enable	Channel0 Enable: 0=Disable, 1=Enable	BOOL
Sts_Ch0ExcitationError	Channel0 Excitation Error	BOOL
Sts_Ch0FaultActive	Channel0 Fault Active	BOOL
Sts_Ch0Motion	Channel0 Weight is unstable	BOOL
Sts_Ch0UnitDisplay	Channel0 Unit	BOOL
Sts_Ch1ADCCConvertError	Channel1 ADC Convert Error	BOOL
Sts_Ch1ADCErr	Channel1 ADC Error	BOOL
Sts_Ch1ADCFailure	Channel1 ADC Failure	BOOL
Sts_Ch1CommFailure	Channel1 Comm Failure	BOOL
Sts_Ch1EEPromWriteError	Channel1 EEPROM write Error	BOOL
Sts_Ch1Enable	Channel1 Enable: 0=Disable, 1=Enable	BOOL
Sts_Ch1ExcitationError	Channel1 Excitation Error	BOOL
Sts_Ch1FaultActive	Channel1 Fault Active	BOOL
Sts_Ch1Motion	Channel1 Motion	BOOL
Sts_Ch1UnitDisplay	Channel1 Unit	BOOL
Sts_CommunicationOk	Communication Between Controller & 1756xWS working OK	BOOL
Sts_Connected	Device is connected to the Programmable Automation Controller	BOOL
Sts_Failure	Application fault; predictive diagnostics alarm triggered or command cannot be executed as requested	BOOL
Sts_Faulted	Device faulted status: 1 = an active fault exists	BOOL
Sts_InhibitCfg	Disable Configuration inputs from external sources	BOOL

Output	Function/Description	Data Type
Sts_InhibitCmd	Disable Command inputs from external sources	BOOL
Sts_InhibitSet	Disable Setting inputs from external sources	BOOL
Sts_Ready	Device is ready to perform primary function	BOOL
Sts_Warning	Device warning status: 1 = an active alarm or warning exists	BOOL
Val_Ch0AutoZeroTol	Channel0 Auto Zero Tolerance	REAL
Val_Ch0CalLowWt	Channel0 Calibration Low Weight	DINT
Val_Ch0CalStNew	Channel0 Cal status	DINT
Val_Ch0ChActive	Channel0 Enable : 0=Disable, 1=Enable	DINT
Val_Ch0CmdStatus	Channel0 Command status	DINT
Val_Ch0GrossWeight	Channel0 Gross Weight	REAL
Val_Ch0InstStatus	Channel0 Status	DINT
Val_Ch0MaxTrendLimit	Channel0 Maximum Trend Limit	REAL
Val_Ch0Metric	Channel0 Units : 0=lb, 1=Kg	DINT
Val_Ch0MinTrendLimit	Channel0 minimum Trend Limit	REAL
Val_Ch0MotionTol	Channel0 Motion Tolerance	REAL
Val_Ch0NetWeight	Channel0 Net Weight	REAL
Val_Ch0NumAvg	Channel0 Num Averages	DINT
Val_Ch0NumDecPlaces	Channel0 Number of Decimal Places : 0=0, 1=1, 2=2, 3=3, 4=4, 5=5, 6=6	DINT
Val_Ch0ROC	Channel0 ROC	REAL
Val_Ch0RocTb	Channel0 ROC Timebase	DINT
Val_Ch0SpanWt	Channel0 Span Weight	REAL
Val_Ch0TareWeight	Channel0 Tare Weight	REAL
Val_Ch0TraditionCalStNew	Channel0 Traditional Calibration	DINT
Val_Ch0WaverSaver	Channel0 WaveSaver: 0=7.5 Hz, 1=3.5 Hz, 2=1 Hz, 3=0.5 Hz, 4=0.25 Hz	DINT
Val_Ch0ZeroTol	Channel0 Zero Tolerance	REAL
Val_Ch0ZeroTrkEn	Channel0 Zero Trk En: 0=NotEnable, 1=Enable	DINT
Val_Ch1AutoZeroTol	Channel1 Auto Zero Tolerance	REAL
Val_Ch1CalLowWt	Channel1 Calibration Low Weight	REAL
Val_Ch1CalStNew	Channel1 Cal status	DINT
Val_Ch1ChActive	Channel1 Enable: 0=Disable, 1=Enable	DINT
Val_Ch1CmdStatus	Channel1 Command status	INT
Val_Ch1GrossWeight	Channel1 Gross Weight	REAL
Val_Ch1InstStatus	Channel1 Status	DINT

Output	Function/Description	Data Type
Val_Ch1MaxTrendLimit	Channel Maximum Trend Limit	REAL
Val_Ch1Metric	Channel Units: 0=lb, 1=Kg	DINT
Val_Ch1MinTrendLimit	Channel minimum Trend Limit	REAL
Val_Ch1MotionTol	Channel Motion Tolerance	REAL
Val_Ch1NetWeight	Channel Net Weight	REAL
Val_Ch1NumAvg	Channel Num Averages	DINT
Val_Ch1NumDecPlaces	Channel Number of Decimal Places : 0=0, 1=1, 2=2, 3=3, 4=4, 5=5, 6=6	DINT
Val_Ch1ROC	Channel ROC	REAL
Val_Ch1RocTb	Channel ROC Timebase	DINT
Val_Ch1SpanWt	Channel Span Weight	REAL
Val_Ch1TareWeight	Channel Tare Weight	REAL
Val_Ch1TraditionCalStNe w	Channel Maximum Trend Limit	DINT
Val_Ch1Waversaver	Channel WaveSaver: 0=7.5 Hz, 1=3.5 Hz, 2=1 Hz, 3=0.5 Hz, 4=0.25 Hz	DINT
Val_Ch1ZeroTol	Channel Zero Tolerance	REAL
Val_Ch1ZeroTrkEn	Channel Zero Trk En: 0=NotEnable, 1=Enable	DINT

Data Types

The following Hardy Common Control Interface tags are the primary device program tags to read and write to when interfacing to Hardy devices. The value of using these tags in your specific application code is that you may use a number of different Hardy devices such as 1756xWS without having to update your application device interface tags.

Refer to the [Interfaces](#) section for detailed information on interfaces.

raC_UDT_ItfAD_Hardy_CtrlSet

This is the Hardy Common Control Interface User-Defined Data Type for device settings. Its members provide application program access to allow or inhibit commands and settings from the device faceplate or other external sources. The table below shows member names, descriptions, and tag data types.

For example, to inhibit write commands from the device faceplate or other external sources write a 1 to the `_InstanceName_CtrlSet.InhibitCmd` program tag from your application program. This would prevent a Clear Tare command from the device faceplate. You may also set the Pre-Tare Value for the device.

Member	Description	Data Type
InhibitCmd	1 = Inhibit user Commands from external sources, 0 = Allow.	BOOL
InhibitSet	1 = Inhibit user Settings from external sources, 0 = Allow.	BOOL
InhibitCfg	1 = Inhibit user Configuration from external sources, 0 = Allow.	BOOL
Ch0TareValue	Channel0 Setpoint of Tare Value	REAL
Ch1TareValue	Channel1 Setpoint of Tare Value	REAL
Ch0Units	Channel0 Units 0=lb , 1=Kg	INT
Ch0Units	Channel1 Units 0=lb, 1=Kg	INT
ReserveSetDINT1	ReserveSetDINT1	DINT
ReserveSetDINT2	ReserveSetDINT2	DINT
ReserveSetREAL1	ReserveSetREAL1	REAL
ReserveSetREAL2	ReserveSetREAL2	REAL

raC_UDT_ItfAD_Hardy_CtrlCmd

This is the Hardy Common Control Interface User-Defined Data Type for device commands. Its members provide application program access to common device commands.

Only write to these common command members to control the device. If you write directly to the device’s output command tags directly unexpected device operation could occur.

For example, to tare the weight write a 1 to the `_InstanceName_CtrlCmd.TareImmediate`. Although, you can write to the uncommon command tags in the device’s output tag if a specific common control interface tag does not exist.

The table below shows member names, descriptions, and tag data types.

Member	Description	Data Type
bCmd	Commands (Bit Overlay).	INT
ResetWarn	1 = Reset device warning [No warning reset].	BOOL
ResetFault	1 = Reset device trip or fault [No Fault reset,- Automatic fault reset only].	BOOL
Physical	1 = Operate as Physical Device - hold for future use.	BOOL
Virtual	Virtual mode not implemented - hold for future use.	BOOL
Ch0Tare	1 = Trigger execution of Tare Command	BOOL
Ch0Zero	1 = Trigger execution of Zero Command	BOOL
Ch0C2Cal	1 = Trigger Ch0 C2 Calibration	BOOL
Ch0LoCal	1 = Trigger Ch0 Low Cal	BOOL
Ch0HiCal	1 = Trigger Ch0 High Cal	BOOL
Ch1Tare	1 = Trigger execution of Tare Command	BOOL
Ch1Zero	1 = Trigger execution of Zero Command	BOOL
Ch1C2Cal	1 = Trigger Ch1 C2 Calibration	BOOL
Ch1LoCal	1 = Trigger Ch1 Low Cal	BOOL

Member	Description	Data Type
Ch1HiCal	1 = Trigger Ch1 High Cal	BOOL
Reserve1	Reserved 1	BOOL
Reserve2	Reserved 2	BOOL
Reserve3	Reserved 3	BOOL
Reserve4	Reserved 4	BOOL

raC_UDT_ItfAD_Hardy_CtrlSts

This is the Hardy Common Control Interface User-Defined Data Type for device status. Its members provide application program access to device states, status, and diagnostic data. The table below shows member names, descriptions, and tag data types.

Input	Description	Data Type
eState	Enumerated state value: 0 = Unused, 1 = Initializing, 2 = Disconnected, 3 = Disconnecting, 4 = Connecting, 5 = Idle, 6 = Configuring, 7 = Available.	DINT
FirstWarning	First Warning Event Data.	raC_UDT_Event
FirstFault	First Fault Event Data.	raC_UDT_Event
eCmdFail	Enumerated command failure code. See extended help for enumeration values.	DINT
bSts	Status (Bit Overlay). 0 = Connected, 1 = Available, 2 = Warning, 3 = Faulted, 4 = Ready, 5 = Active.	DINT
Connected	1 = PAC to device connection has been established.	BOOL
Available	1 = The device is available for interaction with the user program.	BOOL
Warning	1 = A warning is active on the device.	BOOL
Faulted	1 = A fault is active on the device.	BOOL
Physical	1 = Controlling physical device.	BOOL
Virtual	1 = Controlling virtual device.	BOOL
Ch0GrossWeight	Channel0 Gross Weight	Real
Ch0NetWeight	Channel0 Net Weight	Real
Ch0TareWeight	Channel0 Tare Weight	Real
Ch0Units	Channel0 Units 0=lb 1=Kg	INT
Ch1GrossWeight	Channel1 Gross Weight	Real
Ch1NetWeight	Channel1 Net Weight	Real
Ch1TareWeight	Channel1 Tare Weight	Real
Ch1Units	Channel1 Units 0=lb 1=Kg	INT
ReserveSetDINT1	ReserveSetDINT1	DINT
ReserveSetDINT2	ReserveSetDINT2	DINT
ReserveSetREAL1	ReserveStatusREAL1	REAL
ReserveStatusREAL2	ReserveStatusREAL2	REAL

raC_UDT_Event

An array of size 4 is to be used to log the FirstWarning and FirstFault capture. The data should be FIFO order. The same should be displayed on the Faceplate.

Member	Description	Data Type
Type	Event type: 1 = Status, 2 = Warning, 3 = Fault, 4...n = User.	DINT
ID	User definable event ID.	DINT
Category	User definable category (Electrical, Mechanical, Materials, Utility, etc.).	DINT
Action	User definable event action code.	DINT
Value	User definable event value or fault code.	DINT
Message	Event message text.	STRING
EventTime_L	Timestamp (Date/Time format).	LINT
EventTime_D	Timestamp (Y,M,D,h,m,s,us).	DINT[7]

raC_UDT_HI_1756xWS_Cfg

The Hardy configuration Control Interface UDT serves as a structured way to represent and manage device status information within the Hardy system.

It encapsulates device-specific attributes and provides an interface for both reading (monitoring) and writing (controlling) these attributes.

Member	Description	Data Type
Type	Event type: 1 = Status, 2 = Warning, 3 = Fault, 4...n = User.	DINT
ChOCopyConfigEnable	ChannelO Copy Configuration Enable : 0=Disable, 1=Enable	raC_UDT_HI_1756xWS_CfgBoolData
ChOChannelEnable	ChannelO Enable : 0=Disable, 1=Enable	raC_UDT_HI_1756xWS_CfgBoolData
ChONumDecPlaces	ChannelO Number of Decimal Places : 0=0, 1=1, 2=2, 3=3, 4=4, 5=5, 6=6	DINT
ChOMetric	ChannelO Num Averages	DINT
ChONumAverages	ChannelO Number of Decimal Places : 0=0, 1=1, 2=2, 3=3, 4=4, 5=5, 6=6	DINT
ChOWaversaver	0=7.5 Hz, 1=3.5 Hz, 2=1 Hz, 3=0.5 Hz, 4=0.25 Hz	DINT
ChOSpanWeight	ChannelO Span Weight	DINT
ChOCalLowWeight	ChannelO Calibration Low Weight	DINT
ChOZeroTrackEnable	ChannelO Zero Trk En : 0=NotEnable, 1=Enable	raC_UDT_HI_1756xWS_CfgBoolData
ChOAutoZeroTolerance	ChannelO Auto Zero Tolerance	DINT
ChOMotionTolerance	ChannelO Motion Tolerance	DINT

Member	Description	Data Type
Ch0TareWeight	Channel0 Tare Weight	DINT
Ch0ButtonCalibEnable	Channel0 Button Calibration Enable : 0=Disable, 1=Enable	raC_UDT_HI_1756xWS_CfgBoolData
Ch0ROCTimeBase	Channel0 ROC Time Base	DINT
Ch0ZeroTolerance	Channel0 Zero Tolerance	DINT
Ch0Spare	Channel0 Spare	DINT
Ch1CopyConfigEnable	Channel1 Copy Configuration Enable : 0=Disable, 1=Enable	raC_UDT_HI_1756xWS_CfgBoolData
Ch1ChannelEnable	Channel1 Enable : 0=Disable, 1=Enable	raC_UDT_HI_1756xWS_CfgBoolData
Ch1NumDecPlaces	Channel1 Number of Decimal Places : 0=0, 1=1, 2=2, 3=3, 4=4, 5=5, 6=6	DINT
Ch1Metric	Channel1 Num Averages	DINT
Ch1NumAverages	Channel1 Number of Decimal Places : 0=0, 1=1, 2=2, 3=3, 4=4, 5=5, 6=6	DINT
Ch1Waversaver	0=7.5 Hz, 1=3.5 Hz, 2=1 Hz, 3=0.5 Hz, 4=0.25 Hz	DINT
Ch1SpanWeight	Channel1 Span Weight	DINT
Ch1CalLowWeight	Channel1 Calibration Low Weight	
Ch1ZeroTrackEnable	Channel1 Zero Trk En : 0=NotEnable, 1=Enable	raC_UDT_HI_1756xWS_CfgBoolData
Ch1AutoZeroTolerance	Channel1 Auto Zero Tolerance	DINT
Ch1MotionTolerance	Channel1 Motion Tolerance	DINT
Ch1TareWeight	Channel1 Tare Weight	DINT
Ch1ButtonCalibEnable	Channel1 Button Calibration Enable : 0=Disable, 1=Enable	raC_UDT_HI_1756xWS_CfgBoolData
Ch1ROCTimeBase	Channel1 ROC TimeBase	DINT
Ch1ZeroTolerance	Channel1 Zero Tolerance	DINT
Ch1Spare	Channel1 Spare	DINT

raC_UDT_LookupMember_STR0082

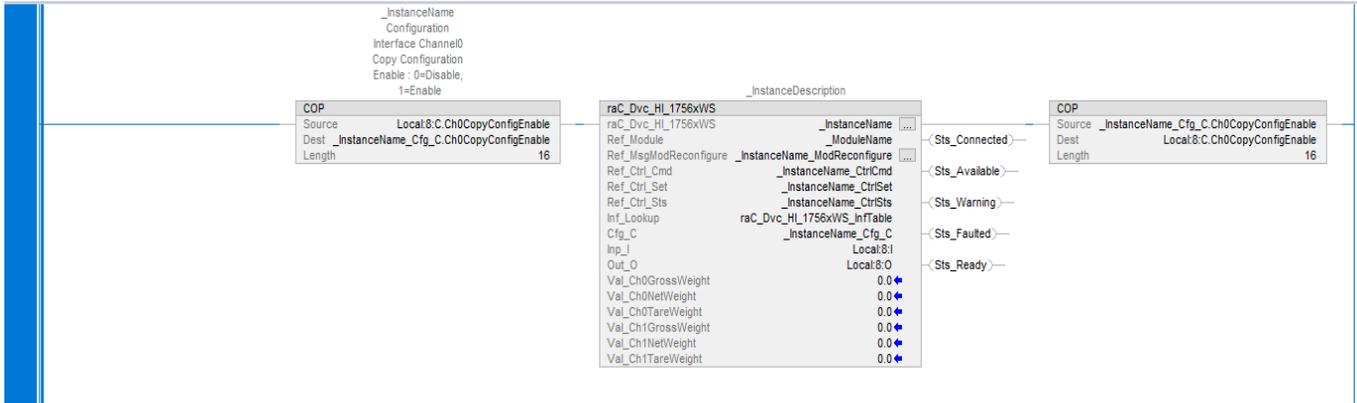
Member	Description	Data Type
Code	Code	DINT
Desc	Code Description	STRING

Programming Example

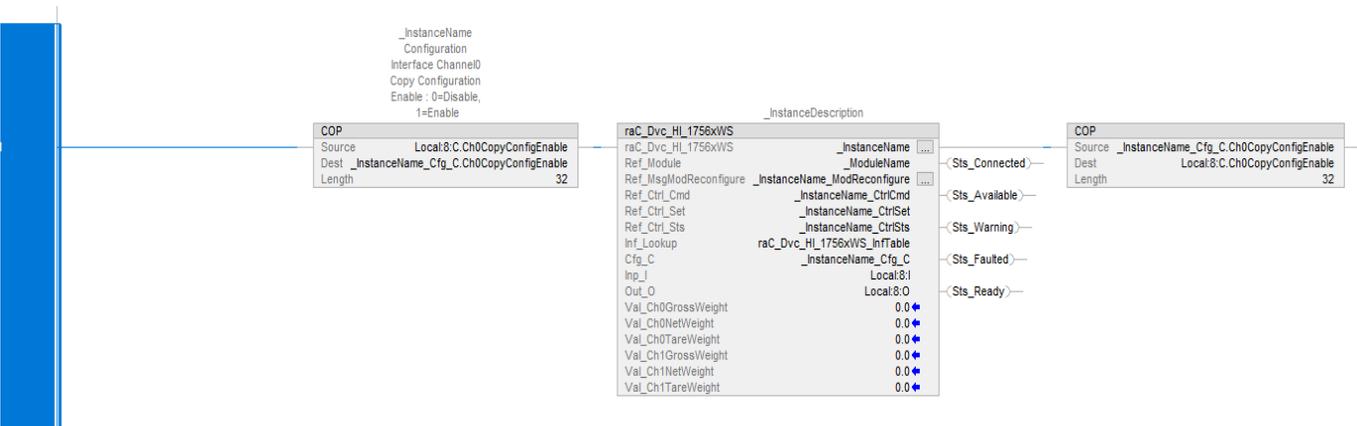
Fully configured device on a rung is provided below for reference. This example includes the device objects for a HI1756 - ControlLogix® Weigh Scale Module (raC_Dvc_HI_1756xWS).

Note that this programming example is the same code that is imported when either importing the supplied rung .L5X files or when using Application Code Manager or the Studio 5000® Import Library Objects wizard plug-in.

When you configure a one-channel import, be sure to import the supplied rung raC_Dvc_HI_1756_1WS_1.01_RUNG



When you configure a Two-channel import, be sure to import the supplied rung raC_Dvc_HI_1756_2WS_1.01_RUNG



The device (ie: HI1756 - ControlLogix® Weigh Scale Module) must also be configured with the correct device definition. Note that this configuration is completed automatically when using Application Code Manager or the Studio 5000® Import Library Objects wizard plug-in. For details on setting up the device, refer to the [Device Definition](#) section.

Graphic Symbols

Graphic Symbols are used as launch buttons within HMI applications to open up faceplate displays. Alternatively, faceplates may also be launch from related instructions such as the navigate to device faceplate buttons in the Process Library or the Machine Builder Library faceplates.

All icons display the following information:

- Device label (Tag.@Description or custom label entered in parameter #104)
- Device Warning/Fault Indication
- Device not ready indication

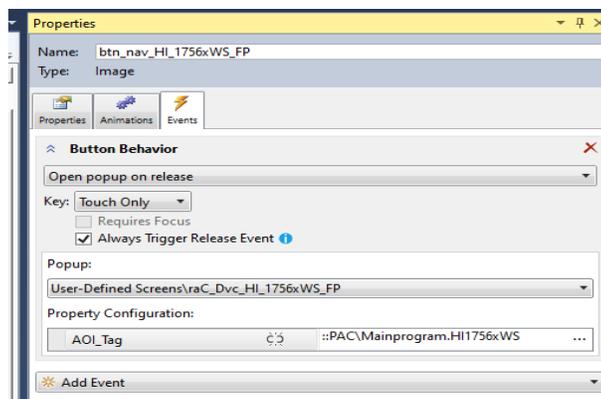
See [Launch Buttons](#) for more general information on launch button diagnostics and usage.

FactoryTalk® View ME/SE Graphic Symbols

Graphic Symbol Name	Graphic Symbol	Description	Global Object Parameter Values
GO_LaunchFP		Faceplate navigation button with string tag label. This launch button graphic object allows the user to navigate to the device object faceplate.	#102: Backing Tag (e.g. {[PAC]Program::Program_ InstanceName}) #104: Custom button label. Leave blank to use Tag.@Description #120: Display's left position (e.g. 100, optional) #121: Display's top position (e.g. 100, optional)

Studio 5000 View Designer® Graphic Symbols

All Studio 5000 View Designer® graphic symbols must be configured with an *Event* to open up the appropriate Popup screen. Select the graphic symbol and in the *Properties* window navigate to the *Events* tab. Assign a *Button Behavior* event to *Open popup on release*. Assign the required Popup screen (e.g. User-Defined Screens\raC_Dvc_HI_1756xWS_FP). The required *Property Configurations* are found in the following table where you may assign the *AOI_Tag* to the object's Add-On Instruction tag.



Graphic Symbol Name	Graphic Symbol	Description	Property Configuration
Launch_HI_1756xWS_FP		Faceplate navigation button with string tag label. Use Properties > General > Text to modify the button label text.	
AOG_Generic_Launch_001		Faceplate navigation button with string tag label. Use Properties > General > Text to modify the button label text.	AOL_Tag: Object's Add-On Instruction Tag

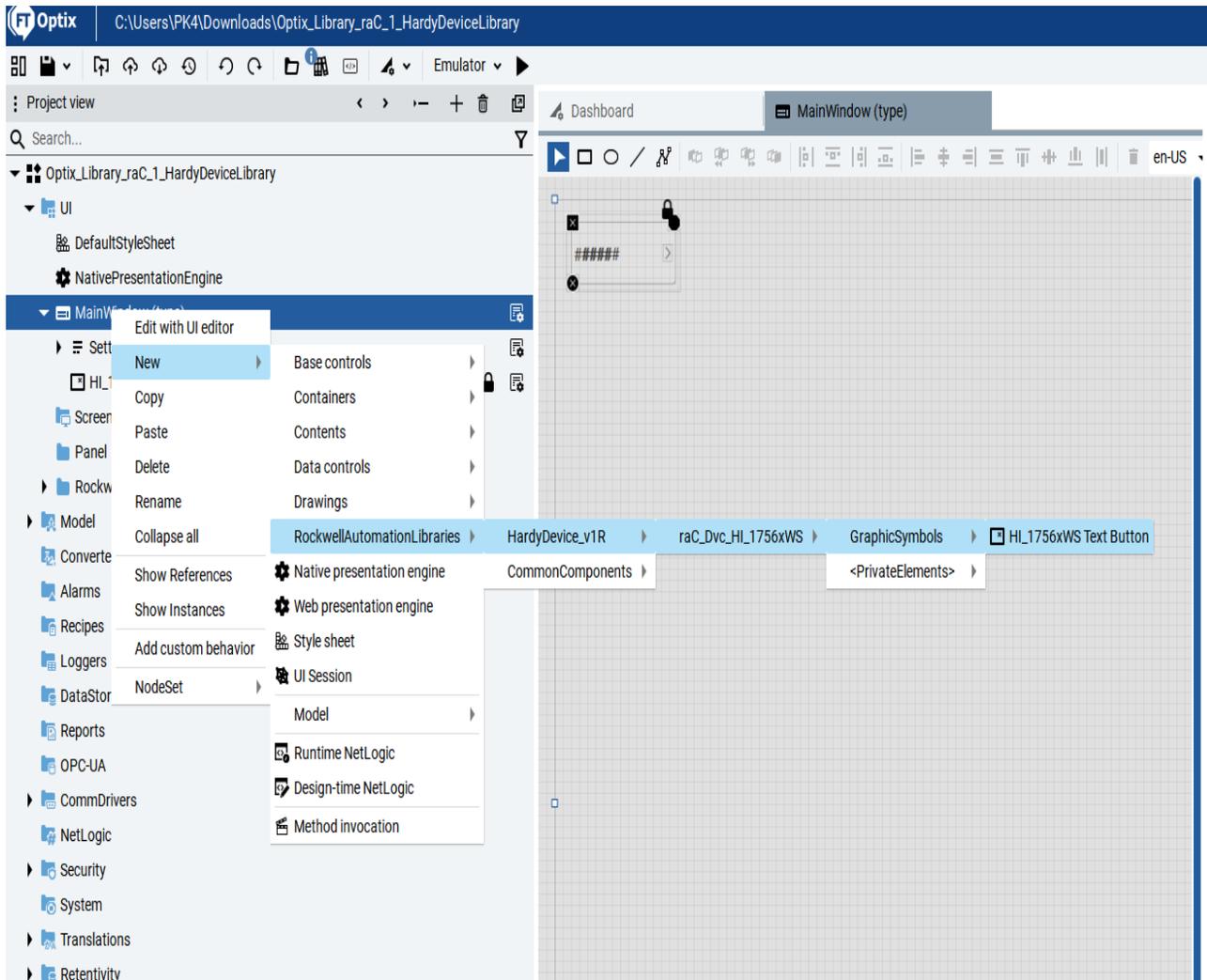
FactoryTalk® Optix Graphic Symbols

Graphic Symbols are used as launch buttons within HMI applications to open up faceplate displays. All graphical symbols for Hardy Devices display the following information:

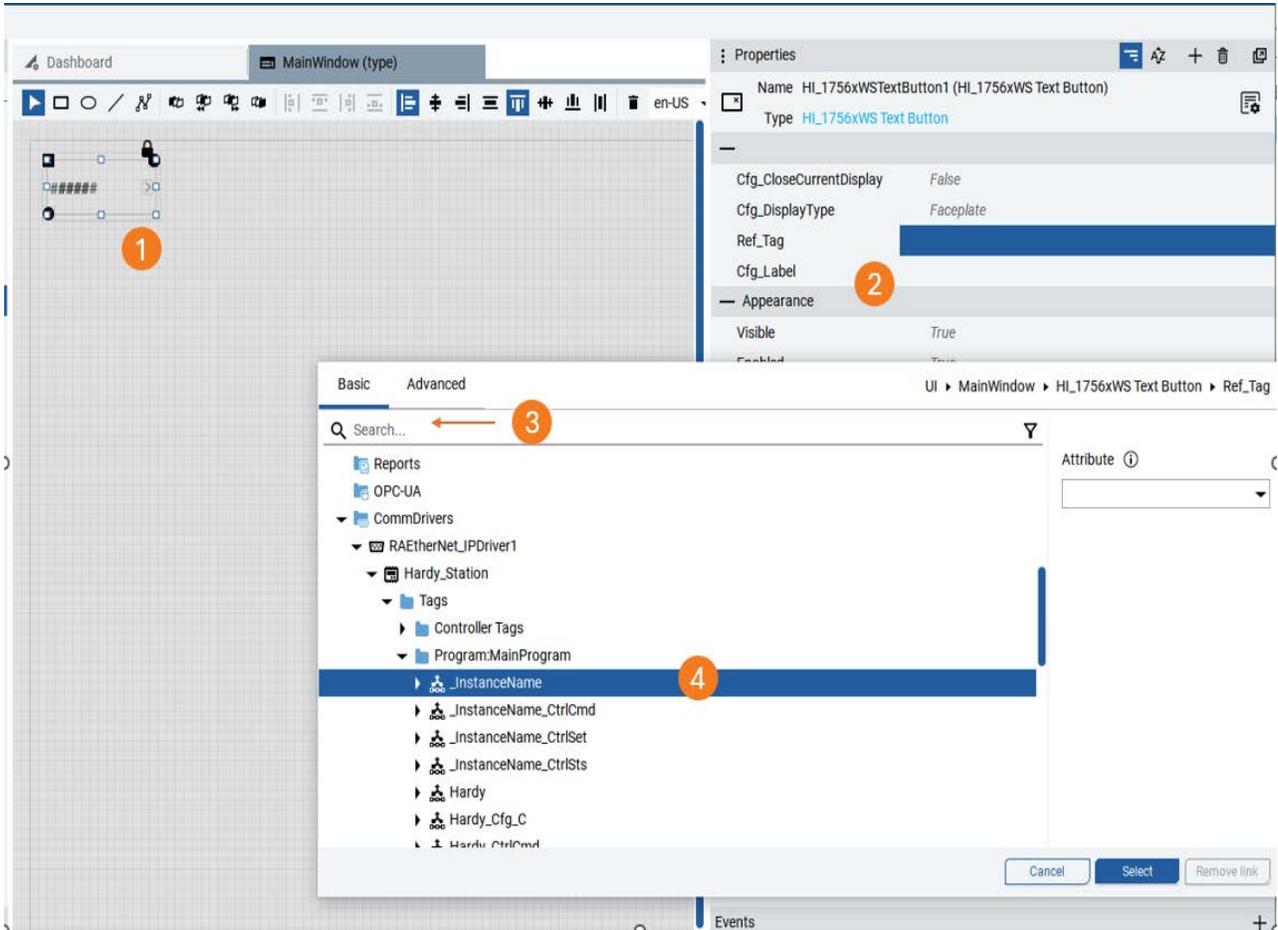
- - Device label (Tag.@Description or custom label)
- -Device Warning/Fault Indication
- - Device not ready indication
- -Device Active/Inactive

See [Basic Launch Button Attributes](#) section for more general information on launch button diagnostics and usage.

Once the Objects have been imported into the FactoryTalk® Optix Studio project, you can begin using them in your application. To add a new Launch Button to a Main window, navigate to raC_1_xx_raC_Dvc_ObjectName_UI > Graphic Symbols > raC_1_xx_raC_Dvc_ObjectName_GS_NavText Button to insert a navigation launch button with a text label.



After placing the graphic symbol on a UI panel, link the “Ref_Tag” property to the targeted Asset under Asset tag Text label shown on button can be configured using “Cfg_Label” property, If it is not configured then description of the asset will be shown on the button face.



This is the only step needed to link the UI to the asset data model. For more information on graphic symbols, refer to the Graphic Symbols section of the Hardy device type in this manual.

Graphic Symbol Name	Graphic Symbol	Description	Global Object Parameter Values
raC_1_xx_raC_Dvc_Devicename_GS _NavText		Faceplate navigation button. Use Cfg_Label Variable to modify the button label text.	<p>Cfg_CloseCurrentDisplay: Set to 'True' to close the previously open display when launching the object faceplate</p> <p>Cfg_DisplayType: Faceplate to be opened on button click. This should not be modified.</p> <p>Ref_Tag: Object's Add-On Instruction Tag</p> <p>Cfg_Label: Text label shown on the button face</p>

Variable Name	Description	Default Value
Cfg_CloseCurrentDisplay	Set to 'True' to close the previously open display when launching the object faceplate	False
Cfg_DisplayType	Faceplate to be opened on button click. This should not be modified	Faceplate
Ref_Tag	Link to instance of desired target Asset model found in Model > Asset folder.	N/A - User must configure
Cfg_Label	Text label shown on the button face. Defaults to the description of the asset but users may replace in instances with other desired text.	Ref_Tag@Description

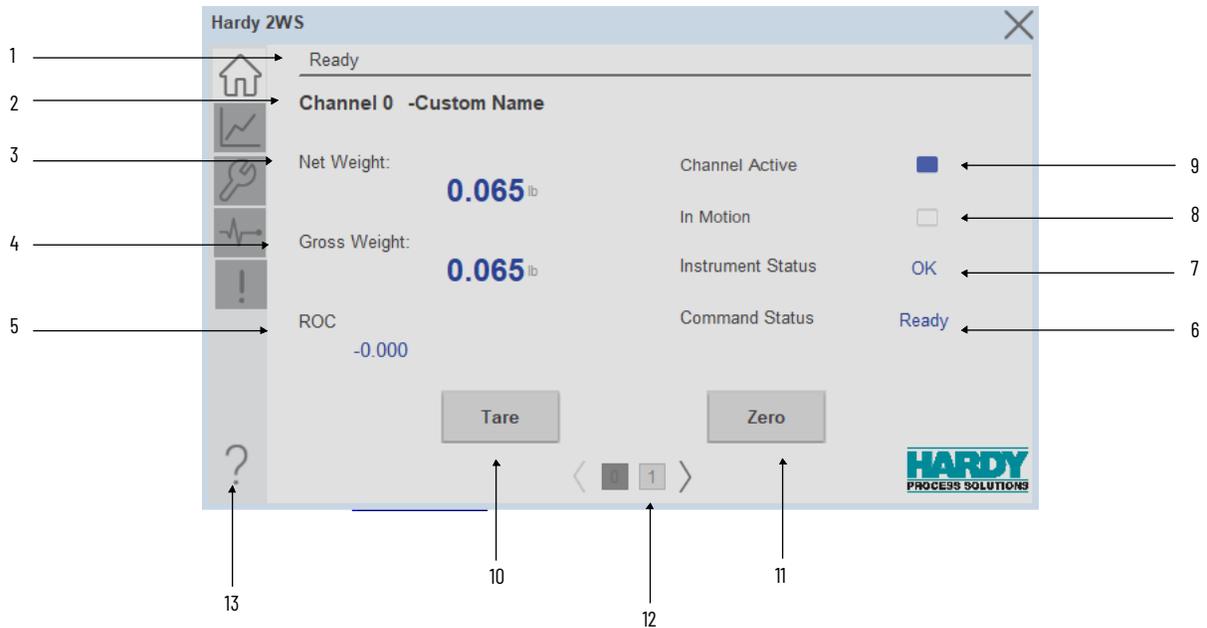
Faceplates

There are basic faceplate attributes that are common across all instructions. See [Basic Faceplate Attributes on page 17](#).

The faceplate title is linked to _InstanceName.@description, the .@description extended tag property of the Add-On Instruction instance. This is user configurable from controller/program tags in Studio 5000 Logix Designer.

Home Tab

The Home tab is the main tab of the faceplate. It contains Primary weight parameters as well weighing terminal parameters of the device, Device status information and primary commands of the device.



Item	Description
1	Banner
2	Channel number with custom naming
3	Net Weight
4	Gross Weight
5	ROC
6	Command status : When the tare command or zero command is passed, it will show the status
7	Instrument status: It will show the current instrument status
8	0 = No motion ; 1 = In motion
9	0 = Channel not active; 1 = Channel active
10	The "Tare button" is pressed to initiate the "tare command" regardless of the stability of the weight value
11	The "Zero button" is pressed to execute the "zero command" regardless of the stability of the weight value. This feature is specifically designed for making minor adjustments to the "zero point" to compensate for drifting
12	Tab Once available in same as channel Zero
13	Help File button

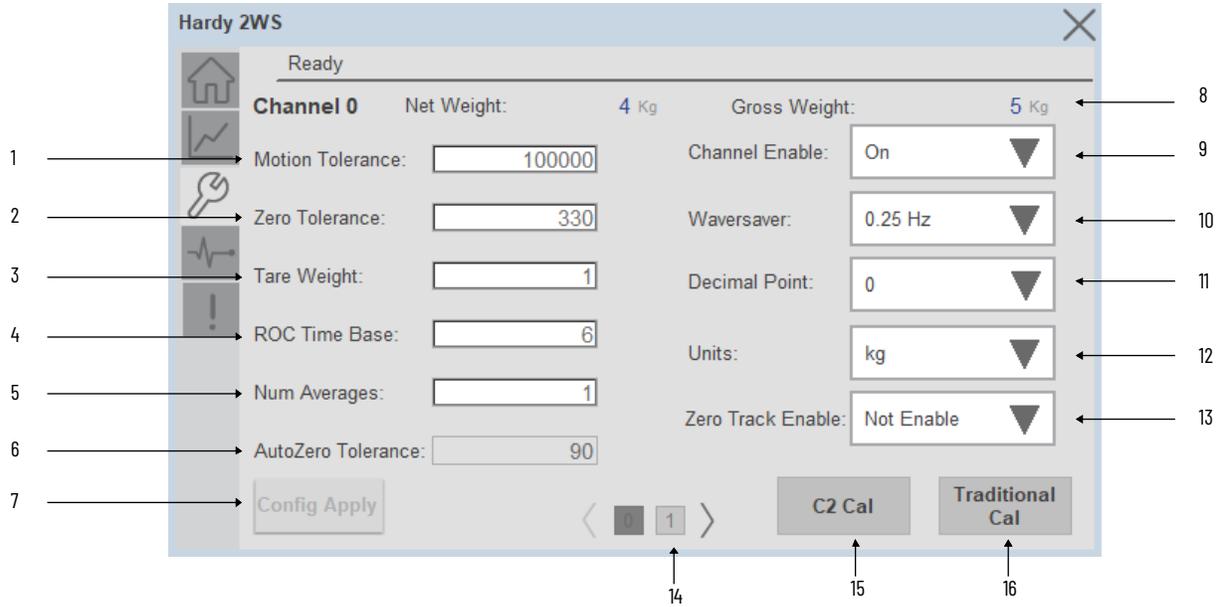
Trend Tab

Trends display values over time, often used to compare similar or related values and to allow operators to predict future states to make control action decisions. There are total two trends are displayed as follows Net Weight, Gross Weight.



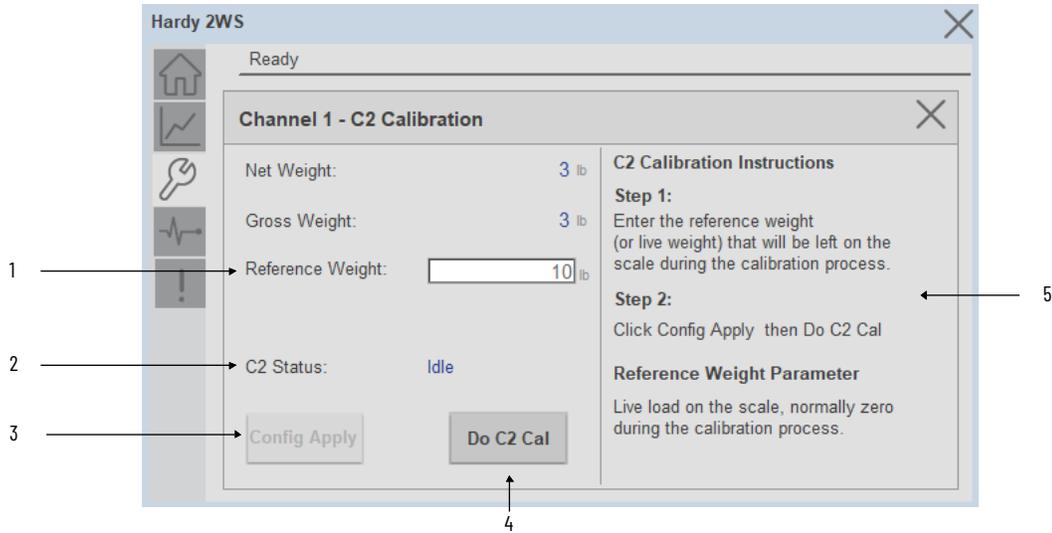
Configure Tab

The Configure tab acts as a control center for technicians performing maintenance. It offers various settings that can be adjusted to fine-tune the performance of an object managed on another tab. These settings include enabling or disabling channels, utilizing the Waversaver® function, selecting the preferred decimal point and unit format, activating zero track functionality, and defining tolerances for motion, zero point, and automatic zeroing. Additionally, the technician can set the tare weight, ROC time base, and the number of averages used for calculations. To ensure optimal performance, the tab also provides navigation buttons for calibrating the device.



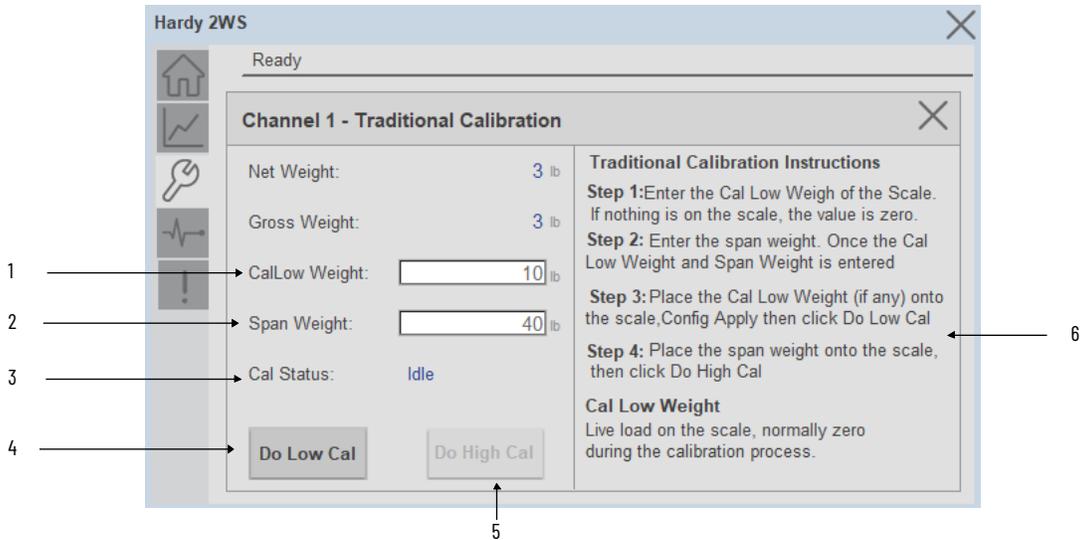
Item	Description
1	Setpoint of Motion Tolerance
2	Setpoint of Zero Tolerance
3	Setpoint of Tare weight
4	Setpoint of ROC Time Base :- Legal Values 1-1800 seconds
5	Setpoint of Num Averages :- Legal Values 1-255
6	The AutoZero Tolerance setting becomes active when the Zero Track Enable selection is turned on
7	The "Config Apply" button remains disabled until a change is made within the Configuration tab. Once any configuration element is modified (e.g., enabling a channel, adjusting motion tolerance), the button becomes active, prompting the user to apply the changes
8	Real time Gross weight
9	Channel Enable and disable.
10	WAVERSAVER® can be configured to ignore noise with frequencies as low as 0.25 Hz. One of five higher additional cut off frequencies may be selected to provide a faster instrument response time. The default factory configuration is 1.00 Hz vibration frequency immunity.
11	Decimal point selection
12	Units selection
13	Zero Track Enable
14	Channel navigation :Tab Once available in same as channel Zero
15	Navigate the C2 Calibration screen
16	Navigate Traditional Calibration screen

C2 Calibration



Item	Description
1	Reference Weight:- The C2 calibration command executes a calibration process for the C2 system. It employs the CalLow Weight value as a starting point and progresses through the weight range from zero to the maximum span weight.
2	C2 Status: Displays the real-time state of the C2 calibration process. Possible states include: Idle, Calibrating, Calibration Failed, and Calibration ok
3	The "Config Apply" button remains disabled until a change is made within the Configuration tab. Once any configuration element is modified (e.g. adjusting Reference weight), the button becomes active, prompting the user to apply the changes
4	Do C2 calibration. C2 calibration becomes available after configuration is applied successfully. Users can initiate the C2 calibration process once the configuration is complete.
5	The following steps are provided for user reference

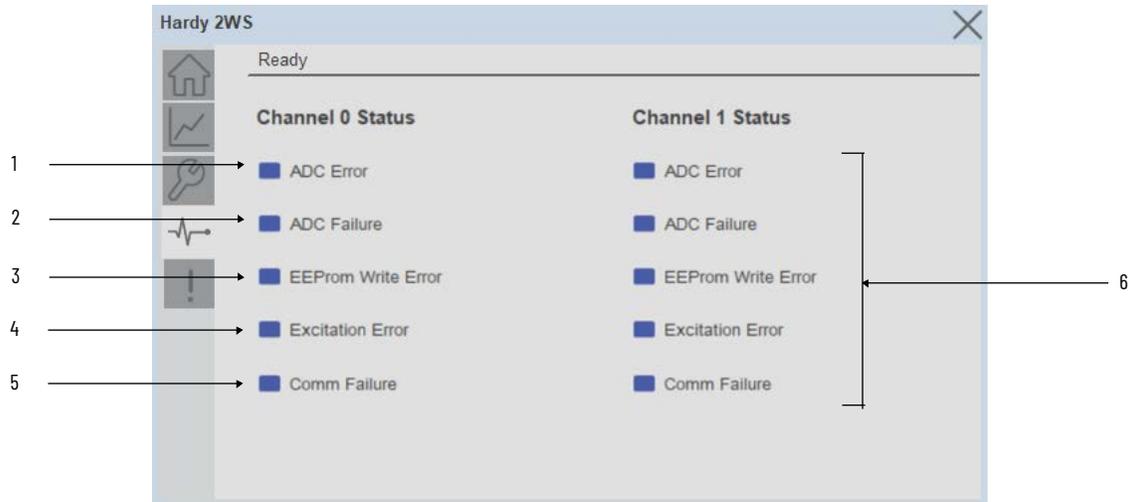
Traditional Calibration



Item	Description
1	CalLow Weight :- CalLow calibration carries out a calibration process for the low calibration system. Taking the CalLow Weight value as a starting point, it progresses from zero to the maximum span weight.
2	Span Weight :- Span weight calibration occurs after CalLow calibration. It assigns a weight to the cell process for the upcoming high calibration. Starting from the minimum CalLow value, the calibration process advances in steps determined by the span weight
3	Cal Status: Idle: System is ready for calibration. Calibrating: Calibration process is in progress. Calibration Failed: Calibration process encountered an error. Calibration Ok: Calibration completed successfully. Do High Calibration : High precision calibration is needed. High Calibration in Progress: High precision calibration is in progress. High Calibration Failed: High precision calibration encountered an error. High Calibration ok: High precision calibration completed successfully.
4	Do Low cal :- Do Low Calibration button will appear after configuration is applied. This button initiates the low calibration process.
5	The "Do High Calibration" button becomes available after the configuration is applied and the low calibration is complete. This button initiates the high calibration process.
6	The following steps are provided for user reference

Diagnostics Tab

The device diagnostics tab includes a list of information available in the device for troubleshooting. Diagnostics tab includes Device status and Failure Reason.

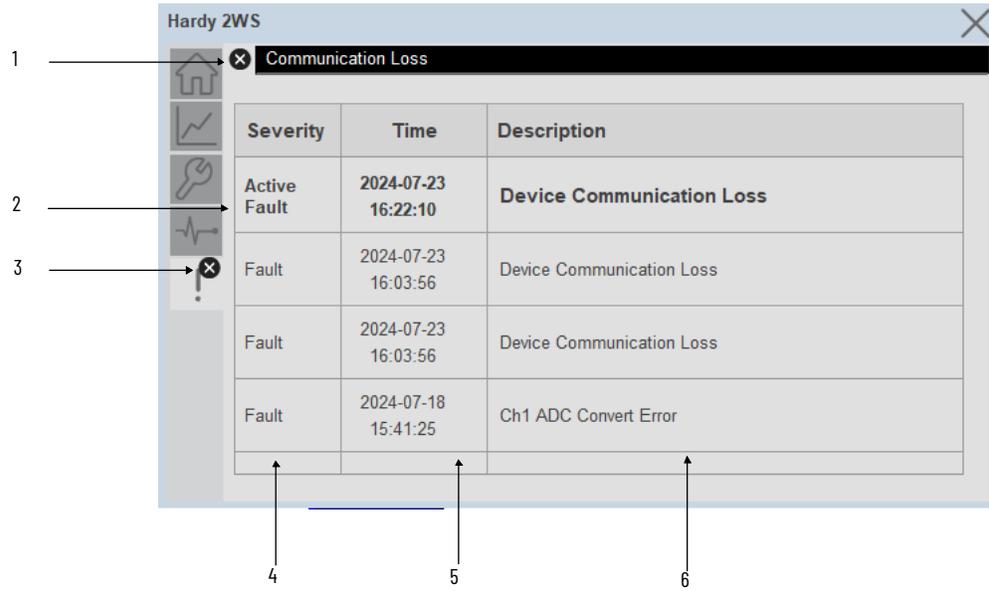


Item	Description
1	ADC Error: Load cell input out of range
2	ADC Failure: Output from the A/D converter to processor is bad. The module shows a solid red LED
3	EEProm Write Error: Module cannot write (save settings) to non-volatile memory. EEPROM is probably bad.
4	Excitation Error:-Continuously monitors a system's excitation current to check for open or shorted load sensors or damaged or broken excitation wire(s), including the wires
5	Comm Failure
6	Channel1 is the same as the content of channel0

Fault Warning Tab

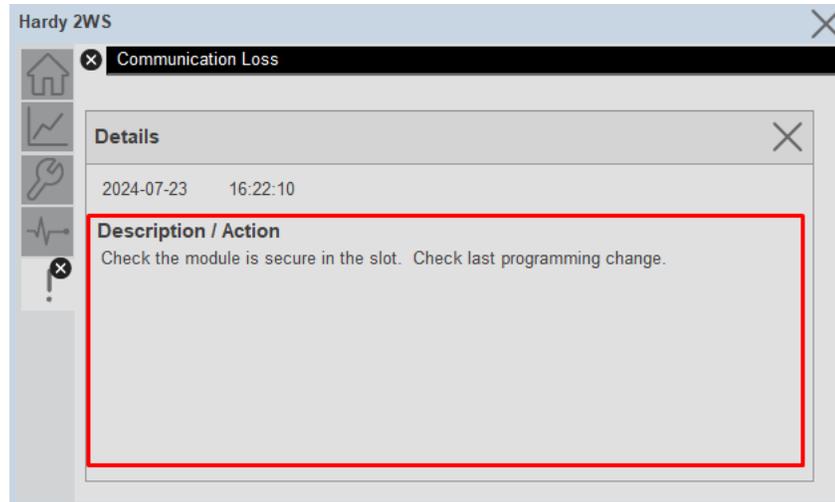
The Fault Warning tab displays information for up to four faults for the device. The fault table displays the Severity level (Fault, Warning or Active Fault), time (and date) and a description of the fault.

Note, only row 1 will display the “Active Fault” in the severity column if there is a current active fault, else it will display the last fault. Rows 2-4 only display past faults and warnings, not an active fault.



Item	Description
1	Banner
2	Last fault is in first row and show in bold if active
3	Fault tab icon visible when a fault is active
4	Fault severity
5	Fault event time
6	4 most recent fault/warning event messages

Click on any row in the fault table to view fault details. The details window provides a more detailed description and possible action steps to remedy condition.



Application Code Manager

All Hardy device objects have similar configuration parameters in Application Code Manager. The following section defines the common parameters.

Refer to the section [Using Application Code Manager](#) for complete details.

Definition Object: raC_Dvc_HI_1756xWS

This object contains the AOI definition and used as linked library to implement object. This gives flexibility to choose to instantiate only definition and create custom implement code. User may also create their own implement library and link with this definition library object.

Implementation Object: raC_LD_Dvc_HI_1756xWS

Parameter Name	Default Value	Instance Name	Definition	Description
RoutineName	{ObjectName}	{RoutineName}	Routine	Enter Routine name. Routine will be created and Object implement rung(s) inserted. A JSR will be inserted in MainRoutine. If routine name already exists, then object will be inserted into existing routine. By default, parameter is set to Object Name.
TagName	{ObjectName}	{TagName}	Backing Tag	Enter the backing tag of the main AOI. This will serve as the base tag name for other tags in this object that are derived from the base.
TagDescription	{ObjectDescription}	{TagDescription}		Tag Description of the main AOI backing tag
ModuleName	Mod_{ObjectName}	{ModuleName}	Input Parameter	Enter the Module Name. This is the name for the module that appears in the Controller Organizer tree.

Parameter Name	Default Value	Instance Name	Definition	Description
IncludeHW	1			Allow ACM to create the Hardware Module. If the module already exists in the Controller Organizer, select False or existing module properties will be overwritten.
Slot	0		Input Parameter	Select the channel slot number
NumberOfChannel	One Channel		Input Parameter	Select FieldBus Format selected on device
RPI	10.0		Input Parameter	This is the Requested Packet Interval (RPI) of the module (1.0ms - 100ms).
ParentModule	Local		Input Parameter	Select the Parent Module. This represents the name of the communication adapter this module will communicate through. If connecting to a non-library object module, enter the name of the module only. If the module is connected directly to the controller ethernet port, enter "Local". Note: entering non-library object modules will result in the parameter displaying a red X. This will still generate properly as long as the entered name exists in the project.
ChassisName	{ParentModule}			Warning removal

Linked Libraries

Link Name	Catalog Number	Revision	Solution	Category
raC_Dvc_HI_1756xWS	raC_Dvc_HI_1756xWS	1.0	(RA-LIB) Device	Hardy

Configured HMI Content

HMI Content	Instance Name	Description
Launch Button SE	{ObjectName}_GO_LaunchFP	Global Object configured callout instance
Launch Button ME	{ObjectName}_GO_LaunchFP	Global Object configured callout instance

Attachments

Name	Description	File Name	Extraction Path
V1_raC_Dvc_Global	Graphic Symbols SE	{raC-1-SE} Graphic Symbols - Hardy Device.ggfx	{ProjectName}\Visualization\FTViewSE\GlobalObjects
V1_raC_Dvc_Global	Graphic Symbols ME	{raC-1-ME} Graphic Symbols - Hardy Device.ggfx	{ProjectName}\Visualization\FTViewME\GlobalObjects
V1_raC_Dvc_Global	Toolbox SE	{raC-1-SE} Toolbox - Hardy Device.ggfx	{ProjectName}\Visualization\FTViewSE\GlobalObjects

VI_raC_Dvc_Global	Toolbox ME	(raC-1-ME) Toolbox - Hardy Device.ggfx	{ProjectName}\Visualization\FTViewME\GlobalObjects
VI_raC_Dvc_HI_1756xWS	Faceplate SE	(raC-1_00-SE) raC_Dvc_HI_1756xWS-Faceplate.gfx	{ProjectName}\Visualization\FTViewSE\Displays
VI_raC_Dvc_HI_1756xWS	Faceplate ME	(raC-1_00-ME) raC_Dvc_HI_1756xWS-Faceplate.gfx	{ProjectName}\Visualization\FTViewME\Displays
VI_raC_Dvc_Hardy	View Designer	(raC-1_00-VD) raC_Dvc_Hardy.vpd	{ProjectName}\Visualization\ViewDesigner
VI_Hardy_Manual	Reference Manual	DEVICE-RM915A-EN-P.pdf	{ProjectName}\Documentation
VI_Hardy_Images	HMI Image Set	Hardy_Images.zip	{ProjectName}\Visualization\Images
VI_IO_Device_HMI_Tag	HMI Tag	FTViewStudio_HardyLibrary_Tags_1.xx.CSV	{ProjectName}\Visualization\

HI5069 - CompactLogix® Weigh Scale Module (raC_Dvc_HI_5069xWS)

Overview

The Hardy device object includes a faceplate which displays status and configuration information of HI5069-CompactLogix® Weigh Scale Module series (raC_Dvc_HI_5069xWS). Hardy HI5069 Plug-In-Modules are high performance single or dual channel weigh scale modules that feature a powerful 24-bit sigmadelta (Σ - Δ) analog-to-digital converter (ADC), that when combined with Hardy's WAVERSAVER® filtering technology ensure accurate, fast, and stable weight data in even the most adverse conditions where noise and mechanical vibrations can plague process control. HI5069-WS Weigh Scale Modules are self-contained, microprocessorbased I/O modules that produce weight data when connected to strain gauge load sensors (load cells, load points, platform scales); and are plugged directly into the backplane of Allen-Bradley Compact5000 I/O Systems. The HI5069-WS and HI5069-2WS Weigh Scale Modules can be used for a wide variety of process weighing applications such as batching, blending, filling/dispensing, check weighing, force measurement, level by weight and weight rate monitoring. With Integrated Technician®, the module monitors and troubleshoots the weighing system by diagnosing individual sensors without the need to disconnect.

Functional Description

The HI5069-CompactLogix® Weigh Scale Module pre-configured Device Objects:

- Collect, Process and Deliver Data between Smart Devices and Application Logic
- Provide Device Status & Diagnostics Faceplates for Machine Startup, Commissioning, Operations, and Maintenance
- Include Common HMI Faceplate Components, Device States, and Control Interfaces providing Application Development and Operation Efficiency

All these features provide quick feedback, shorten recovery time, and simplify implementation.

Required Files

Device Objects include Add-On Instructions (AOIs) and HMI faceplates. The revision number (e.g. 1.00) used in file names can change as new revisions are created. While using FactoryTalk® View ME/SE you must also import the tag import file FTViewStudio_HardyLibrary_Tags_1_01.CSV to open the Help file.

Controller Files

Add-On Instructions are reusable code objects that contain encapsulated logic that can streamline implementing your system. This lets you create your own instruction set for programming logic as a supplement to the instruction set provided natively in the CompactLogix® firmware. An Add-On Instruction is defined once in each controller project, and can be instantiated multiple times in your application code as needed.

The Add-On Instruction must be imported into the controller project to be used in the controller configuration. These can be imported as Add-On Instruction files, or as part of the Rung Import or Import Library Objects wizard.

All Add-On Instruction and Rung Import files can be found in the */Studio 5000 Logix Designer Files - L5X/* folder in the library.

Device/Item	Add-On Instruction	Rung Import
HI5069xWS	raC_Dvc_HI_5069xWS_1.01_A01.L5X	raC_Dvc_HI_5069_1WS_1.01_RUNG.L5X
HI5069x2WS		raC_Dvc_HI_5069_2WS_1.01_RUNG.L5X

FactoryTalk® View HMI Files

FactoryTalk® View ME or SE applications require importing the desired device faceplates in addition to all Global Object (ggfx) files and all images located in the */HMI FactoryTalk® View Images - png/* folder of the library. FactoryTalk® View ME files are stored in the */HMI - FactoryTalk® View ME/* library folder and FactoryTalk® View SE files are stored in the */HMI - FactoryTalk® View SE/* library folder.

Device/Item	Type	FactoryTalk® View ME Faceplate	FactoryTalk® View SE Faceplate
HI5069xWS	Display	(raC-1.01-ME) raC_Dvc_HI_5069xWS-Faceplate.gfx	(raC-1.01-SE) raC_Dvc_HI_5069xWS-Faceplate.gfx
Graphic Symbols	Global Object	(raC-1-ME) Graphic Symbols - Hardy Device.ggfx	(raC-1-SE) Graphic Symbols - Hardy Device.ggfx
Toolbox	Global Object	(raC-1-ME) Toolbox - Hardy Device.ggfx	(raC-1-SE) Toolbox - Hardy Device.ggfx

Studio 5000® Application Code Manager Files

Studio 5000® Application Code Manager (ACM) can be optionally used if it is installed. All devices can be easily registered in the ACM repositories by running the *setup.cmd* file located in the root folder of the library.

Individual HSL4 files are provided as an alternative to running the *setup.cmd* to allow users to manually register specific implementation objects. Each object has two files - an Asset Control file and a Device file. The Asset Control files include attachments of all required files for that object. The Device files are used to actually add that device into a Studio 5000® project and these reference the Asset Control files.

All Studio 5000® Application Code Manager files can be found in the / *ApplicationCodeManagerLibraries/* folder of the library. The files included are as follows:

Implementation Object	Asset Control File (.HSL4)	Device File (.HSL4)
5069xWS	(RA-LIB)_Device_Asset-Control_HardyProcessSolutions_raC_Dvc_HI_5069xWS_(1.1)	(RA-LIB)_Device_Device_HardyProcessSolutions_raC_LD_Dvc_HI_5069xWS_(1.1)

Device Definition

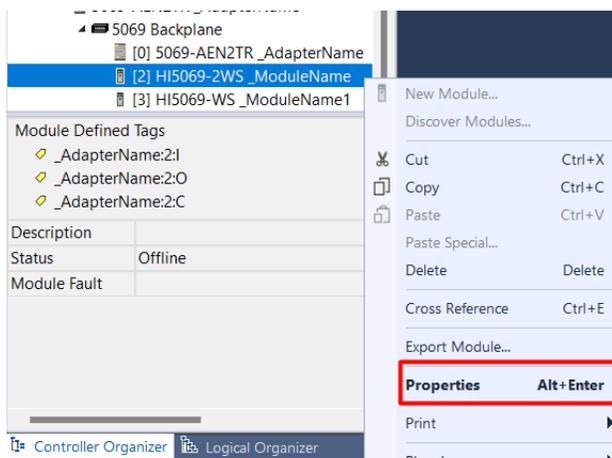
The device (ie: HI5069-CompactLogix® Weigh Scale Module) must be configured with the correct device definition. Proper device configuration enables the required cyclic device data to pass information from the device into the add-on instruction.



Note that this configuration is completed automatically when using Application Code Manager or the Studio 5000® Import Library Objects wizard plug-in.

To verify the device definition:

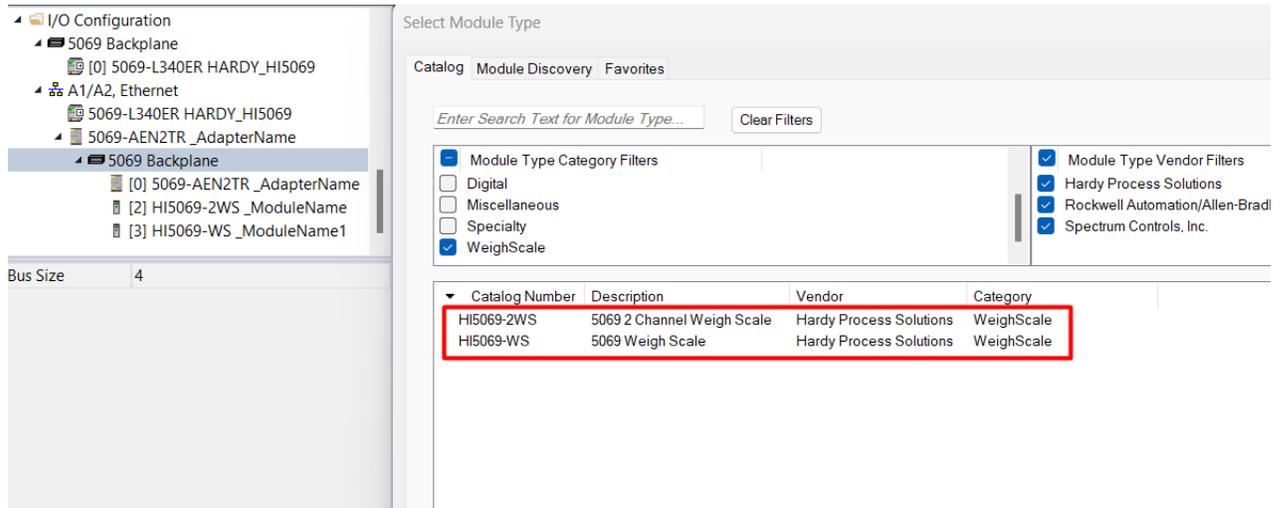
- Find the device in the *Controller Organizer* pane in Studio 5000 Logix Designer® and open the *Module Properties* by double-clicking or right-click and select *Properties*. There are two module types: one-channel and two-channel.



- Refer to the following sections for specific device configuration.

HARDWARE Definition

- Right click on the 5069 Backplane and click on New Module.
- Select "HI5069-WS" for single-channel use or "HI5069-2WS" for two-channel use.



Operations

The Hardy objects provide only physical mode of operation. There is no virtual device mode offered.

Faults & Warnings

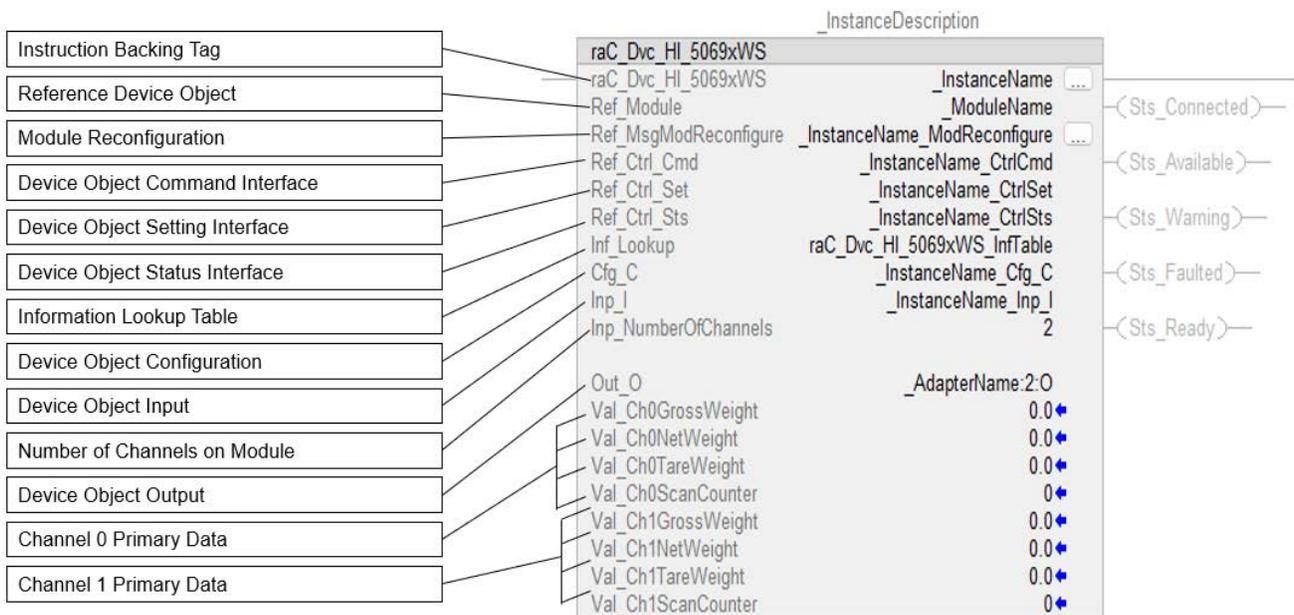
- **First Warning:** This function helps in capturing the first warning triggered in the device. Display the respective description in faceplate.
- **First Fault:** Capture the first fault from device. Display the respective description in faceplate.
- **Event log:** Log Warning and Fault the last 4 events in a log queue. The queue contains fault code, description, and time stamp. Display the same in faceplate.

Execution

The following table explains the handling of instruction execution conditions.

Condition	Description
EnableIn False (false rung)	Processing for EnableIn False (false rung) is handled the same as if the device were taken out of service by Command. The device outputs are de-energized and the device is shown as Program Out of Service on the HMI. All alarms are cleared.
Powerup (prescan, first scan)	On prescan, any commands that are received before first scan are discarded. The device is de-energized. On first scan, the device is treated as if it were returning from Hand command source: the instruction state is set based on the position feedback that is received from the device. If the feedback is valid for one position, the device is set to that position. If the device does not have position feedback or the position feedback is invalid, the device is set to the 'unknown/powerup' state. The command source is set to its default, either Operator or Program (unlocked).
Postscan	No SFC Postscan logic is provided.

Add-On Instruction I/O Data Add-On Instruction Ladder Implementation



InOut Data

InOut	Function / Description	Data Type
Ref_Module	Reference to module in I/O tree	MODULE
Ref_MsgModReconfigure	Message Module Reconfiguration Write	MESSAGE
Ref_Ctrl_Cmd	Hardy Device Command Interface	raC_UDT_ItfAD_Hardy_CtrlCmd
Ref_Ctrl_Set	Hardy Device Setting Interface	raC_UDT_ItfAD_Hardy_CtrlSet
Ref_Ctrl_Sts	Hardy Device Setting Interface	raC_UDT_ItfAD_Hardy_CtrlSts
Inf_Lookup	Code / Description List Entry	raC_UDT_LookupMember_STR0082[3 1]
Cfg_C	Device Object Inputs	raC_UDT_HI_5069xWS_Cfg
Inp_I	Device Object Inputs	HI:5069_xWS:I:0
Inp_NumberOfChannels	Number of Channels for Device Identifier: 1 = WS; 2 = 2WS	INT
Out_O	Device Object Output	HI:5069_xWS:O:0

Input Data

Input	Function/Description	Data Type
Cfg_Ch0Units	Channel0 Units : 0=oz, 1=lb, 2=ton, 3=g, 4=kg, 5=t	DINT
Cfg_Ch0WaverSaver	Channel0 WaveSaver : 0=Off, 1=7.5 Hz, 2=3.5 Hz, 3=1 Hz, 4=0.5 Hz, 5=0.25 Hz	DINT
Cfg_Ch0ZeroTrkEn	Channel0 Zero Trk En : 0=NotEnable, 1=Enable	DINT
Cfg_Ch0LoadCellSensitivity	Channel0 Load Cell Sensitivity : 0=1.0mV/V, 1=1.5mV/V, 2=2.0mV/V, 3=2.5 mV/V, 4=3.0mV/V, 5=3.5 mV/V, 6=4.0mV/V, 7=4.5 mV/V, 8=5.0mV/V	DINT
Cfg_Ch0NoOfSensorsJB1	Channel0 Number Of Sensors on J-Box1: 0=1, 1=2, 2=3, 3=4	DINT
Cfg_Ch0NoOfSensorsJB2	Channel0 Number Of Sensors on J-Box2: 0=1, 1=2, 2=3, 3=4	DINT
Cfg_Ch1Units	Channel1 Units : 0=oz, 1=lb, 2=ton, 3=g, 4=kg, 5=t	DINT
Cfg_Ch1WaverSaver	Channel1 WaveSaver : 0=Off, 1=7.5 Hz, 2=3.5 Hz, 3=1 Hz, 4=0.5 Hz, 5=0.25 Hz	DINT
Cfg_Ch1ZeroTrkEn	Channel1 Zero Trk En : 0=NotEnable, 1=Enable	DINT
Cfg_Ch1LoadCellSensitivity	Channel1 Load Cell Sensitivity : 0=1.0mV/V, 1=1.5mV/V, 2=2.0mV/V, 3=2.5 mV/V, 4=3.0mV/V, 5=3.5 mV/V, 6=4.0mV/V, 7=4.5 mV/V, 8=5.0mV/V	DINT
Cfg_Ch1NoOfSensorsJB1	Channel1 Number Of Sensors on J-Box1: 0=1, 1=2, 2=3, 3=4	DINT
Cfg_Ch1NoOfSensorsJB2	Channel1 Number Of Sensors on J-Box2: 0=1, 1=2, 2=3, 3=4	DINT
Cmd_ApplyConfiguration	Module reconfigure Command	BOOL
Cmd_Ch0ITReduced	Ch0 IT Reduced Start Command: 0 = Start, 1 = Stop	BOOL
Cmd_Ch1ITReduced	Ch1 IT Reduced Start Command: 0 = Start, 1 = Stop	BOOL
EnableIn	Enable Input - System Defined Parameter	BOOL
Set_Ch0AutoZeroTol	Channel0 Setpoint of Auto Zero Tolerance	REAL
Set_Ch0RefWt	Channel0 Setpoint of Reference Weight	REAL
Set_Ch0DiscreteCmds	Channel0 Commands : 0=Zero cmd, 1=Tare cmd, 5= Cal Low cmd, 6=Cal High cmd, 7=C2 Cal cmd, 3 = Start IT Test cmd	INT
Set_Ch0MotionTol	Channel0 Setpoint of Motion Tolerance	REAL

Input	Function/Description	Data Type
Set_Ch0NumAvg	Channel0 Setpoint of Num Averages	DINT
Set_Ch0SpanWt	Channel0 Setpoint of Span Weight	REAL
Set_Ch0TareWeight	Channel0 Setpoint of Tare Weight	REAL
Set_Ch0ZeroTol	Channel0 Setpoint of Zero Tolerance	REAL
Set_Ch0GravityCorrection	Channel0 Setpoint of Gravity Correction	REAL
Set_Ch1AutoZeroTol	Channel1 Setpoint of Auto Zero Tolerance	REAL
Set_Ch1RefWt	Channel1 Setpoint of Reference Weight	REAL
Set_Ch1DiscreteCmds	Channel1 Commands : 0=Zero cmd, 1=Tare cmd, 5= Cal Low cmd, 6=Cal High cmd, 7=C2 Cal cmd, 3 = Start IT Test cmd	INT
Set_Ch1MotionTol	Channel1 Setpoint of Motion Tolerance	REAL
Set_Ch1NumAvg	Channel1 Setpoint of Num Averages	DINT
Set_Ch1SpanWt	Channel1 Setpoint of Span Weight	REAL
Set_Ch1TareWeight	Channel1 Setpoint of Tare Weight	REAL
Set_Ch1ZeroTol	Channel1 Setpoint of Zero Tolerance	REAL
Set_Ch1GravityCorrection	Channel1 Setpoint of Gravity Correction	REAL

Output Data

Output	Function/Description	Data Type
Sts_Active	Device active status: 1 = output power structure is active	BOOL
Sts_ApplyConfiguration	Module Reconfigure Request On Status	BOOL
Sts_Available	Device is available for interaction with user code	BOOL
Sts_bNotReady	Bitwise device 'not ready' reason 0 = Reserved 1 = Device not connected 2 = Device not available 3 = Device Faulted 4 - 31 = Reserved	DINT
Sts_Ch0ADCCovertError	Channel0 ADC Covert Error	BOOL
Sts_Ch0ADCErr	Channel0 ADC Error	BOOL
Sts_Ch0ADCFailure	Channel0 ADC Failure	BOOL
Sts_Ch0CommFailure	Channel0 Comm Failure	BOOL
Sts_Ch0EEPROMWriteError	Channel0 EEPROM write Error	BOOL
Sts_Ch0FaultActive	Channel0 Fault Active	BOOL
Sts_Ch0Motion	Channel0 Weight is unstable	BOOL
Sts_Ch1ADCCovertError	Channel1 ADC Covert Error	BOOL
Sts_Ch1ADCErr	Channel1 ADC Error	BOOL
Sts_Ch1ADCFailure	Channel1 ADC Failure	BOOL
Sts_Ch1CommFailure	Channel1 Comm Failure	BOOL
Sts_Ch1EEPROMWriteError	Channel1 EEPROM write Error	BOOL
Sts_Ch1FaultActive	Channel1 Fault Active	BOOL
Sts_Ch1Motion	Channel1 Motion	BOOL
Sts_CommunicationOk	Communication Between Controller & 5069xWS working OK	BOOL
Sts_Connected	Device is connected to the Programmable Automation Controller	BOOL
Sts_Failure	Application fault; predictive diagnostics alarm triggered or command cannot be executed as requested	BOOL
Sts_Faulted	Device faulted status: 1 = an active fault exists	BOOL
Sts_InhibitCfg	Disable Configuration inputs from external sources	BOOL
Sts_InhibitCmd	Disable Command inputs from external sources	BOOL
Sts_InhibitSet	Disable Setting inputs from external sources	BOOL
Sts_Ready	Device is ready to perform primary function	BOOL
Sts_Warning	Device warning status: 1 = an active alarm or warning exists	BOOL
Val_Ch0AutoZeroTol	Channel0 Auto Zero Tolerance	REAL
Val_Ch0CalStNew	Channel0 Cal Status: 0= Success, 1=Fail, 2=Fail - ADC error, 20=Ready, 4=Fail - Motion, 5=Fail - No C2 Load Cells Found, 6=Fail - C2 Capacities Not Equal, 7=Fail - Non-Hardy C2 Load Cells, 255=Command in Progress	DINT

Output	Function/Description	Data Type
Val.Ch0CmdStatus	Channel0 Command Status: 0 = Success, 1 = Fail, 2 = Fail - ADC error, 3 = Fail - Out Of Tolerance, 4 = Fail - Motion, 5 = Fail - No C2 Load Cells Found, 6 = Fail - C2 Capacities Not Equal, 7 = Fail - Non-Hardy C2 Load Cells, 8 = Fail - Not Enough Counts Between Cal Low And Cal High Weights, 11 = Fail - Value Too High, 12 = Fail - Value Too Low, 13 = Fail - Not Allowed, 128 = Fail - Parameter ID Not Found; 20 = Ready, 255 = Command In Progress	DINT
Val.Ch0GravityCorrection	Channel0 Gravity Correction	REAL
Val.Ch0GrossWeight	Channel0 Gross Weight	REAL
Val.Ch0InstStatus	Channel0 Status	DINT
Val.Ch0ITCmdStatus	Channel0 IT Command Status: 0 = Success, 1 = Fail, 20 = Ready, 255 = Command In Progress	DINT
Val.Ch0LoadCellSensitivity	Channel0 Load Cell Sensitivity : 0=1.0mV/V, 1=1.5mV/V, 2=2.0mV/V, 3=2.5 mV/V, 4=3.0mV/V, 5=3.5 mV/V, 6=4.0mV/V, 7=4.5 mV/V, 8=5.0mV/V	DINT
Val.Ch0MotionTol	Channel0 Motion Tolerance	REAL
Val.Ch0MVSensor1JB1	Channel0 IT MV Sensor1 JB1	REAL
Val.Ch0MVSensor1JB2	Channel0 IT MV Sensor1 JB2	REAL
Val.Ch0MVSensor2JB1	Channel0 IT MV Sensor2 JB1	REAL
Val.Ch0MVSensor2JB2	Channel0 IT MV Sensor2 JB2	REAL
Val.Ch0MVSensor3JB1	Channel0 IT MV Sensor3 JB1	REAL
Val.Ch0MVSensor3JB2	Channel0 IT MV Sensor3 JB2	REAL
Val.Ch0MVSensor4JB1	Channel0 IT MV Sensor4 JB1	REAL
Val.Ch0MVSensor4JB2	Channel0 IT MV Sensor4 JB2	REAL
Val.Ch0NetWeight	Channel0 Net Weight	REAL
Val.Ch0NoOfSensorsJB1	Channel0 Number Of Sensors on J-Box1: 0=1, 1=2, 2=3, 3=4	DINT
Val.Ch0NoOfSensorsJB2	Channel0 Number Of Sensors on J-Box2: 0=1, 1=2, 2=3, 3=4	DINT
Val.Ch0NumAvg	Channel0 Num Averages	DINT
Val.Ch0ITRedStatus	Channel0 IT Reduced Command Status	DINT
Val.Ch0RefWt	Channel0 Reference Weight	REAL
Val.Ch0ScanCounter	Channel0 Scan Counter	SINT
Val.Ch0SpanWt	Channel0 Span Weight	REAL
Val.Ch0TareWeight	Channel0 Tare Weight	REAL
Val.Ch0TraditionCalStNew	Channel0 Traditional Calibration	DINT
Val.Ch0Units	Channel0 Units: 0=oz, 1=lb, 2=ton, 3=g, 4=kg, 5=t	DINT
Val.Ch0WaverSaver	Channel0 WaveSaver : 0=Off, 1=7.5 Hz, 2=3.5 Hz, 3=1 Hz, 4=0.5 Hz, 5=0.25 Hz	DINT
Val.Ch0WeightSensor1JB1	Channel0 IT Weight Sensor1 JB1	REAL
Val.Ch0WeightSensor1JB2	Channel0 IT Weight Sensor1 JB2	REAL

Output	Function/Description	Data Type
Val.Ch0WeightSensor2JB1	Channel0 IT Weight Sensor2 JB1	REAL
Val.Ch0WeightSensor2JB2	Channel0 IT Weight Sensor2 JB2	REAL
Val.Ch0WeightSensor3JB1	Channel0 IT Weight Sensor3 JB1	REAL
Val.Ch0WeightSensor3JB2	Channel0 IT Weight Sensor3 JB2	REAL
Val.Ch0WeightSensor4JB1	Channel0 IT Weight Sensor4 JB1	REAL
Val.Ch0WeightSensor4JB2	Channel0 IT Weight Sensor4 JB2	REAL
Val.Ch0ZeroTol	Channel0 Zero Tolerance	REAL
Val.Ch0ZeroTrkEn	Channel0 Zero Tracking Enable : 0=NotEnable, 1=Enable	DINT
Val.Ch1AutoZeroTol	Channel1 Auto Zero Tolerance	REAL
Val.Ch1CalStNew	Channel1 Cal Status: 0= Success, 1=Fail, 2=Fail - ADC error, 20=Ready, 4=Fail - Motion, 5=Fail - No C2 Load Cells Found, 6=Fail - C2 Capacities Not Equal, 7=Fail - Non-Hardy C2 Load Cells, 255=Command in Progress	DINT
Val.Ch1CmdStatus	Channel1 Command Status: 0 = Success, 1 = Fail, 2 = Fail - ADC error, 3 = Fail - Out Of Tolerance, 4 = Fail - Motion, 5 = Fail - No C2 Load Cells Found, 6 = Fail - C2 Capacities Not Equal, 7 = Fail - Non-Hardy C2 Load Cells, 8 = Fail - Not Enough Counts Between Cal Low And Cal High Weights, 11 = Fail - Value Too High, 12 = Fail - Value Too Low, 13 = Fail - Not Allowed, 128 = Fail - Parameter ID Not Found; 20 = Ready, 255 = Command In Progress	DINT
Val.Ch1GravityCorrection	Channel1 Gravity Correction	REAL
Val.Ch1GrossWeight	Channel1 Gross Weight	REAL
Val.Ch1InstStatus	Channel1 Status	DINT
Val.Ch1ITCmdStatus	Channel1 IT Command Status: 0 = Success, 1 = Fail, 20 = Ready, 255 = Command In Progress	DINT
Val.Ch1LoadCellSensitivity	Channel1 Load Cell Sensitivity : 0=1.0mV/V, 1=1.5mV/V, 2=2.0mV/V, 3=2.5 mV/V, 4=3.0mV/V, 5=3.5 mV/V, 6=4.0mV/V, 7=4.5 mV/V, 8=5.0mV/V	DINT
Val.Ch1MotionTol	Channel1 Motion Tolerance	REAL
Val.Ch1MVSensor1JB1	Channel1 IT MV Sensor1 JB1	REAL
Val.Ch1MVSensor1JB2	Channel1 IT MV Sensor1 JB2	REAL
Val.Ch1MVSensor2JB1	Channel1 IT MV Sensor2 JB1	REAL
Val.Ch1MVSensor2JB2	Channel1 IT MV Sensor2 JB2	REAL
Val.Ch1MVSensor3JB1	Channel1 IT MV Sensor3 JB1	REAL
Val.Ch1MVSensor3JB2	Channel1 IT MV Sensor3 JB2	REAL
Val.Ch1MVSensor4JB1	Channel1 IT MV Sensor4 JB1	REAL
Val.Ch1MVSensor4JB2	Channel1 IT MV Sensor4 JB2	REAL
Val.Ch1NetWeight	Channel1 Net Weight	REAL
Val.Ch1NoOfSensorsJB1	Channel1 Number Of Sensors on J-Box1: 0=1, 1=2, 2=3, 3=4	DINT
Val.Ch1NoOfSensorsJB2	Channel1 Number Of Sensors on J-Box2: 0=1, 1=2, 2=3, 3=4	DINT

Output	Function/Description	Data Type
Val.Ch1NumAvg	Channel1 Num Averages	DINT
Val.Ch1ITRedStatus	Channel1 IT Reduced Command Status	DINT
Val.Ch1RefWt	Channel1 Reference Weight	REAL
Val.Ch1ScanCounter	Channel1 Scan Counter	SINT
Val.Ch1SpanWt	Channel1 Span Weight	REAL
Val.Ch1TareWeight	Channel1 Tare Weight	REAL
Val.Ch1TraditionCalStNew	Channel1 Traditional Calibration	DINT
Val.Ch1Units	Channel1 Units: 0=oz, 1=lb, 2=ton, 3=g, 4=kg, 5=t	DINT
Val.Ch1WaverSaver	Channel1 WaverSaver : 0=Off, 1=7.5 Hz, 2=3.5 Hz, 3=1 Hz, 4=0.5 Hz, 5=0.25 Hz	DINT
Val.Ch1WeightSensor1JB1	Channel1 IT Weight Sensor1 JB1	REAL
Val.Ch1WeightSensor1JB2	Channel1 IT Weight Sensor1 JB2	REAL
Val.Ch1WeightSensor2JB1	Channel1 IT Weight Sensor2 JB1	REAL
Val.Ch1WeightSensor2JB2	Channel1 IT Weight Sensor2 JB2	REAL
Val.Ch1WeightSensor3JB1	Channel1 IT Weight Sensor3 JB1	REAL
Val.Ch1WeightSensor3JB2	Channel1 IT Weight Sensor3 JB2	REAL
Val.Ch1WeightSensor4JB1	Channel1 IT Weight Sensor4 JB1	REAL
Val.Ch1WeightSensor4JB2	Channel1 IT Weight Sensor4 JB2	REAL
Val.Ch1ZeroTol	Channel1 Zero Tolerance	REAL
Val.Ch1ZeroTrkEn	Channel1 Zero Tracking Enable: 0=NotEnable, 1=Enable	DINT

Data Types

The following Hardy Common Control Interface tags are the primary device program tags to read and write to when interfacing to Hardy devices. The value of using these tags in your specific application code is that you may use a number of different Hardy devices such as 1756xWS without having to update your application device interface tags.

Refer to the [Interfaces](#) section for detailed information on interfaces.

raC_UDT_ItfAD_Hardy_CtrlSet

This is the Hardy Common Control Interface User-Defined Data Type for device settings. Its members provide application program access to allow or inhibit commands and settings from the device faceplate or other external sources. The table below shows member names, descriptions, and tag data types.

For example, to inhibit write commands from the device faceplate or other external sources write a 1 to the `_InstanceName_CtrlSet.InhibitCmd` program tag from your application program. This would prevent a Clear Tare command from the device faceplate. You may also set the Pre-Tare Value for the device.

Member	Description	Data Type
InhibitCmd	1 = Inhibit user Commands from external sources, 0 = Allow.	BOOL
InhibitSet	1 = Inhibit user Settings from external sources, 0 = Allow.	BOOL
InhibitCfg	1 = Inhibit user Configuration from external sources, 0 = Allow.	BOOL
Ch0TareValue	Channel0 Setpoint of Tare Value	REAL
Ch1TareValue	Channel1 Setpoint of Tare Value	REAL
Ch0Units	Channel0 Units 0=oz 1=lb 2=ton 3=g 4=kg 5=t	INT
Ch0Units	Channel1 Units 0=oz 1=lb 2=ton 3=g 4=kg 5=t	INT
ReserveSetDINT1	ReserveSetDINT1	DINT
ReserveSetDINT2	ReserveSetDINT2	DINT
ReserveSetREAL1	ReserveSetREAL1	REAL
ReserveSetREAL2	ReserveSetREAL2	REAL

raC_UDT_ItfAD_Hardy_CtrlCmd

This is the Hardy Common Control Interface User-Defined Data Type for device commands. Its members provide application program access to common device commands.

Only write to these common command members to control the device. If you write directly to the device’s output command tags directly unexpected device operation could occur.

For example, to tare the weight write a 1 to the `_InstanceName_CtrlCmd.TareImmediate`. Although, you can write to the uncommon command tags in the device’s output tag if a specific common control interface tag does not exist.

The table below shows member names, descriptions, and tag data types.

Member	Description	Data Type
bCmd	Commands (Bit Overlay).	INT
ResetWarn	1 = Reset device warning [No warning reset].	BOOL
ResetFault	1 = Reset device trip or fault [No Fault reset,- Automatic fault reset only].	BOOL
Physical	1 = Operate as Physical Device - hold for future use.	BOOL
Virtual	Virtual mode not implemented - hold for future use.	BOOL
Ch0Tare	1 = Trigger execution of Tare Command	BOOL
Ch0Zero	1 = Trigger execution of Zero Command	BOOL
Ch0C2Cal	1 = Trigger Ch0 C2 Calibration	BOOL
Ch0LoCal	1 = Trigger Ch0 Low Cal	BOOL
Ch0HiCal	1 = Trigger Ch0 High Cal	BOOL
Ch1Tare	1 = Trigger execution of Tare Command	BOOL
Ch1Zero	1 = Trigger execution of Zero Command	BOOL
Ch1C2Cal	1 = Trigger Ch1 C2 Calibration	BOOL
Ch1LoCal	1 = Trigger Ch1 Low Cal	BOOL

Member	Description	Data Type
Ch1HiCal	1 = Trigger Ch1 High Cal	BOOL
Reserve1	Reserved 1	BOOL
Reserve2	Reserved 2	BOOL
Reserve3	Reserved 3	BOOL
Reserve4	Reserved 4	BOOL

raC_UDT_ItfAD_Hardy_CtrlSts

This is the Hardy Common Control Interface User-Defined Data Type for device status. Its members provide application program access to device states, status, and diagnostic data. The table below shows member names, descriptions, and tag data types.

Input	Description	Data Type
eState	Enumerated state value: 0 = Unused, 1 = Initializing, 2 = Disconnected, 3 = Disconnecting, 4 = Connecting, 5 = Idle, 6 = Configuring, 7 = Available.	DINT
FirstWarning	First Warning Event Data.	raC_UDT_Event
FirstFault	First Fault Event Data.	raC_UDT_Event
eCmdFail	Enumerated command failure code. See extended help for enumeration values.	DINT
bSts	Status (Bit Overlay). 0 = Connected, 1 = Available, 2 = Warning, 3 = Faulted, 4 = Ready, 5 = Active.	DINT
Connected	1 = PAC to device connection has been established.	BOOL
Available	1 = The device is available for interaction with the user program.	BOOL
Warning	1 = A warning is active on the device.	BOOL
Faulted	1 = A fault is active on the device.	BOOL
Physical	1 = Controlling physical device.	BOOL
Virtual	1 = Controlling virtual device.	BOOL
Ch0GrossWeight	Channel0 Gross Weight	Real
Ch0NetWeight	Channel0 Net Weight	Real
Ch0TareWeight	Channel0 Tare Weight	Real
Ch0Units	Channel0 Units 0=oz 1=lb 2=ton 3=g 4=kg 5=t	INT
Ch1GrossWeight	Channel1 Gross Weight	Real
Ch1NetWeight	Channel1 Net Weight	Real
Ch1TareWeight	Channel1 Tare Weight	Real
Ch1Units	Channel1 Units 0=oz 1=lb 2=ton 3=g 4=kg 5=t	INT
ReserveSetDINT1	ReserveSetDINT1	DINT
ReserveSetDINT2	ReserveSetDINT2	DINT
ReserveSetREAL1	ReserveStatusREAL1	REAL
ReserveStatusREAL2	ReserveStatusREAL2	REAL

raC_UDT_Event

An array of size 4 is to be used to log the FirstWarning and FirstFault capture. The data should be FIFO order. The same should be displayed on the Faceplate.

Member	Description	Data Type
Type	Event type: 1 = Status, 2 = Warning, 3 = Fault, 4...n = User.	DINT
ID	User definable event ID.	DINT
Category	User definable category (Electrical, Mechanical, Materials, Utility, etc.).	DINT
Action	User definable event action code.	DINT
Value	User definable event value or fault code.	DINT
Message	Event message text.	STRING
EventTime_L	Timestamp (Date/Time format).	LINT
EventTime_D	Timestamp (Y,M,D,h,m,s,us).	DINT[7]

raC_UDT_HI_5069xWS_Cfg

The Hardy configuration Control Interface UDT serves as a structured way to represent and manage device status information within the Hardy system.

It encapsulates device-specific attributes and provides an interface for both reading (monitoring) and writing (controlling) these attributes.

	Member	Description	Data Type
Ch0		Channel0 Configuration Interface	HI:5069_WS_Channel:C:0
	GravityCorrection	Channel0 Gravity Correction	REAL
	MotionTolerance	Channel0 Motion Tolerance	REAL
	ZeroTolerance	Channel0 Zero Tolerance	REAL
	TareWeight	Channel0 Tare Weight	REAL
	RefWeight	Channel0 Reference Weight	REAL
	SpanWeight	Channel0 Span Weight	REAL
	AutoZeroTolerance	Channel0 Auto Zero Tolerance	REAL
	AutoZeroTrackEnabled	Channel0 Zero Tracking Enable: 0=NotEnable, 1=Enable	SINT
	Units	Channel0 Units: 0=oz, 1=lb, 2=ton, 3=g, 4=kg, 5=t	SINT
	LoadCellSensitivity	Channel0 Load Cell Sensitivity: 0=1.0mV/V, 1=1.5mV/V, 2=2.0mV/V, 3=2.5 mV/V, 4=3.0mV/V, 5=3.5 mV/V, 6=4.0mV/V, 7=4.5 mV/V, 8=5.0mV/V	SINT
	Waversaver	Channel0 WaveSaver: 0=Off, 1=7.5 Hz, 2=3.5 Hz, 3=1 Hz, 4=0.5 Hz, 5=0.25 Hz	SINT
	NumAverages	Channel0 Num Averages	INT

	Member	Description	Data Type
Ch1		Channel1 Configuration Interface	HI:5069_WS_Channel:C:0
	GravityCorrection	Channel1 Gravity Correction	REAL
	MotionTolerance	Channel1 Motion Tolerance	REAL
	ZeroTolerance	Channel1 Zero Tolerance	REAL
	TareWeight	Channel1 Tare Weight	REAL
	RefWeight	Channel Reference Weight	REAL
	SpanWeight	Channel1 Span Weight	REAL
	AutoZeroTolerance	Channel1 Auto Zero Tolerance	REAL
	AutoZeroTrackEnabled	Channel1 Zero Tracking Enable: 0=NotEnable, 1=Enable	SINT
	Units	Channel1 Units: 0=oz, 1=lb, 2=ton, 3=g, 4=kg, 5=t	SINT
	LoadCellSensitivity	Channel1 Load Cell Sensitivity: 0=1.0mV/V, 1=1.5mV/V, 2=2.0mV/V, 3=2.5 mV/V, 4=3.0mV/V, 5=3.5 mV/V, 6=4.0mV/V, 7=4.5 mV/V, 8=5.0mV/V	SINT
	Waversaver	Channel1 WaveSaver: 0=Off, 1=7.5 Hz, 2=3.5 Hz, 3=1 Hz, 4=0.5 Hz, 5=0.25 Hz	SINT
NumAverages	Channel1 Num Averages	INT	

raC_UDT_HI_5069xWS_Inp

The Hardy configuration Control Interface UDT serves as a structured way to represent and manage device status information within the Hardy system.

It encapsulates device-specific attributes and provides an interface for both reading (monitoring) and writing (controlling) these attributes.

	Member	Description	Data Type
	RunMode	1 = Run Mode Active	BOOL
	ConnectionFaulted	1 = Device to Master Connection Faulted	BOOL
	FieldSidePwrFault	1 = Field Side Power Fault	BOOL
	DiagnosticActive	1 = Diagnostic Active	BOOL
	DiagnosticSequenceCount	Diagnostic Sequence Count Data	SINT
Cmd	ChannelNumber	0 = Channel 0, 1 = Channel 1	DINT
	CMD_Echo	Echo the command from Output Table, to ensure correct command is executed	INT
	CMD_Status	Returns the value of command passed in command register	INT
	ParameterValue	Provides the value for specified parameter ID	DINT
	ParameterID	Echo of the value sent in the output table	INT
	ParameterRD1	Read only values for the different user selectable parameter ID values set in the output table	DINT
	ParameterRD2		DINT

Member		Description	Data Type
Ch0	ADConvertError	Bad input from the load cells	BOOL
	ADFailure	No output from the converter to the processor	BOOL
	InMotion	Indicates weight is in motion (changing)	BOOL
	NVMWriteError	Problem writing to the non-volatile memory in the unit	BOOL
	CenterOfZero	Indicates the gross weight is reading at the calibration zero point	BOOL
	SavingToNVM	Saving to Non Volatile Memory	BOOL
	CalibrationInProgress	Calibration in Progress	BOOL
	ParamIDNotFound	The parameter ID is invalid	BOOL
	ScanCounter	Provides constantly changing value, which is used as confirmation of communications	SINT
	NetWeight	Provides the Net Weight Value	REAL
	GrossWeight	Provides the Gross Weight Value	REAL
Ch1	ADConvertError	Bad input from the load cells	BOOL
	ADFailure	No output from the converter to the processor	BOOL
	InMotion	Indicates weight is in motion (changing)	BOOL
	NVMWriteError	Problem writing to the non-volatile memory in the unit	BOOL
	CenterOfZero	Indicates the gross weight is reading at the calibration zero point	BOOL
	SavingToNVM	Saving to Non Volatile Memory	BOOL
	CalibrationInProgress	Calibration in Progress	BOOL
	ParamIDNotFound	The parameter ID is invalid	BOOL
	ScanCounter	Provides constantly changing value, which is used as confirmation of communications	SINT
	NetWeight	Provides the Net Weight Value	REAL
	GrossWeight	Provides the Gross Weight Value	REAL

raC_UDT_Dropdown

Member	Description	Data Type
Slider_Min	Slider Minimum	SINT
Slider_Max	Slider Maximum	SINT
Total_Item_Count	Total Length of Dropdown	SINT
List_Shift	Slider Value for Total Length of Dropdown	SINT
List_Select	Slider Value for Visible rows of Dropdown	SINT
Selected	Slider Value as per Total Count of Dropdown	SINT
Selected_Item	Selected Item From Dropdown	INT
Animation_Active	Dropdown List Visible	INT
Set_Up	Slider Up Command	BOOL
Set_Down	Slider Down Command	BOOL
Trigger_Tag	After Selection Trigger Bit	BOOL
List_Display	Dropdown List Item	STR0020[5]
List_Item	Enter Dropdown item names. e.g. Option0, Option1...etc	STR0020[16]

raC_UDT_LookupMember_STR0082

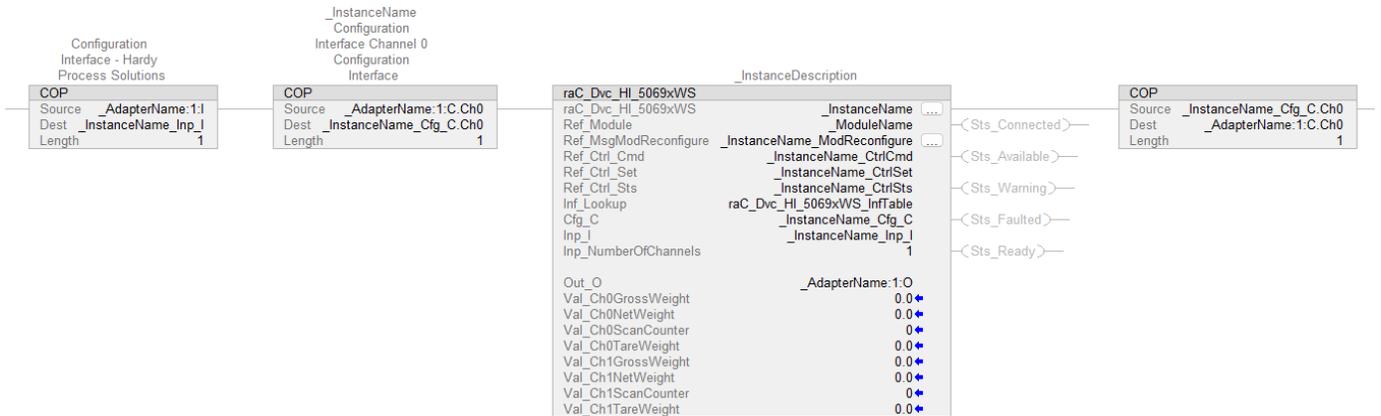
Member	Description	Data Type
Code	Code	DINT
Desc	Code Description	STRING

Programming Example

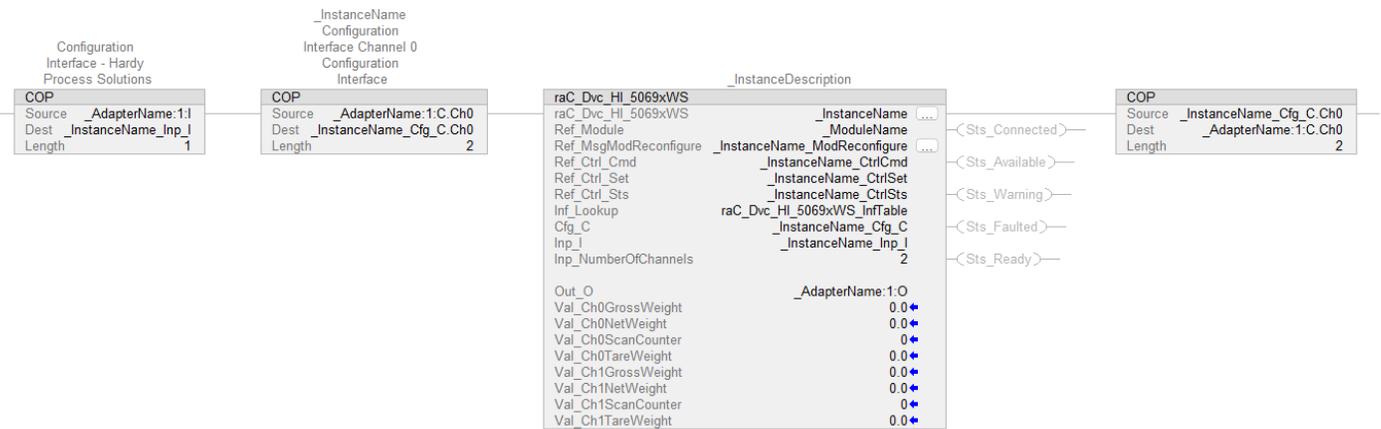
Fully configured device on a rung is provided below for reference. This example includes the device objects for a HI5069 - CompactLogix® Weigh Scale Module (raC_Dvc_HI_5069xWS).

Note that this programming example is the same code that is imported when either importing the supplied rung .L5X files or when using Application Code Manager or the Studio 5000® Import Library Objects wizard plug-in.

When you configure a one-channel import, be sure to import the supplied rung raC_Dvc_HI_5069_1WS_1.01_RUNG



When you configure a Two-channel import, be sure to import the supplied rung raC_Dvc_HI_5069_2WS_1.01_RUNG



The device (ie: HI5069- CompactLogix® Weigh Scale Module) must also be configured with the correct device definition. Note that this configuration is completed automatically when using Application Code Manager or the Studio 5000® Import Library Objects wizard plug-in. For details on setting up the device, refer to the [Device Definition](#) section.

Graphic Symbols

Graphic Symbols are used as launch buttons within HMI applications to open up faceplate displays. Alternatively, faceplates may also be launch from related instructions such as the navigate to device faceplate buttons in the Process Library or the Machine Builder Library faceplates.

All icons display the following information:

- - Device label (Tag.@Description or custom label entered in parameter #104)
- - Device Warning/Fault Indication
- - Device not ready indication

See [Launch Buttons](#) for more general information on launch button diagnostics and usage.

FactoryTalk® View ME/SE Graphic Symbols

Graphic Symbol Name	Graphic Symbol	Description	Global Object Parameter Values
GO_LaunchFP		Faceplate navigation button with string tag label. This launch button graphic object allows the user to navigate to the device object faceplate.	#102: Backing Tag (e.g. {[PAC]Program::Program._InstanceName}) #104: Custom button label. Leave blank to use Tag.@Description #120: Display's left position (e.g. 100, optional) #121: Display's top position (e.g. 100, optional)

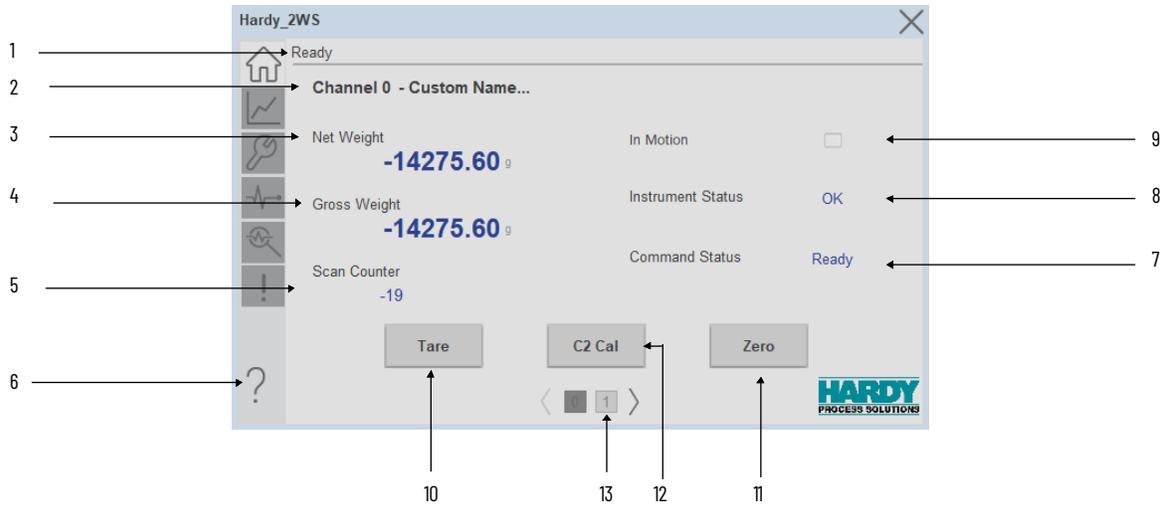
Faceplates

There are basic faceplate attributes that are common across all instructions. See [Basic Faceplate Attributes on page 17](#).

The faceplate title is linked to _InstanceName.@description, the .@description extended tag property of the Add-On Instruction instance. This is user configurable from controller/program tags in Studio 5000 Logix Designer.

Home Tab

The Home tab is the main tab of the faceplate. It contains Primary weight parameters as well weighing terminal parameters of the device, Device status information and primary commands of the device.



Item	Description
1	Banner
2	Channel number with custom naming
3	Net Weight
4	Gross Weight
5	Scan Counter
6	Help File button
7	Command status : When the tare command, Zero or C2 Cal Command is passed, it will show the status
8	Instrument status: It will show the current instrument status
9	0 = No motion ; 1 = In motion
10	The "Tare button" is pressed to initiate the "tare command" regardless of the stability of the weight value
11	The "Zero button" is pressed to execute the "zero command" regardless of the stability of the weight value. This feature is specifically designed for making minor adjustments to the "zero point" to compensate for drifting
12	The "C2 Cal" button is used to execute C2 Calibration.
13	Channel navigation: Tab Once available in same as channel Zero

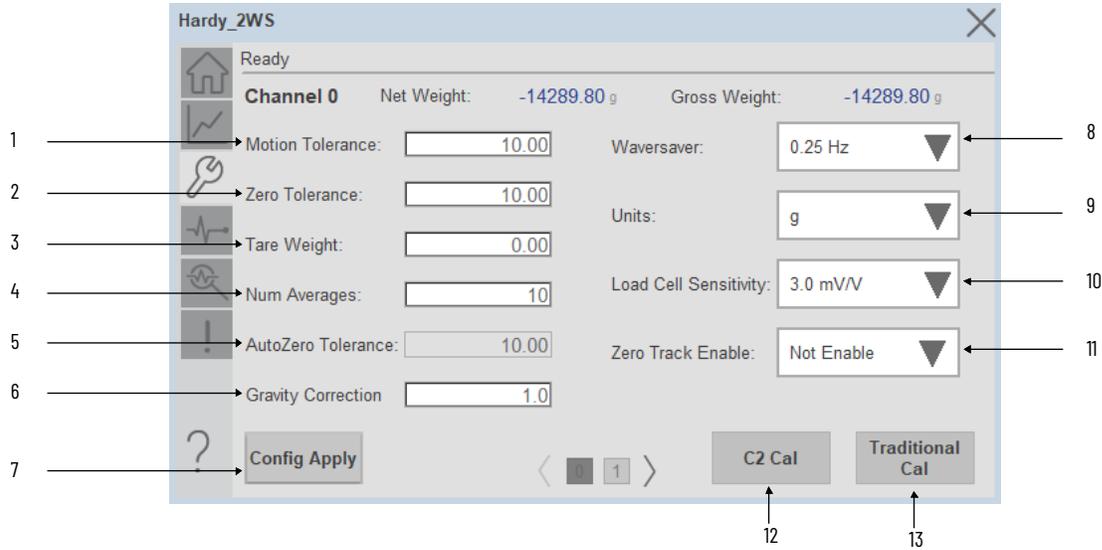
Trend Tab

Trends display values over time, often used to compare similar or related values and to allow operators to predict future states to make control action decisions. There are total two trends are displayed as follows Net Weight, Gross Weight.



Configure Tab

The Configure tab acts as a control center for technicians performing maintenance. It offers various settings that can be adjusted to fine-tune the performance of an object managed on another tab. These settings include utilizing the Waversaver® function, selecting the preferred unit format, activating zero track functionality, and defining tolerances for motion, zero point, and automatic zeroing. Additionally, the technician can set the tare weight, Gravity Correction, and the number of averages used for calculations. To ensure optimal performance, the tab also provides navigation buttons for calibrating the device.

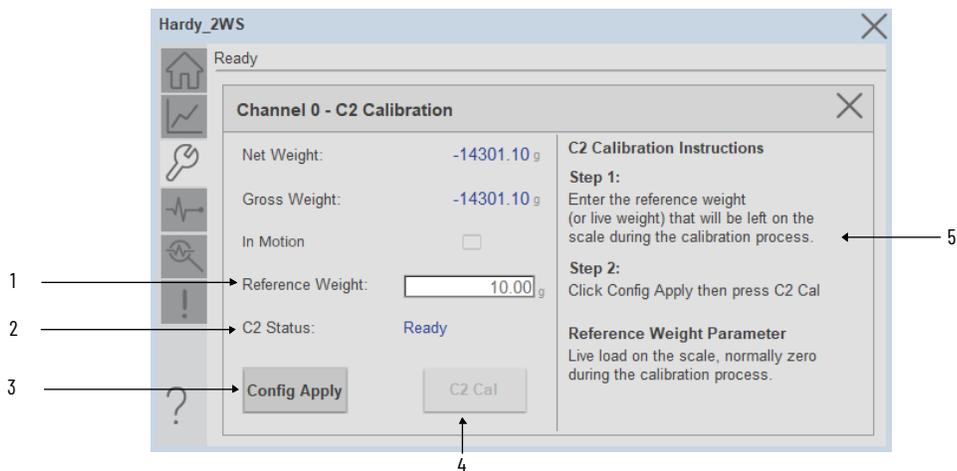


Item	Description
1	Setpoint of Motion Tolerance
2	Setpoint of Zero Tolerance
3	Setpoint of Tare weight
4	Setpoint of Num Averages
5	The AutoZero Tolerance setting becomes active when the Zero Track Enable selection is Enabled
6	Setpoint of Gravity Correction
7	The "Config Apply" button remains disabled until a change is made within the Configuration tab. Once any configuration element is modified (e.g., enabling a channel, adjusting motion tolerance), the button becomes active, prompting the user to apply the changes
8	WAVERSAVER® can be configured to ignore noise with frequencies as low as 0.25 Hz. One of five higher additional cut off frequencies may be selected to provide a faster instrument response time.
9	Units selection
10	Load cell sensitivity is a measure of how a load cell responds to changes in applied force, it is expressed in millivolts per volt (mV/V). Adjusting the sensitivity parameter is not required when using C2 load cells. When using non-C2 load cells, check the sensitivity rating on the load cell data sheet and adjust the setting accordingly.
11	Zero Track Enable
12	Navigate the C2 Calibration screen
13	Navigate Traditional Calibration screen



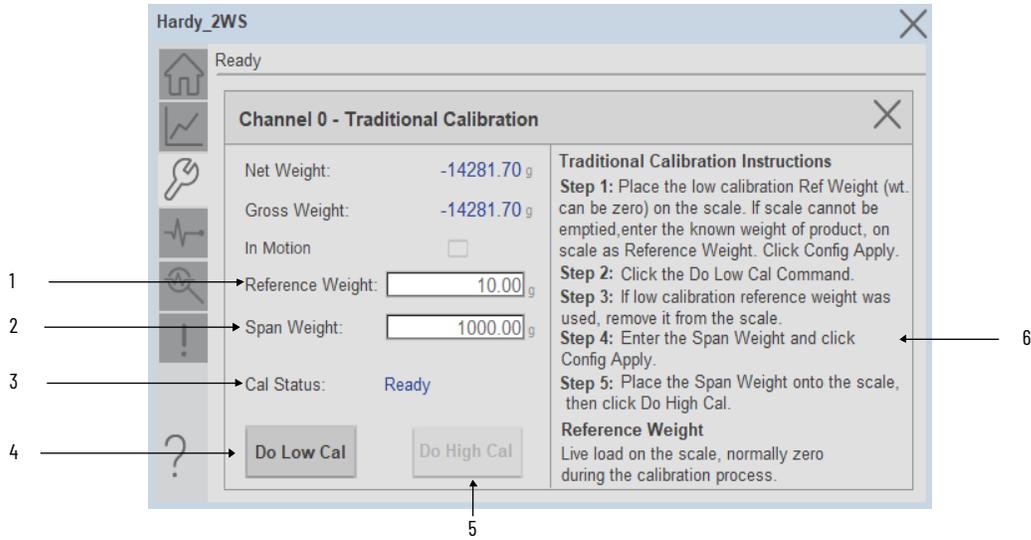
The Gravity Correction value displayed is rounded to the nearest 6 significant digits due to limitations of the 32-bit floating point values in FactoryTalk View ME. Values with up to 6 decimal places are written to the PLC and used in the device despite the display limitation.

C2 Calibration



Item	Description
1	Reference Weight:- The C2 calibration command executes a calibration process for the C2 system. It employs the Reference Weight value as a starting point and progresses through the weight range from zero to the maximum span weight.
2	C2 Status: Displays the real-time state of the C2 calibration process. Possible states include: Ready, Calibration In Progress, Fail, etc.
3	The "Config Apply" button remains disabled until a change is made within the Configuration tab. Once any configuration element is modified (e.g. adjusting Reference weight), the button becomes active, prompting the user to apply the changes
4	C2 Cal. C2 Calibration becomes available after configuration is applied successfully. Users can initiate the C2 calibration process once the configuration is complete.
5	The following steps are provided for user reference

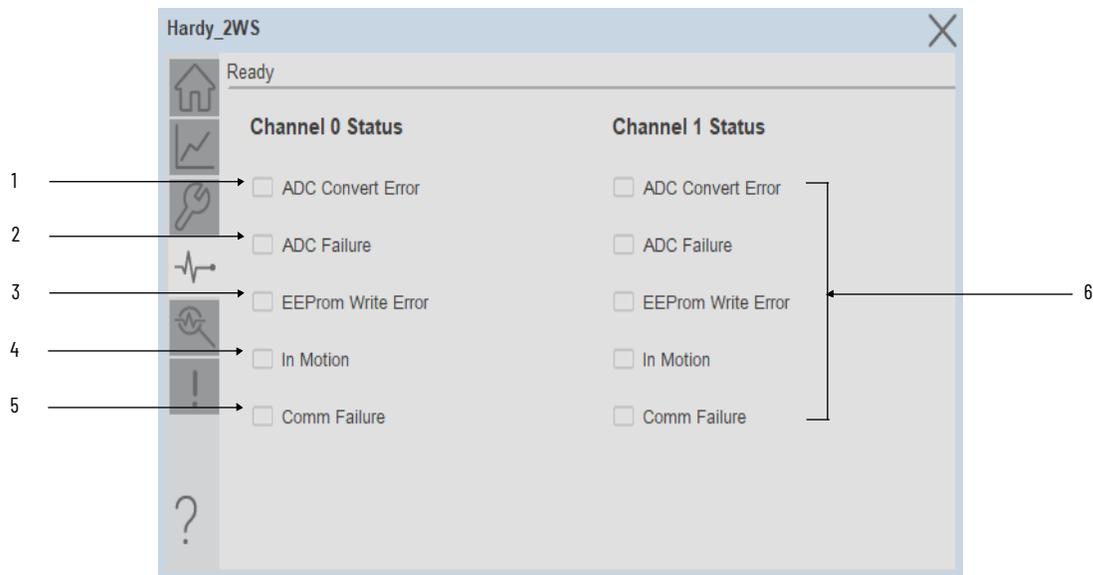
Traditional Calibration



Item	Description
1	Reference Weight:- Reference Weight calibration carries out a calibration process for the low calibration system. Taking the CalLow Weight value as a starting point, it progresses from zero to the maximum span weight.
2	Span Weight:- The Span Weight is a Calibration high reference point derived from an actual measured weight. This should not be confused with the Scale Capacity. If you have a 100-pound weight and you place it on the scale, the Span Weight would be 100 pounds.
3	Cal Status: Success, Fail, Fail - ADC error, Ready, Fail - motion, Do High Cal, Fail - Not Enough Counts Between Cal Low & Cal High Wgts, Command In progress
4	Do Low cal :- Do Low Calibration button will appear after configuration is applied. This button initiates the low calibration process.
5	The "Do High Calibration" button becomes available after the configuration is applied and the low calibration is complete. This button initiates the high calibration process.
6	The following steps are provided for user reference

Diagnostics Tab

The device diagnostics tab includes a list of information available in the device for troubleshooting. Diagnostics tab includes Device status and Failure Reason.

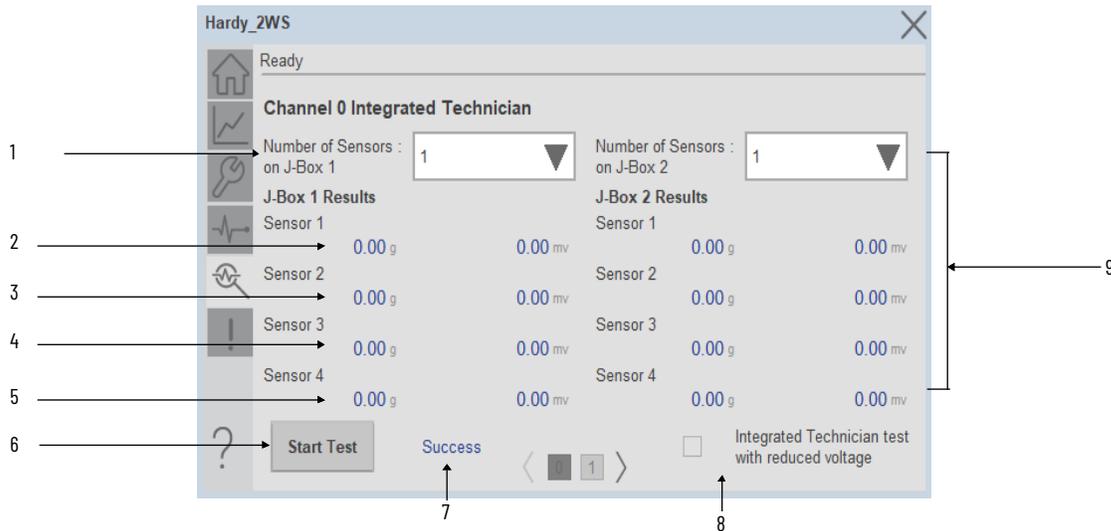


Item	Description
1	ADC Convert Error : Load cell input out of range (i.e., voltage not 0-15 mV and flashing red LED will display). Can result from overloaded or mismatched load cell. In this state weight readings do not respond to changes.
2	ADC Failure: Output from the A/D converter to processor is bad. The module shows a solid red LED
3	EEPROM Write Error: Module cannot write (save settings) to non-volatile memory. EEPROM is probably bad.
4	In Motion:-The rate of scale weight change over 1 second exceeds the motion tolerance setting. If the setting is too low, motion may be indicated when no changes are occurring.
5	Comm Failure
6	Channel1, same as Channel0

Integrated Technician Tab

The INTEGRATED TECHNICIAN™ (IT) tab is a system diagnostics utility which, in conjunction with an HI6020IT series

junction box, monitors the excitation circuit for possible malfunctions. IT reads individual load sensor voltages and weights, then isolates individual system components for quick and easy troubleshooting.

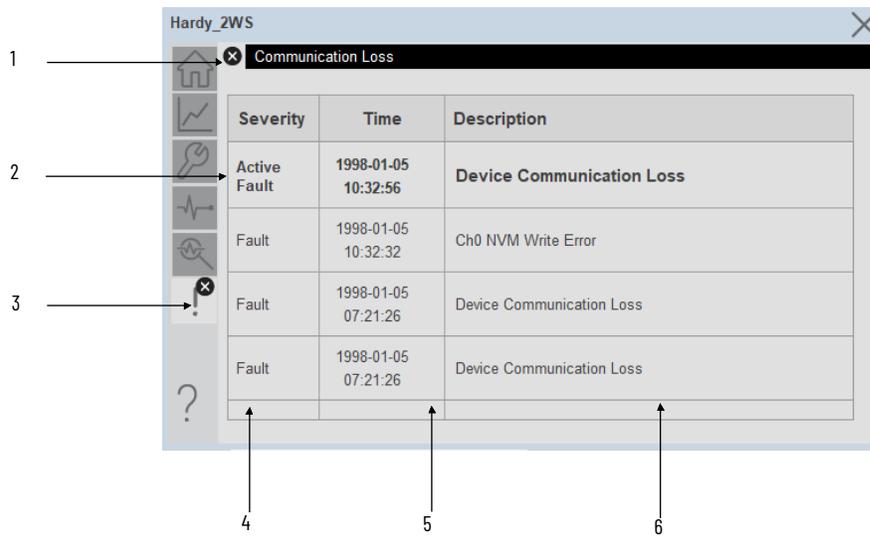


Item	Description
1	Number of Sensors Selection for Junction Box 1, (1 to 4)
2	Sensor 1: Weight and Voltage of Sensor1 connected to J-Box1
3	Sensor 2: Weight and Voltage of Sensor2 connected to J-Box1
4	Sensor 3: Weight and Voltage of Sensor3 connected to J-Box1
5	Sensor 4: Weight and Voltage of Sensor4 connected to J-Box1
6	Start Test Command
7	IT Status
8	Same as J-Box1

Fault Warning Tab

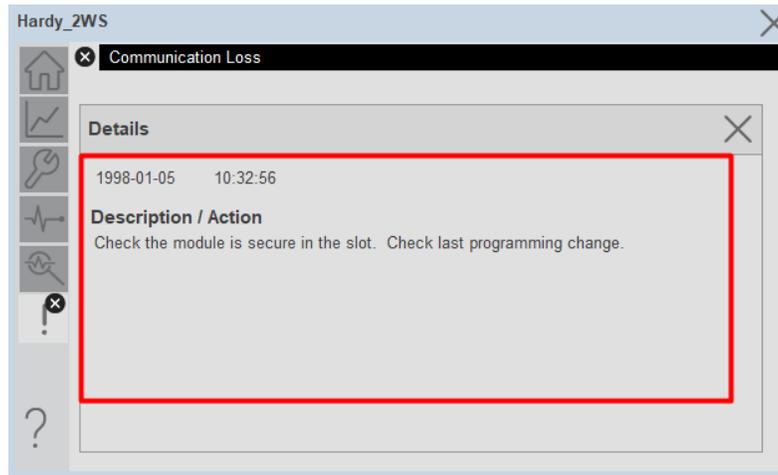
The Fault Warning tab displays information for up to four faults for the device. The fault table displays the Severity level (Fault, Warning or Active Fault), time (and date) and a description of the fault.

Note, only row 1 will display the “Active Fault” in the severity column if there is a current active fault, else it will display the last fault. Rows 2-4 only display past faults and warnings, not an active fault.



Item	Description
1	Banner
2	Last fault is in first row and show in bold if active
3	Fault tab icon visible when a fault is active
4	Fault severity
5	Fault event time
6	4 most recent fault/warning event messages

Click on any row in the fault table to view fault details. The details window provides a more detailed description and possible action steps to remedy condition.



Application Code Manager

All Hardy device objects have similar configuration parameters in Application Code Manager. The following section defines the common parameters.

Refer to the section [Using Application Code Manager](#) for complete details.

Definition Object: raC_Dvc_HI_5069xWS

This object contains the AOI definition and used as linked library to implement object. This gives flexibility to choose to instantiate only definition and create custom implement code. User may also create their own implement library and link with this definition library object.

Implementation Object: raC_LD_Dvc_HI_5069xWS

Parameter Name	Default Value	Instance Name	Definition	Description
RoutineName	{ObjectName}	{RoutineName}	Routine	Enter Routine name. Routine will be created and Object implement rung(s) inserted. A JSR will be inserted in MainRoutine. If routine name already exists, then object will be inserted into existing routine. By default, parameter is set to Object Name.
TagName	{ObjectName}	{TagName}	Backing Tag	Enter the backing tag of the main AOI. This will serve as the base tag name for other tags in this object that are derived from the base.
TagDescription	{ObjectDescription}	{TagDescription}		Tag Description of the main AOI backing tag
ModuleName	Mod_{ObjectName}	{ModuleName}	Input Parameter	Enter the Module Name. This is the name for the module that appears in the Controller Organizer tree.
IncludeHW	1			Allow ACM to create the Hardware Module. If the module already exists in the Controller Organizer, select False or existing module properties will be overwritten.
Slot	0		Input Parameter	Select the channel slot number
NumberOfChannel	One Channel		Input Parameter	Select FieldBus Format selected on device
RPI	4.0		Input Parameter	This is the Requested Packet Interval (RPI) of the module (1.0ms - 100ms).
ParentModule	Local		Input Parameter	Select the Parent Module. This represents the name of the communication adapter this module will communicate through. If connecting to a non-library object module, enter the name of the module only. If the module is connected directly to the controller ethernet port, enter "Local". Note: entering non-library object modules will result in the parameter displaying a red X. This will still generate properly as long as the entered name exists in the project.
ChassisName	{ParentModule}			Warning removal

Linked Libraries

Link Name	Catalog Number	Revision	Solution	Category
raC_Dvc_HI_5069xWS	raC_Dvc_HI_5069xWS	1.01	(RA-LIB) Device	Hardy

Configured HMI Content

HMI Content	Instance Name	Description
Launch Button SE	{ObjectName}_GO_LaunchFP	Global Object configured callout instance
Launch Button ME	{ObjectName}_GO_LaunchFP	Global Object configured callout instance

Attachments

Name	Description	File Name	Extraction Path
V1_raC_Dvc_Global	Graphic Symbols SE	(raC-1-SE) Graphic Symbols - Hardy Device.ggfx	{ProjectName}\Visualization\FTViewSE\GlobalObjects
V1_raC_Dvc_Global	Graphic Symbols ME	(raC-1-ME) Graphic Symbols - Hardy Device.ggfx	{ProjectName}\Visualization\FTViewME\GlobalObjects
V1_raC_Dvc_Global	Toolbox SE	(raC-1-SE) Toolbox - Hardy Device.ggfx	{ProjectName}\Visualization\FTViewSE\GlobalObjects
V1_raC_Dvc_Global	Toolbox ME	(raC-1-ME) Toolbox - Hardy Device.ggfx	{ProjectName}\Visualization\FTViewME\GlobalObjects
V1_raC_Dvc_HI_5069xWS	Faceplate SE	(raC-1_00-SE) raC_Dvc_HI_5069xWS-Faceplate.gfx	{ProjectName}\Visualization\FTViewSE\Displays
V1_raC_Dvc_HI_5069xWS	Faceplate ME	(raC-1_00-ME) raC_Dvc_HI_5069xWS-Faceplate.gfx	{ProjectName}\Visualization\FTViewME\Displays
V1_raC_Dvc_Hardy	View Designer	(raC-1_00-VD) raC_Dvc_Hardy.vpd	{ProjectName}\Visualization\ViewDesigner
V1_Hardy_Manual	Reference Manual	DEVICE-RM915C-EN-P.pdf	{ProjectName}\Documentation
V1_Hardy_Images	HMI Image Set	Hardy_Images.zip	{ProjectName}\Visualization\Images
V1_IO_Device_HMI_Tag	HMI Tag	FTViewStudio_HardyLibrary_Tags_1.xx.CSV	{ProjectName}\Visualization\

Rockwell Automation Support

Use these resources to access support information.

Technical Support Center	Find help with how-to videos, FAQs, chat, user forums, Knowledgebase, 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)	Download firmware, associated files (such as AOP, EDS, and DTM), and access product release notes.	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 compliance information on its website at rok.auto/pec.

Allen-Bradley, ArmorStart, CompactLogix, ControlLogix, FactoryTalk, Integrated Architecture, iTRAK, Kinetix, Logix 5000, MagneMotion, PlantPAx, PowerFlex, RSLinx, RSLogix, RSLogix 5000, SoftLogix, Stratix, Studio 5000, Studio 5000 Logix Designer, TechConnect, Rockwell Automation, and Rockwell Software are trademarks of Rockwell Automation, Inc.

EtherNet/IP is a trademark of ODVA, Inc.

Microsoft and Windows are trademarks of Microsoft.

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