Reference Manual

PlantPAx
Process Automation System

Rockwell Automation Library of Process Objects

Version 3.1

Allen-Bradley • Rockwell Software

Rockwell Automation

Important User Information

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

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

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

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

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

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

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

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

	**WARNING:** Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.
	**ATTENTION:** Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence.
IMPORTANT	Identifies information that is critical for successful application and understanding of the product.

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

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

This manual is updated throughout for Version 3.1 of the Rockwell Automation Library of Process Objects. Changes for this revision are marked by change bars, as shown to the right of this paragraph.

New and Updated Information

This table contains the changes made to this revision.

Topic	Page
New categories for the Library object overviews	14
Motor Control overviews have new objects for PowerFlex® drive (P_PF52x), Smart Motor Controllers (P_SMC50,P_ SMCFlex), and Overload Relays (P_E1Plus, P_E3Ovld, P_E300Ovld)	27, 28, 29, 30, 31
New Steam Table object overviews (P_Steam_hs, P_Steam_ph, P_Steam_ps)	39
Graphics for Built-in Instructions (Autotune, CC, IMC, MMC, PIDE, RMPS, TOT) added to overviews	46, 47, 48, 49
Alarm types table updated for new categories and objects	88
Status and Control bits for Suppress and Disabled alarms are separate for version 3.1 for import behavior	92
Alarm severity range added for version 3.1	94
Adds FactoryTalk View ME values for positioning faceplates	108
Expands on procedures for using the process library	113
Logix tag severity values section added to Alarms Builder appendix	131
Custom build procedures for FactoryTalk View® SE alarms repositioned for Alarms Builder tool	138
New appendix section documents built-in instruction faceplates	183

Notes:

Chapter 3

Common Configuration Considerations

Chapter 4

Use the Library

Appendix A

Online Configuration Tool

Appendix B

Alarms Builder Tool

Appendix C

Color Change Utility

Appendix D

Additional Add-On Instructions

Appendix E

HMI Security Codes Configuration

Appendix F

Faceplates for Built-in Logix5000 Instructions

The purpose of this manual is to facilitate the use of the Rockwell Automation Library of Process Objects and associated productivity tools. The library consists of predefined application code for Logix controllers and graphics for FactoryTalk View software.

The combination of programming logic and HMI visualization files lets you accelerate control project development by starting from a proven, tested, and documented set of code. The instructions provide common process objects for controlling and interacting with motors, valves, pumps, and numerous other devices.

See page 14 for links to the individual reference manuals for objects that comprise the Rockwell Automation Library of Process Objects.

Additional Resources

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

Resource	Description
PlantPAx Process Automation System Selection Guide, publication PROCES-SG001	Provides information to assist with equipment procurement for your PlantPAx® system.
PlantPAx Process Automation System Reference Manual, publication PROCES-RM001	Provides characterized recommendations for implementing your PlantPAx system.
PlantPAx System Application Templates Quick Start, publication PROCES-QS001	Explains procedures for configuring pre-built controller and HMI templates for a basic framework of the PlantPAx system.
FactoryTalk View SE Edition User Manual, publication VIEWSE-UM006	Provides details on how to use this software package for developing and running human-machine interface (HMI) applications that can involve multiple users and servers, distributed over a network.
FactoryTalk View Machine Edition User Manual, publication VIEWME-UM004	Provides details on how to use this software package for creating an automation application.
Rockwell Automation Library of Logix Diagnostic Objects, publication PROCES-RM003	Provides Add-On Instructions for monitoring and diagnostic information of Logix controllers.
Rockwell Automation Library of Steam Table Instructions, publication PROCES-RM004	These Add-On Instructions are an extension of the Library of Process Instructions to provide steam table calculations.
FactoryTalk Alarms and Events System Configuration Guide, publication FTAE-RM001	Provides details on how to install, configure, and use FactoryTalk Alarms and Events services as part of a FactoryTalk-enabled automation system.
Logix5000™ Controllers Add-On Instructions Programming Manual, publication 1756-PM010	Provides information for designing, configuring, and programming Add-On Instructions.

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.

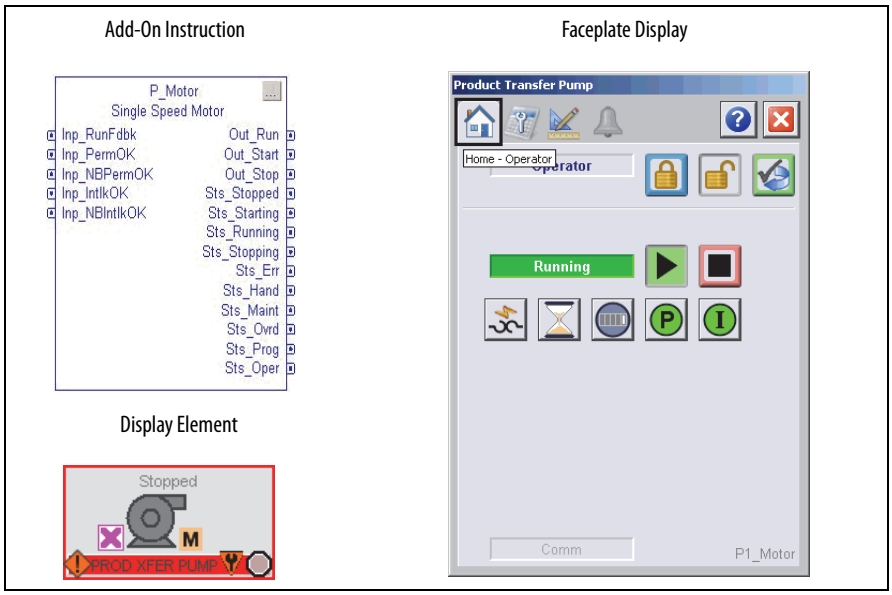
Notes:

Overview

Welcome, and thank you for selecting the Rockwell Automation Library of Process Objects. The Library is a predefined collection of coded objects that offer proven strategies, functionality, and known performance for your process control system.

The library features a variety of instructions for motors, valves, drives, interlocks, permissives, and additional objects that can be used with the PlantPAx system. However, using the library objects is **not** equivalent to designing a PlantPAx system. To have a PlantPAx system, you must use the PlantPAx Selection Guide, publication PROCES-SG001, to properly size your system and use the PlantPAx Reference Manual, publication PROCES-RM001, to implement system guidelines.

Library elements include controller code (Add-On Instructions), display elements (global objects), and faceplates that provide controller-ready logic and visualization tools for the operator.



The table describes the topics in this chapter.

Library Benefits

Using the Library of Process Objects offers these benefits:

- Provides reusable engineering designs with modular programming code
- Simplifies process development with controller-ready logic
- Provides visualization of device and diagnostic information

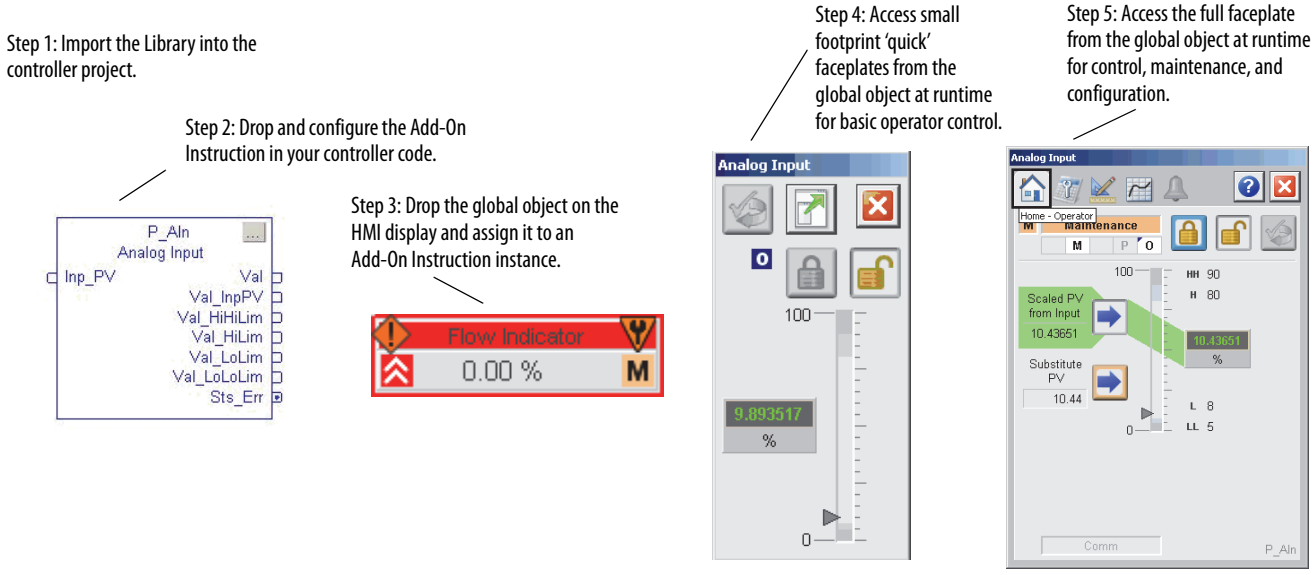
Reusable design — The instruction set lets you control, monitor, and troubleshoot the process with little added engineering effort. Add-On Instructions provide modules of code, with predefined functionality, to monitor and control devices. Developing a system becomes configuring the ready-made objects rather than having to design functionality for each particular tool.

Simplified development — Each instance of an instruction can be configured without changing the source definition. Each instruction has been designed to satisfy a broad range of users. Once the visualization elements are added to your project, you can configure devices from the associated faceplates rather than having to open controller tags.

Visualization of real-time data — Global objects provide access to faceplates that let you see how the device is operating in real-time conditions. Alarms and diagnostic information alert operators to monitor specific conditions for well-informed business decisions.

When the predefined logic is coupled to display elements and faceplates in FactoryTalk View Studio software, objects are configured in a drag-and-drop environment as shown in Figure 1.

Figure 1 - Configuring Library Objects



By using a library of consistent elements, you improve the operability, maintainability, and efficiency of your PlantPAx system. All objects have a common set of security access levels, modes of operation, symbols, and indicators.

See Table 2 on page 14 for a complete listing of the library objects.

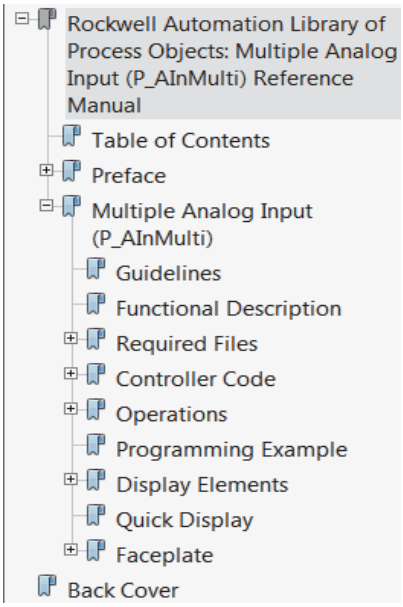
Table 1 describes additional Rockwell Automation libraries that the PlantPAx system leverages for process-specific content.

Table 1 - Rockwell Automation Library Resources

Resource	Description
Rockwell Automation Library of Logic Diagnostic Objects, publication PROCES-RM003	Provides Add-On Instructions that monitor controllers to provide diagnostic information.
Rockwell Automation Library of Steam Table Instructions, publication PROCES-RM004	Provides Add-On Instructions for steam table calculations.
Rockwell Automation Sequencer Object, publication PROCES-RM006	Provides instructions for a controller-based step sequencing solution that reduces engineering time by automating common operator procedures.
Rockwell Automation Library of Process Objects: Operator Prompt, publication SYSLIB-RM046	Provides instructions for using a generic mechanism to interact with the operator for any task.

Using Library Object Documentation

Figure 2 - Add-On Instruction Bookmarks



Each Library object has its own reference manual that defines the parameters and display elements that are specific to the object. The manuals are structured with the following sub-sections (bookmarks in the PDFs as shown in Figure 2):

- Guidelines – Explains when to use this instruction and alternative Add-On instructions for situations that do not apply for this instruction.
- Functional Description – Provides details on how the instruction operates to acquaint you with the instruction's capabilities.
- Required Files – Includes the controller and visualization files that you must import into your project to use this instruction.
- Controller Code –Describes the input and output parameters and local configuration tags for controller configuration and maintenance.
- Operations – Describes primary operations for Add-On Instructions, including modes, alarms, and simulation.
- Programming Example (selected manuals) – Illustrates the use of the instruction for a better understanding of the instruction's logic.
- Display Elements – Depicts the display elements to aid in choosing the ones that you need.
- Quick Display – A small screen that lets operators perform simple interactions with the respective instruction. From the Quick Display, you can navigate to the faceplate for full access for operation, maintenance, and configuration.
- Faceplate – Explains how to use and understand the faceplate.

A brief description of each Library object is provided in the following pages to help you select functionality for your system. The overviews include sample displays and links to the respective Add-On Instruction manual.

Table 2 lists objects per category.

Table 2 - Library Object Table of Contents

I/O Processing

The Process Objects in this group provide analog and discrete input/output signal processing. Pressure/temperature compensated flow calculation and cylindrical tank level interpolation are also provided.

Table 3 - I/O Processing

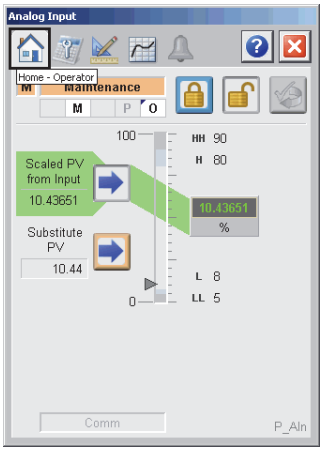
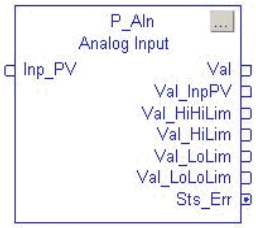
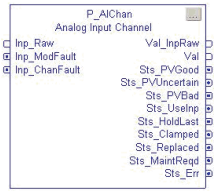
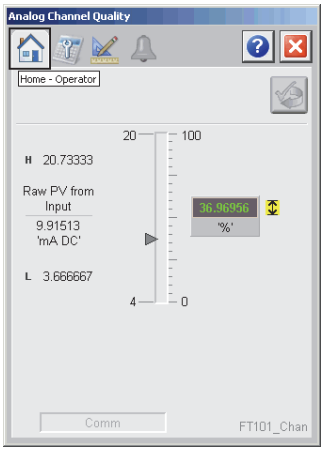
Process Object Description	Object Elements
Basic Analog Input (P_AIn) The P_AIn instruction monitors one analog value, typically from a channel of an analog input module, and provides alarms when the analog value exceeds user-specified thresholds (high and low). This instruction also provides for linear scaling of an analog input value from raw (input) units to engineering (output) units, and entry of a substitute Process Variable. Click the link to access the Reference Manual: SYSLIB-RM001	Global Object  Faceplate  Add-On Instruction 
Analog Input Channel (P_AIChan) The P_AIChan instruction monitors one analog input channel and provides a configurable failure alarm. This instruction is usually associated with other instructions. The P_AIChan faceplate is called from other faceplates, such as the associated analog input instruction faceplate, P_PIDE, and the Dosing faceplates. Click the link to access the Reference Manual: SYSLIB-RM042	Add-On Instruction  There are no dedicated display elements for this instruction. Faceplate 

Table 3 - I/O Processing

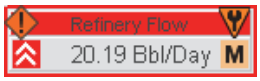
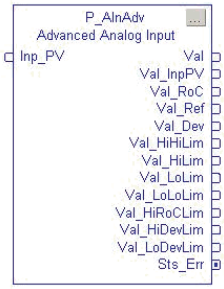
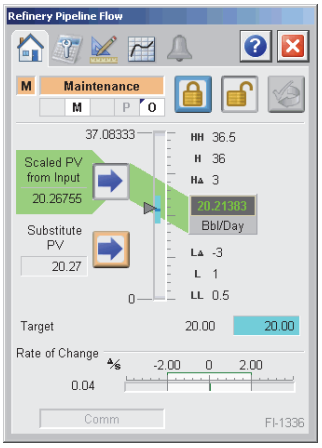
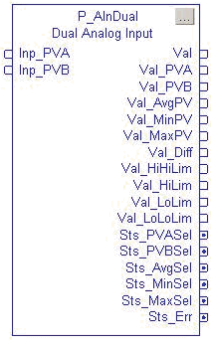
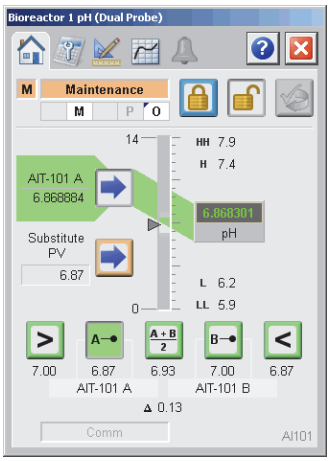
Process Object Description	Object Elements
Advanced Analog Input (P_AInAdv) The P_AInAdv instruction monitors one analog value, typically from an analog input I/O module. This instruction has the following advanced features that are not included in the basic analog input: • Square root scaling to provide positive or negative flow values • Calculation of the PV rate of change and configurable high rate of change alarming • Alarms for deviation from a reference value Click the link to access the Reference Manual: SYSLIB-RM018	Global Object  Add-On Instruction  Faceplate 
Dual Sensor Analog Input (P_AInDual) The P_AInDual instruction monitors one analog process variable (PV) by using two analog input signals (dual sensors, dual transmitters, dual input channels). This instruction has the following advanced features that are not included in the basic analog input: • Dual inputs • Alarm if difference between the two input PVs exceeds a configured limit Click the link to access the Reference Manual: SYSLIB-RM019	Global Object  Add-On Instruction  Faceplate 

Table 3 - I/O Processing

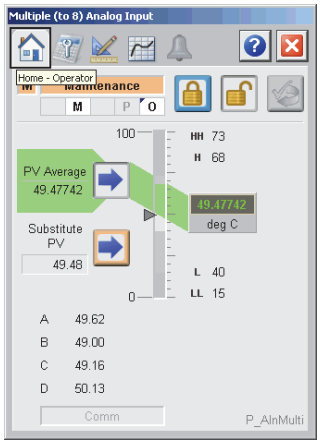
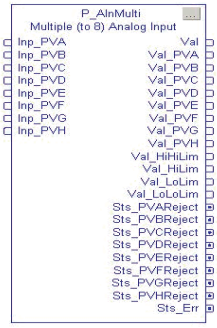
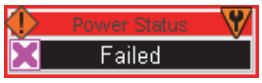
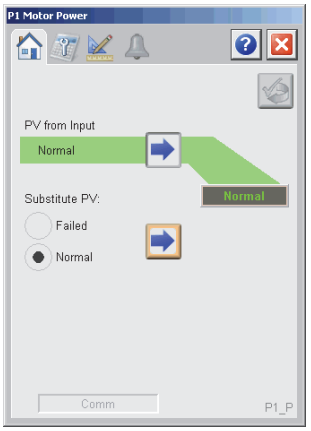
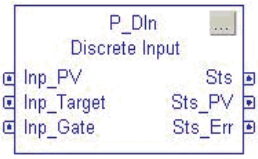
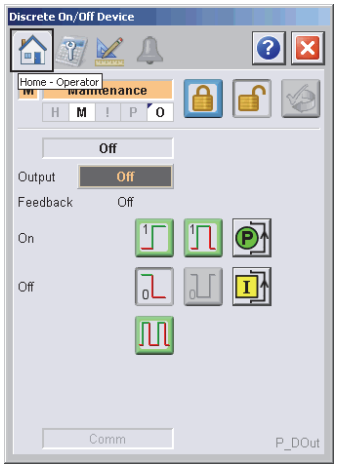
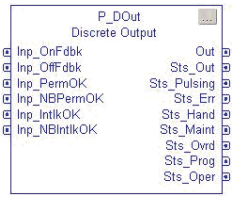
Process Object Description	Object Elements
Multiple Analog Input (P_AInMulti) The P_AInMulti instruction monitors one analog process variable (PV) by using up to eight analog input signals (sensors, transmitters, input channels). Use this instruction if you want to display a temperature or other process variable by averaging multiple measurements. Click the link to access the Reference Manual: SYSLIB-RM026	Global Object  Faceplate  Add-On Instruction 
Discrete Input Object (P_DIn) The P_DIn instruction is used to receive and process a single discrete condition, typically for a channel of a discrete input card. It can be used with any discrete (BOOL) signal. You can use this instruction to display the state of a process temperature, level, flow, proximity, pressure, or other switch. Click the link to access the Reference Manual: SYSLIB-RM003	Global Object  Faceplate  Add-On Instruction 
Discrete Output (P_DOut) The P_DOut instruction controls a device by using a single discrete output signal and monitors feedback from the device to check for device failures. This instruction operates in a variety of modes, and can provide steady, single pulsed, or continually pulsed output. The P_DOut instruction can be a good choice for pilot lights or stack lights that require blinking. Click the link to access the Reference Manual: SYSLIB-RM029	Global Object  Faceplate  Add-On Instruction 

Table 3 - I/O Processing

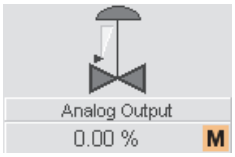
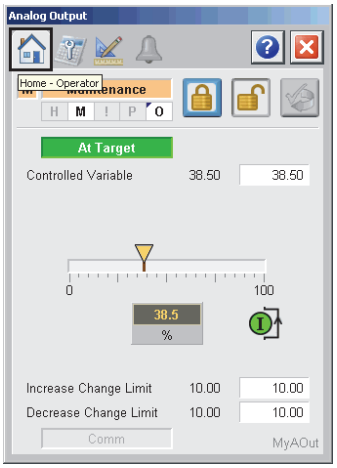
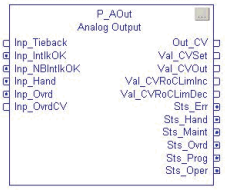
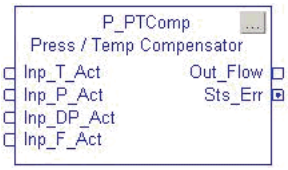
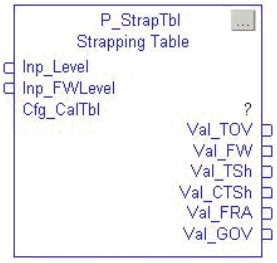
Process Object Description	Object Elements
Analog Output (P_AOut) The P_AOut instruction is used to manipulate an analog output to control a field device, such as a control valve or motorized gate positioner. The output responds to an Operator (manual) or Program setting of the Controlled Variable (CV) signal. The P_AOut instruction controls the analog output in a variety of modes, monitoring for fault conditions. Click the link to access the Reference Manual: SYSLIB-RM011	Global Object  Faceplate  Add-On Instruction 
Pressure/Temp. Compensated Flow (P_PTComp) The P_PTComp instruction is used to calculate a flow at standard temperature and pressure, essentially a mass flow rate, given a volumetric flow rate or differential pressure measurement. This instruction also requires measurements of the actual temperature and pressure of the flowing gas. Click the link to access the Reference Manual: SYSLIB-RM032	The P_PTComp Instruction is intended as a calculation function only, between other blocks, and no HMI components are provided. Add-On Instruction 

Table 3 - I/O Processing

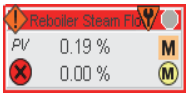
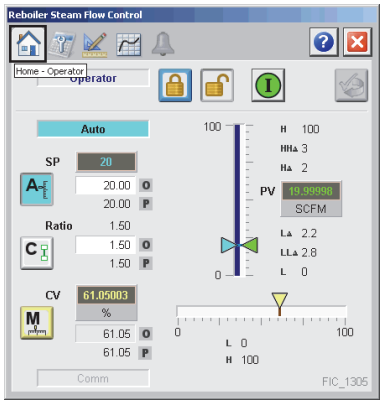
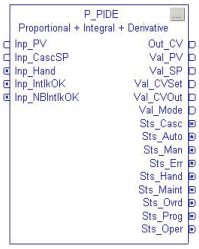
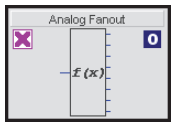
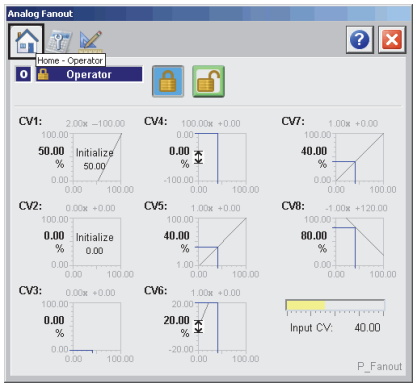
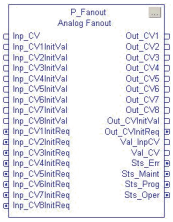
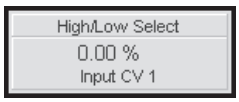
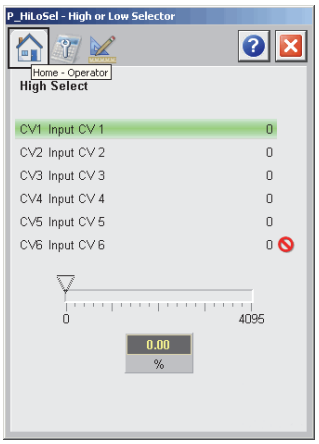
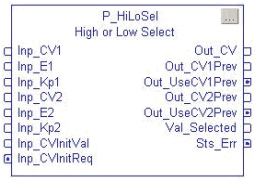
Process Object Description	Object Elements
Tank Strapping Table (P_StrapTbl) The P_StrapTbl instruction calculates the volume of product in an upright cylindrical tank, given the level of the product and the tank calibration table. This instruction can optionally compensate for free water at the bottom of the tank (given a product/water interface level) or for thermal expansion of the tank shell (given the coefficient of linear expansion of the shell material and product and ambient temperatures). Click the link to access the Reference Manual: SYSLIB-RM033	The P_StrapTbl Instruction is intended as a calculation function only, between other blocks, and no HMI components are provided. Add-On Instruction 

Notes:

Regulatory Control

The Process Objects in this group provide regulatory control of a final process element.

Table 4 - Regulatory Control

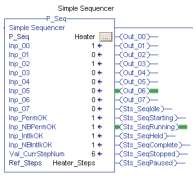
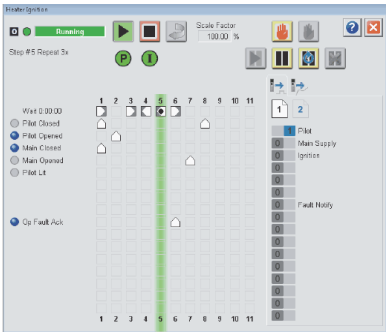
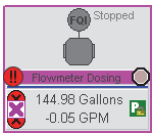
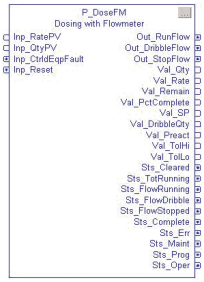
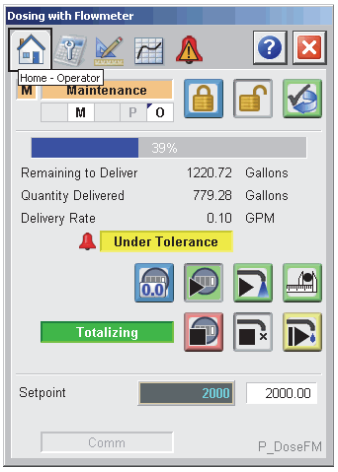
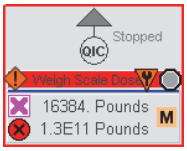
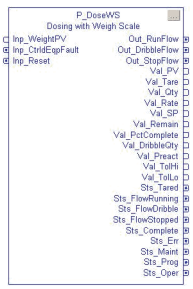
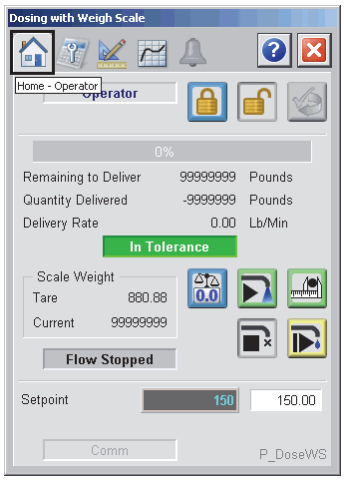
Process Object Description	Object Elements
Proportional + Integral + Derivative Enhanced (P_PIDE) The P_PIDE instruction provides the functionality of the PIDE built-in instruction for PID loop control and additional alarm status information, including limits and severities. Use this instruction when you plan to use the PIDE for loop control and provide visualization to the operator. Click the link to access the Reference Manual: SYSLIB-RM045	Global Object  Faceplate  Add-On Instruction 
Analog Fanout (P_Fanout) The P_Fanout instruction fans one 'primary' analog output signal out to multiple 'secondary' users or devices. Each secondary output has configurable gain and offset. The instruction applies minimum and maximum clamping limits to each output (secondary) CV. Click the link to access the Reference Manual: SYSLIB-RM030	Global Object  Faceplate  Add-On Instruction 
High or Low Selector (P_HiLoSel) The P_HiLoSel instruction selects the lowest of the (up to 6) incoming CVs or the highest of the incoming CVs and outputs the value. The unselected CVs are flagged to track the selected CV. The tracking value can optionally be offset by an amount equal to the upstream PID/PIDE Gain times Error to avoid problems with ever-decreasing (if low-select) or ever-increasing (if high-select) output. Click the link to access the Reference Manual: SYSLIB-RM047	Global Object  Faceplate  Add-On Instruction 

Notes:

Procedural Control

The Process Objects in this group provide procedural control definition via a series of discrete sequential actions.

Table 5 - Procedural Control

Process Object Description	Object Elements
Sequencer Object (P_Seq) The P_Seq instruction is a controller-based step sequencing solution that reduces engineering time by automating common operator procedures. The step-by-step configuration makes it easy to adjust procedures directly from the HMI displays. Click the link to access the Reference Manual: PROCES-RM006	Global Object  Add-On Instruction  Faceplate 
Flowmeter Dosing (P_DoseFM) The P_DoseFM instruction controls an ingredient addition that uses a flowmeter to measure the amount of ingredient added. The flowmeter can be an analog flowmeter (signal proportional to flow), a pulse generating flowmeter (pulse count proportional to quantity delivered), or a digital flowmeter providing flow rate or quantity (totalized flow) information. Click the link to access the Reference Manual: SYSLIB-RM020	Global Object  Add-On Instruction  Faceplate 
Weigh Scale Dosing (P_DoseWS) The P_DoseWS instruction controls an ingredient addition that uses a weigh scale to measure the amount of ingredient added. The weigh scale can be on the receiving vessel (gain in weight) or on the sourcing vessel (loss in weight). The weigh scale can be connected via an analog input, device network, or other connection. Click the link to access the Reference Manual: SYSLIB-RM021	Global Object  Add-On Instruction  Faceplate 

Notes:

Motors

The Process Objects in this group provide control and monitoring of drives, smart motor controllers, and overload relays.

Table 6 - Motors

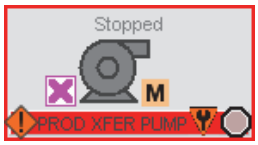
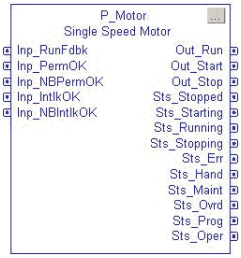
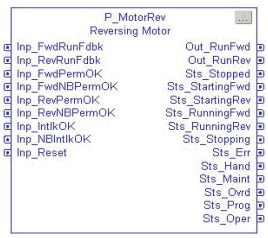
Process Object Description	Object Elements
Single-speed Motor (P_Motor) The P_Motor instruction controls a non-reversing, single-speed motor in a variety of modes and monitors for fault conditions. The motor can use a full voltage starter (FVNR), a soft starter, or other motor protective equipment. The instruction also provides run feedback and a display of actual motor status. Click the link to access the Reference Manual: SYSLIB-RM006	Global Object  Faceplate  Add-On Instruction 
Two-speed Motor (P_Motor2Spd) The P_Motor2Spd instruction controls a non-reversing, two-speed motor (fast/slow/stopped) in a variety of modes and monitors for fault conditions. The motor can optionally have run feedback that, if available, is used to confirm that the motor is running at the commanded speed, and alarm if not. Click the link to access the Reference Manual: SYSLIB-RM012	Global Object  Faceplate  Add-On Instruction 
Reversing Motor (P_MotorRev) The P_MotorRev instruction controls a reversing motor (FVR, forward/reverse/stopped motor) in a variety of modes and monitors for fault conditions. This instruction can optionally have run feedback that, if available, is used to confirm that the motor is running in the commanded direction, and alarm if not. Click the link to access the Reference Manual: SYSLIB-RM013	Global Object  Faceplate  Add-On Instruction 

Table 6 - Motors

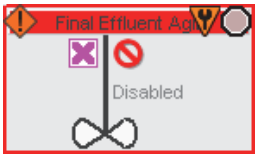
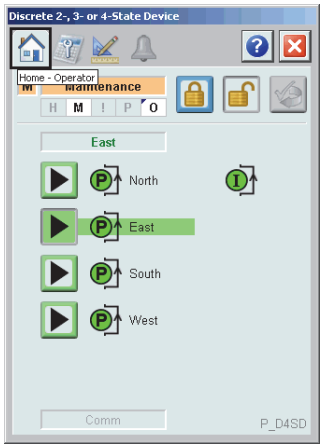
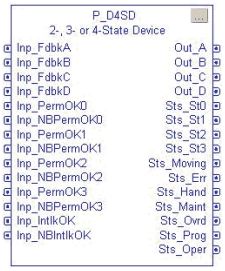
Process Object Description	Object Elements
Hand-Operated Motor (P_MotorHO) The P_MotorHO instruction monitors a locally controlled (hand-operated) motor. The P_MotorHO instruction supports single-speed motors, two-speed motors, and reversing motors. The instruction also supports an optional trip function and output, used to stop the motor. Click the link to access the Reference Manual: SYSLIB-RM022	Global Object  Faceplate  Add-On Instruction 
Discrete 2-, 3-, or 4-state Device (P_D4SD) The P_D4SD instruction controls and monitors feedback from a discrete 2-state, 3-state, or 4-state device, including a multiple-speed motor or a multiple-position valve. The instruction controls up to four discrete outputs, with configurable states of each output in the various device states, and monitors up to four discrete feedback inputs. Click the link to access the Reference Manual: SYSLIB-RM028	Global Object  Faceplate  Add-On Instruction 

Table 6 - Motors

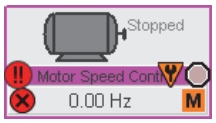
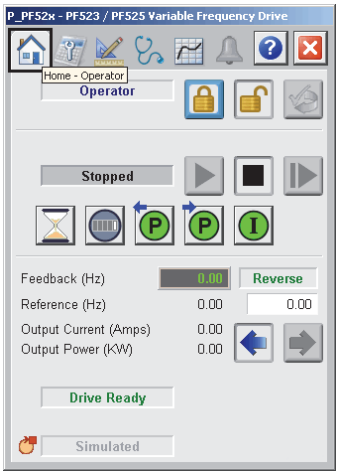
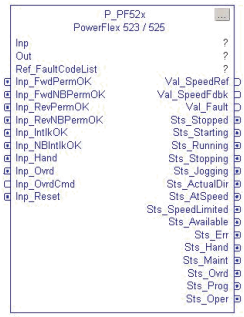
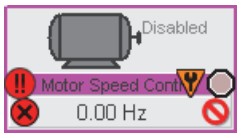
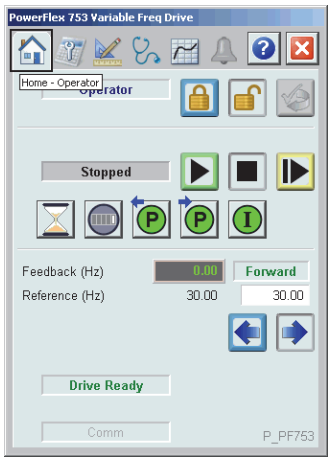
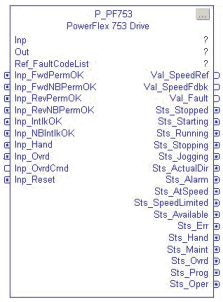
Process Object Description	Object Elements
PowerFlex 523/525 Drives (P_PF52x) The P_PF52x instruction is used to control and monitor a PowerFlex 523 or PowerFlex 525 variable-frequency drive. The instruction collects and displays diagnostic information from the drive by using configured data links on the EtherNet/IP interface. The instruction also provides the following capabilities: • Starting, stopping, jogging of drive, and setting speed reference and direction. • Monitoring of run feedback, display of actual drive status, including acceleration, deceleration, direction, and speed. This instruction is provided as a rung import for installation. Click the link to access the Reference Manual: SYSLIB-RM048	Global Object  Faceplate  Add-On Instruction 
PowerFlex 753 Drive (P_PF753) The P_PF753 instruction operates one variable-speed motor by using a drive (AC variable frequency) in a variety of modes, monitoring for fault conditions. This instruction is designed to work with the PowerFlex 753 drive and a 20-COMM-E Ethernet communication module. The instruction displays drive information, including faults, alarms, and general status. This instruction is provided as a rung import for installation. Click the link to access the Reference Manual: SYSLIB-RM044	Global Object  Faceplate  Add-On Instruction 

Table 6 - Motors

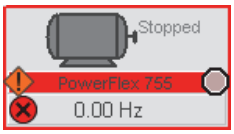
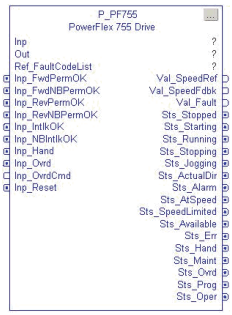
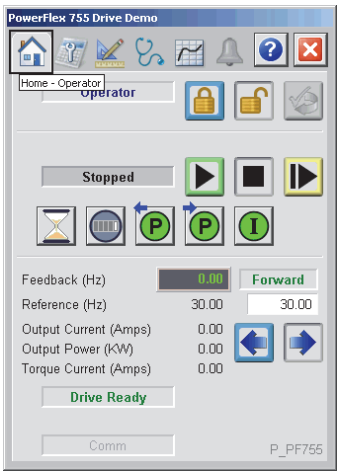
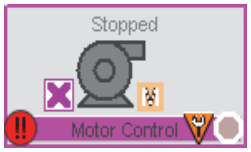
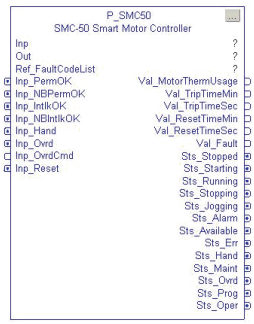
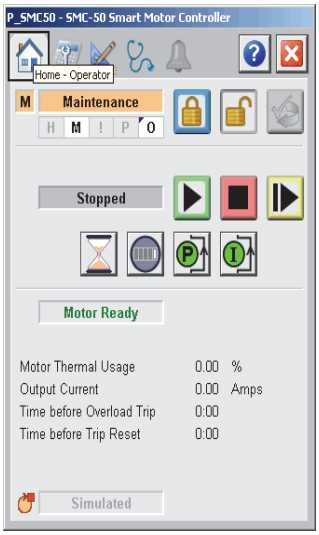
Process Object Description	Object Elements
PowerFlex 755 Drive (P_PF755) The P_PF755 instruction operates one variable-speed motor by using a drive (AC variable frequency or DC) in a variety of modes, monitoring for fault conditions. This instruction is designed to work with a PowerFlex 755 variable frequency AC drive that is communicating with the controller over an EtherNet/IP network. The instruction also works with a PowerFlex 753 drive with an enhanced Ethernet card. This instruction is provided as a rung import for installation. Click the link to access the Reference Manual: SYSLIB-RM040	Global Object  Add-On Instruction  Faceplate 
SMC-50 Smart Motor Controller (P_SMC50) The P_SMC50 instruction controls and monitors a motor via an SMC-50 Smart Motor Controller (soft starter). The instruction communicates with the motor controller to start, stop, and jog the motor. The instruction also monitors the status of the motor, detects motor failure to start or stop, and displays motor runtime information. The runtime data includes power, power factor, motor thermal usage, time to trip, time until reset, and motor controller fault codes. This instruction is provided as a rung import for installation. Click the link to access the Reference Manual: SYSLIB-RM052	Global Object  Add-On Instruction  Faceplate 

Table 6 - Motors

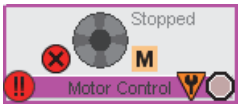
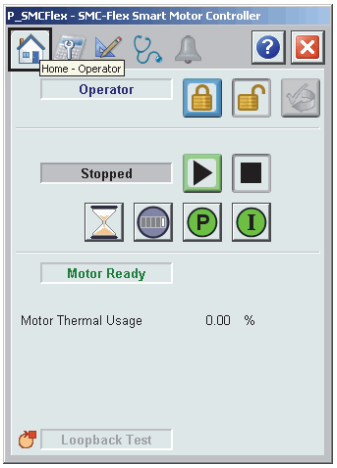
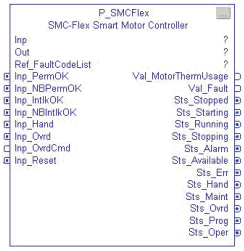
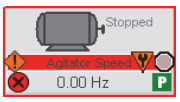
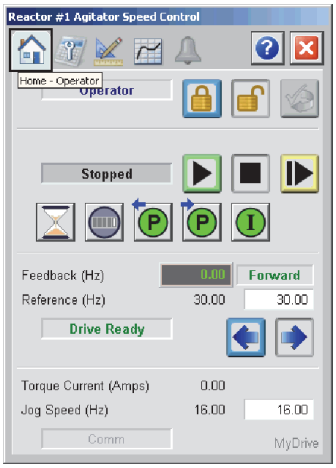
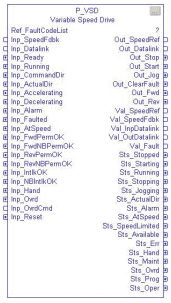
Process Object Description	Object Elements
SMC Flex Smart Motor Controller (P_SMCFlex) The P_SMCFlex instruction controls a motor by using a SMCFlex series Smart Motor Controller (soft starter). The instruction communicates with the motor controller to start and stop the motor. The instruction also monitors the status of the motor, detects motor failure to start or stop, and displays motor runtime information. The runtime data includes phase currents, motor power and power factor, motor thermal usage (% MTU), and motor controller fault codes. This instruction is provided as a rung import for installation. Click the link to access the Reference Manual: SYSLIB-RM053	Global Object  Faceplate  Add-On Instruction 
Variable-speed Drive (P_VSD) The P_VSD instruction operates one variable speed motor by using a drive (AC variable frequency or DC). This instruction is designed to work with all currently available and many legacy Allen-Bradley® drives, including: • Bulletin 1336 • Bulletin 1395 • PowerFlex 4/40/70/ 700/7000 • PowerFlex DC Drives This instruction is provided as a rung import for installation. Click the link to access the Reference Manual: SYSLIB-RM016	Global Object  Faceplate  Add-On Instruction 

Table 6 - Motors

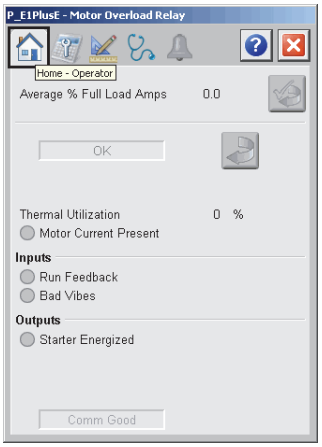
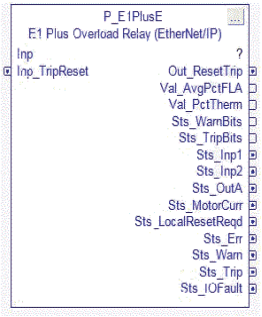
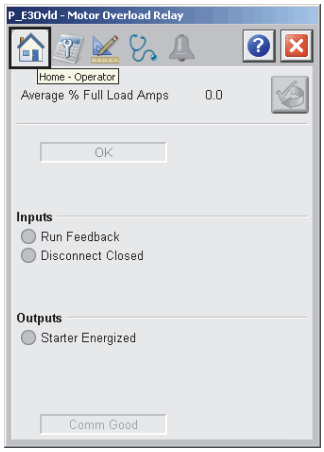
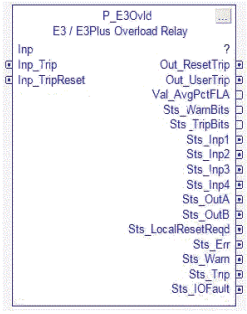
Process Object Description	Object Elements
E1 Plus™ Overload Relay (P_E1 PlusE) The P_E1 Plus (EtherNet/IP) instruction controls and monitors an E1 Plus overload relay by using a 193-ETN EtherNet/IP interface. The instruction monitors the overload relay for warning and trip conditions, displays motor current as a percentage of Full Load Amps (% FLA), and percentage of motor thermal utilization (% MTU). A list includes the causes of the last five overload trips. The instruction also provides a limited capability for remote reset of overload trips. This instruction is provided as a rung import for installation. Click the link to access the Reference Manual: SYSLIB-RM049	Global Object  Faceplate  Add-On Instruction 
E3™/E3Plus™ Overload Relay (P_E3Ovld) The P_E3Ovld instruction controls and monitors the following overload relays: • 193/592-EC1 • 193/592-EC2 • 193/592-EC3 • 193/592-EC5 The instruction monitors the relays by using a built-in DeviceNet interface or by using a 2100-ENET EtherNet/IP interface. The instruction reports warning and trip conditions, displays motor current as a percentage of Full Load Amps (% FLA), and provides commands to initiate a remote trip and a remote trip reset. This instruction is provided as a rung import for installation. Click the link to access the Reference Manual: SYSLIB-RM050	Global Object  Faceplate  Add-On Instruction 

Table 6 - Motors

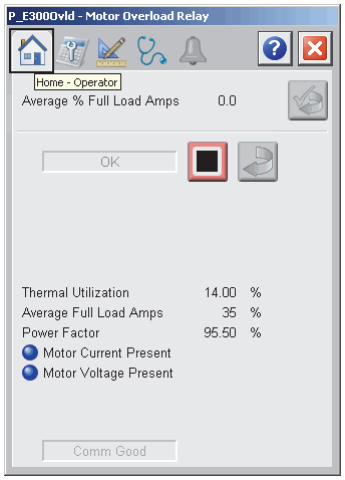
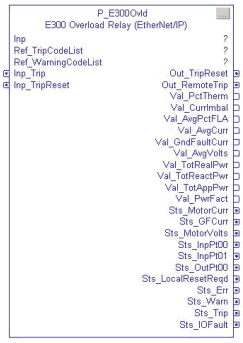
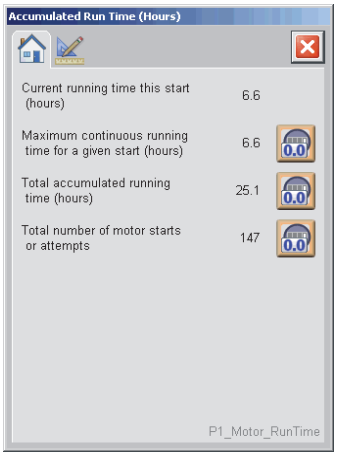
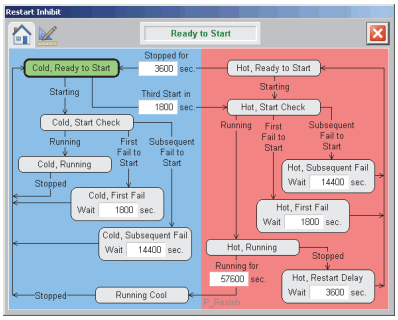
Process Object Description	Object Elements
E300™ Overload Relay (P_E300Ovld) The P_E300Ovld instruction controls and monitors a 193-ECM-ETR overload relay by using its built-in EtherNet/IP interface. The instruction reports warning and trip conditions, displays motor average current and phase currents, and provides commands to initiate a remote trip and a remote trip reset. The instruction also supports add-on options for the overload relay, including its operator interface, sensors for voltages and ground fault current, and optional discrete I/O and analog I/O modules. This instruction is provided as a rung import for installation. Click the link to access the Reference Manual: SYSLIB-RM051	Global Object  Faceplate  Add-On Instruction 
Run Time and Start Counter (P_RunTime) The P_RunTime instruction is used to accumulate the total run time and count of starts for a motor or other equipment. It is a software implementation of the mechanical hour meter that is often mounted in the door of a Motor Control Center (MCC) bucket to show total motor run time. The run time and number of starts are variables used for maintenance activities. Click the link to access the Reference Manual: SYSLIB-RM010	Global Object  Faceplate  Add-On Instruction 

Table 6 - Motors

Process Object Description	Object Elements
Restart Inhibit for Large Motor (P_ResInh) The P_ResInh instruction is used to prevent damage to a large motor through repeated starts. The instruction provides a rule-based state model for restarts and is not intended to model or monitor the motor heating. Click the link to access the Reference Manual: SYSLIB-RM009	Global Object  Add-On Instruction  Faceplate 

Notes:

Valves

The Process Objects in this group provide an interface to a wide range of process valve types as well as valve statistical calculations.

Table 7 - Valves

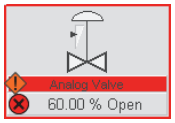
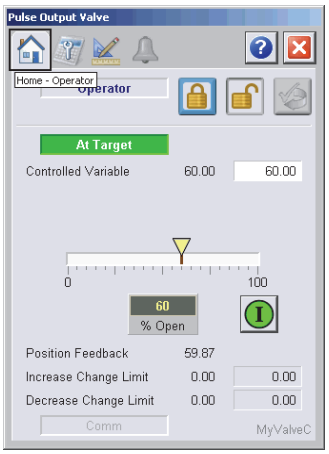
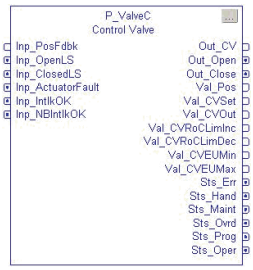
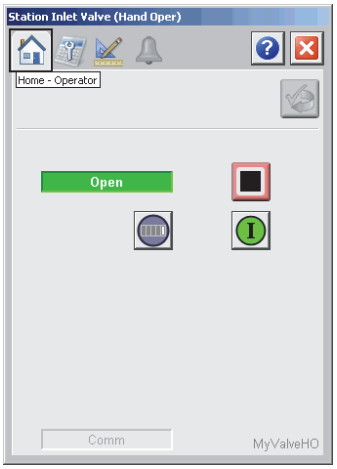
Process Object Description	Object Elements
Analog/Pulsed Control Valve (P_ValveC) The P_ValveC instruction manipulates a control valve by using an analog signal or discrete signals. The valve requires an analog output (or analog value over a network) for the target position, or the valve requires a pair of discrete outputs (or discrete signals over a network) to tell it when to move toward fully closed and when to move toward fully open. Click the link to access the Reference Manual: SYSLIB-RM034	Global Object  Faceplate  Add-On Instruction 
Hand-operated Valve (P_ValveHO) The P_ValveHO instruction monitors a hand (locally) operated valve and displays its current state. The valve can be solenoid operated, motor operated, or manually actuated. The P_ValveHO instruction cannot fully control the valve, but it can optionally provide an output to include in a trip circuit to trip the valve to a default (fail) state. Click the link to access the Reference Manual: SYSLIB-RM025	Global Object  Faceplate  Add-On Instruction 

Table 7 - Valves

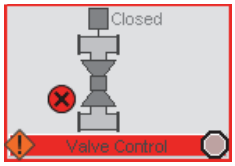
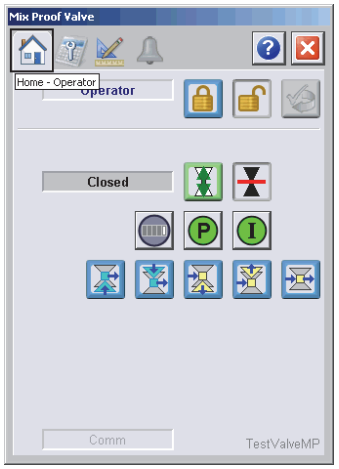
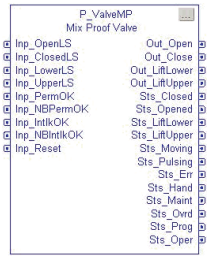
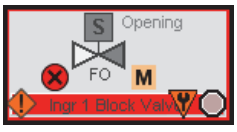
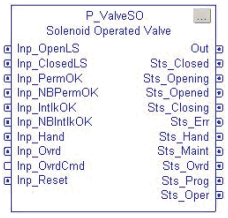
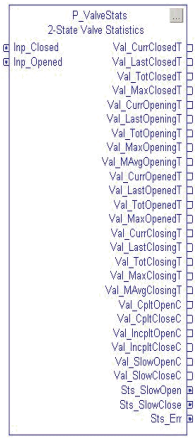
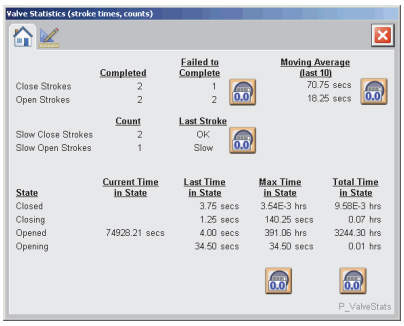
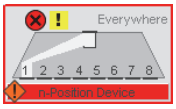
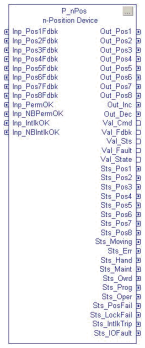
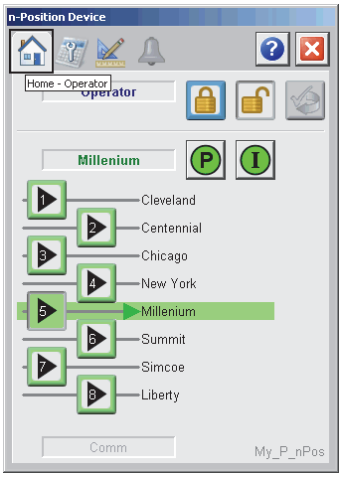
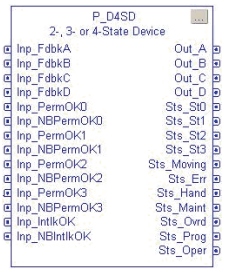
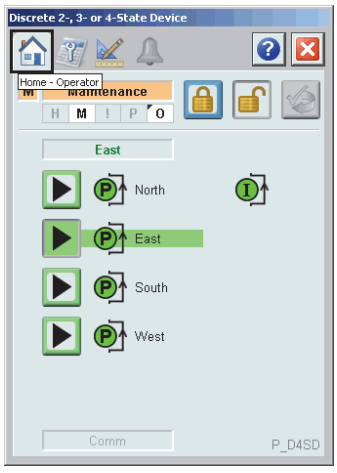
Motor-operated Valve (P_ValveMO) The P_ValveMO instruction is used to operate (open and close) a motor-operated valve in a variety of modes, monitoring for fault conditions. The valve can have, but does not require, limit switch feedback for the ends of travel. The valve can optionally use an output to trigger a 'valve stop' function, such as breaking a seal-in circuit on the valve operator to stop travel or switch the direction of travel. Click the link to access the Reference Manual: SYSLIB-RM014	Global Object  Faceplate  Add-On Instruction 
Mix-proof Valve (P_ValveMP) The P_ValveMP instruction controls one mix-proof valve in a variety of modes and states, and can verify that the valve reached the commanded position. An alarm can be provided on failure to reach a target position. This instruction supports mix-proof valves with or without additional connections for cleaning (CIP, clean in place) or steaming (SIP, sanitize in place). Click the link to access the Reference Manual: SYSLIB-RM035	Global Object  Faceplate  Add-On Instruction 
Solenoid-operated Valve (P_ValveSO) The P_ValveSO instruction is used to operate (open and close) a single solenoid operated valve in a variety of modes, monitoring for fault conditions. Use this instruction to operate a single-solenoid spring-return valve, either energize-to-open (fail closed) or energize-to-close (fail open). The valve can have, but does not require, limit switch feedback for either or both ends of travel. Click the link to access the Reference Manual: SYSLIB-RM015	Global Object  Faceplate  Add-On Instruction 

Table 7 - Valves

2-state Valve Statistics (P_ValveStats) The P_ValveStats instruction monitors a 2-state (open and close) valve and records various statistics related to stroke times and stroke counts. The instruction is designed to work with the P_ValveSO, P_ValveMO, and P_ValveHO instructions and can be used with the P_ValveMP instruction as well. Click the link to access the Reference Manual: SYSLIB-RM036	Add-On Instruction  Faceplate  Global Object 
n-Position Device (P_nPos) The P_n-Pos instruction controls a circular or linear discrete device with 2…8 positions. The instruction provides outputs to select an individual position and outputs to move toward increasing positions (‘clockwise’ for a circular device) or decreasing positions (‘counterclockwise’ for a circular device). Click the link to access the Reference Manual: SYSLIB-RM031	Global Object  Add-On Instruction  Faceplate 
Discrete 2-, 3-, or 4-state Device (P_D4SD) The P_D4SD instruction controls and monitors feedback from a discrete 2-state, 3-state, or 4-state device, including a multiple-speed motor or a multiple-position valve. The instruction controls up to four discrete outputs, with configurable states of each output in the various device states, and monitors up to four discrete feedback inputs. Click the link to access the Reference Manual: SYSLIB-RM028	Global Object  Add-On Instruction  Faceplate 

Notes:

Steam Table

The Process Objects in this group provide steam table calculations.

Table 8 - Steam Table

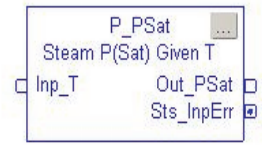
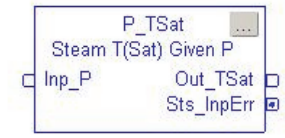
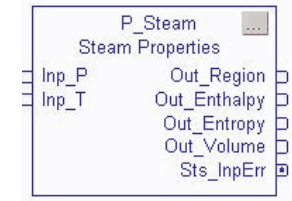
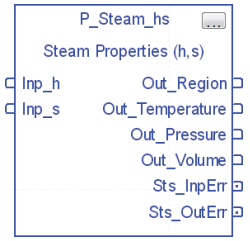
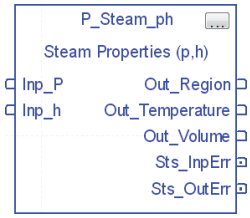
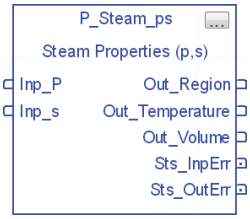
Steam Object Description	Object Elements
Saturated Steam Pressure (P_PSat) The P_PSat instruction calculates the absolute pressure (in MPa or psia) of saturated steam given the temperature (in degrees Celsius or Fahrenheit). It also provides the liquid and vapor enthalpy, entropy, and specific volume at the given temperature. Click the link to access the Reference Manual: PROCES-RM004	Add-On Instruction The P_PSat instruction is a calculation function only, and no HMI components are provided. 
Saturated Steam Temperature (P_TSat) The p_TSat instruction calculates the temperature (in degrees Celsius or Fahrenheit) of saturated steam given the absolute pressure (in MPa or psia). It also provides the liquid and vapor enthalpy, entropy, and specific volume at the given pressure. Click the link to access the Reference Manual: PROCES-RM004	Add-On Instruction The P_TSat instruction is a calculation function only, and no HMI components are provided. 
General Steam Table (P_Steam) The P_Steam instruction calculates the enthalpy, entropy, and specific volume for steam (or water) at the given pressure and temperature. Click the link to access the Reference Manual: PROCES-RM004	Add-On Instruction The P_Steam instruction is a calculation function only, and no HMI components are provided. 

Table 8 - Steam Table

Steam Object Description	Object Elements
Steam Properties Given Enthalpy and Entropy (P_Steam_hs) The P_Steam_hs (Steam Properties Given Enthalpy and Entropy) Add-On Instruction calculates the pressure, temperature, specific volume (Region 3), and vapor fraction (Region 4) at the given enthalpy and entropy. Click the link to access the Reference Manual: PROCES-RM004	The P_Steam_hs instruction is a calculation function only, and no HMI components are provided. Add-On Instruction 
Steam Properties Given Pressure and Enthalpy (P_Steam_ph) The P_Steam_ph (Steam Properties Given Pressure and Enthalpy) Add-On Instruction calculates the temperature and specific volume (Region 3) at the given pressure and enthalpy. Click the link to access the Reference Manual: PROCES-RM004	The P_Steam_ph instruction is a calculation function only, and no HMI components are provided. Add-On Instruction 
Steam Properties Given Pressure and Entropy (P_Steam_ps) The P_Steam_ps (Steam Properties Given Pressure and Entropy) Add-On Instruction calculates the temperature and specific volume (Region 3) at the given pressure and entropy. Click the link to access the Reference Manual: PROCES-RM004	The P_Steam_ps instruction is a calculation function only, and no HMI components are provided. Add-On Instruction 

Cross Functional

The Process Objects in this group are often used to extend the functionality of other objects. However, they also can be used as standalone objects when necessary to implement a desired control scheme.

Table 9 - Cross Functional

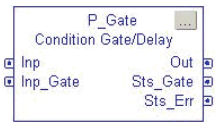
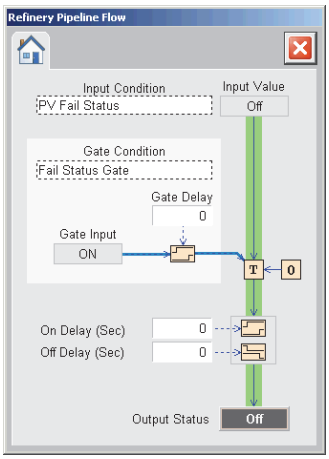
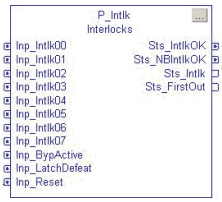
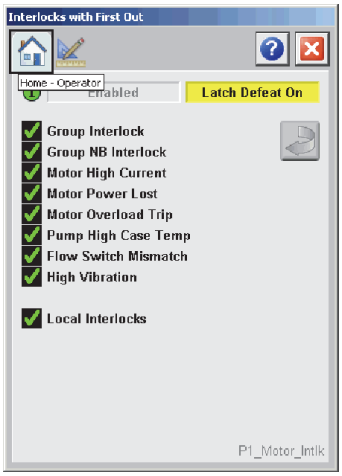
Process Object Description	Object Elements
Condition Gate Delay (P_Gate) The P_Gate instruction provides a 'gate' for a discrete signal and provides on-delay and off-delay timing for the gated signal. This instruction is used within P_DIn, all analog inputs, and P_PIDE for threshold and target disagree status processing. When the gate input is true, the input is passed through to the output by using on-delay and off-delay timing. When the gate input is false, the output is kept off. Click the link to access the Reference Manual: SYSLIB-RM041	Add-On Instruction  Faceplate 
Interlocks with First Out and Bypass (P_Intlk) The P_Intlk instruction is used to collect ('sum up') the interlock conditions that stop or de-energize a running or energized piece of equipment or prevent it from starting or being energized. Interlocks are always evaluated to de-energize equipment. Use this instruction if you want configurable text descriptions of shutdown conditions or other features of the P_Intlk faceplate. Click the link to access the Reference Manual: SYSLIB-RM004	Global Object  Add-On Instruction  Faceplate 

Table 9 - Cross Functional

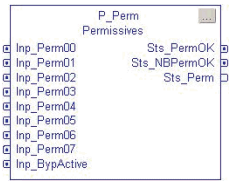
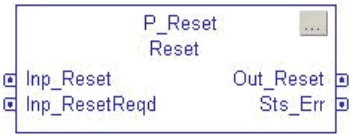
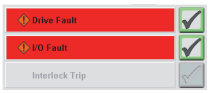
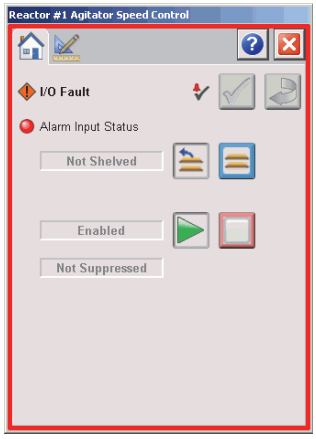
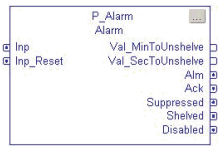
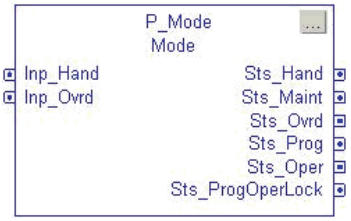
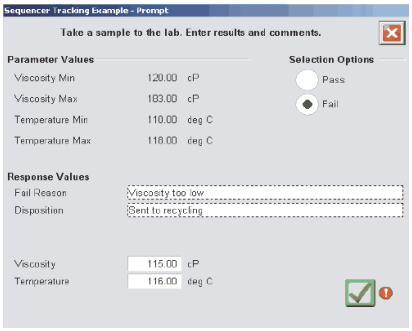
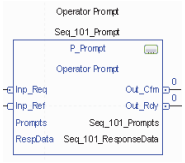
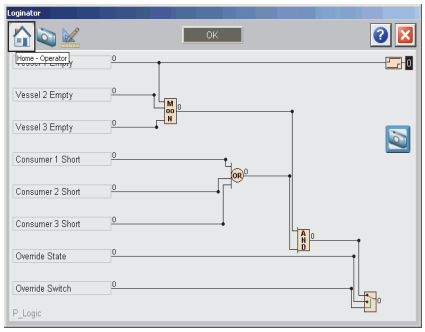
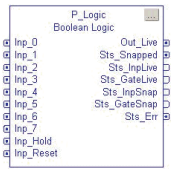
Process Object Description	Object Elements
Permissives with Bypass (P_Perm) The P_Perm instruction is used to collect ('sum up') the permissive conditions that allow a piece of equipment to start. Permissive conditions generally must be true to start the equipment. Once the equipment is running, permissives are ignored. Click the link to access the Reference Manual: SYSLIB-RM007	Global Object  Faceplate  Add-On Instruction 
Central Reset (P_Reset) The P_Reset instruction provides a central point that resets equipment faults and latched alarms for a control strategy. Use this instruction if you want a common reset point (Master Reset) for alarms and fault conditions for a control strategy, process unit, process cell or equipment group, process area or plant section, or even a small site. Click the link to access the Reference Manual: SYSLIB-RM008	Reset Button  Add-On Instruction 
Common Alarm Block (P_Alarm) The P_Alarm instruction is used to monitor an input condition, and, when it is true, raise an alarm. An operator is notified of abnormal conditions or events. This instruction handles Alarm Acknowledgement, Alarm Reset, Alarm Shelving / Disabling, and Alarm Suppression (for FactoryTalk Alarm and Events). Click the link to access the Reference Manual: SYSLIB-RM002	Alarm Display Bars  Faceplate  Add-On Instruction 

Table 9 - Cross Functional

Process Object Description	Object Elements
Common Mode Block (P_Mode) The P_Mode instruction is used to provide selection of the mode (owner) of an instruction or control strategy. Use this instruction if you are creating an Add-On Instruction for a device that requires separate acquisition by an operator and program logic, or that supports Override or Hand capabilities, or that needs a separate Maintenance mode. Click the link to access the Reference Manual: SYSLIB-RM005	Mode Totem Pole  Add-On Instruction 
Operator Prompt (P_Prompt) The P_Prompt instruction is a generic mechanism for operator interaction that can be used for any task. The instruction prompts an operator for some type of information (message or data) and accepts operator-input data and confirmation. Click the link to access the Reference Manual: SYSLIB-RM046	Global Object  Faceplate  Add-On Instruction 
Boolean Logic with Snapshot (P_Logic) The P_Logic instruction executes up to eight gates of configurable Boolean logic. Each gate provides up to four input conditions. Gate types available include AND, OR, XOR (Exclusive-OR), Set/Reset, Select, and Majority. Click the link to access the Reference Manual: SYSLIB-RM027	Global Object  Faceplate  Add-On Instruction 

Notes:

Diagnostic Objects

These objects provide diagnostic information and statistics for Logix controllers to maximize system performance.

Table 10 - Diagnostic Objects

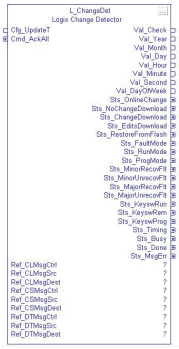
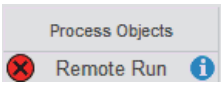
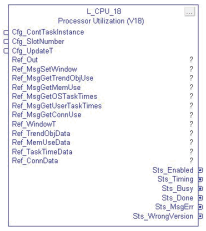
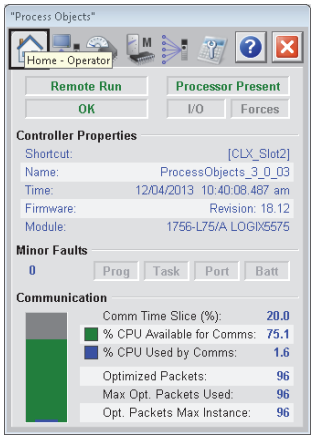
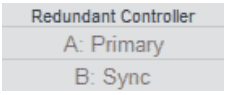
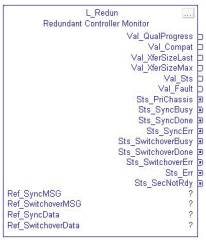
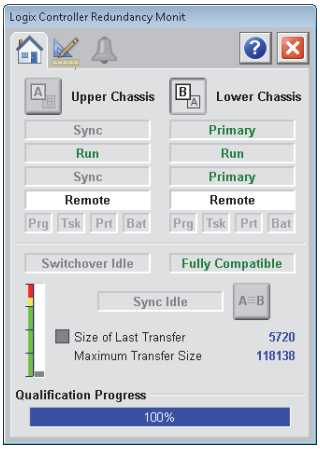
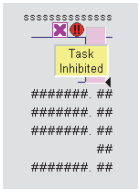
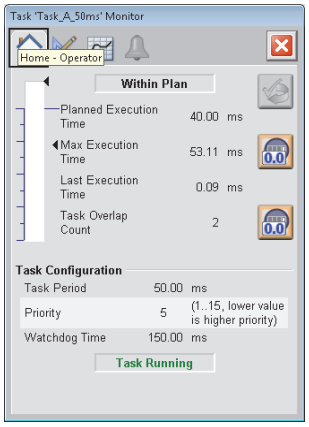
Process Object Description	Object Elements
Logix Change Detector (L_ChangeDet) The L_ChangeDet instruction monitors another Logix controller on the network, checking for changes that impact operation. Changes that can be monitored include downloads, online edits, I/O forcing, and controller mode changes. This instruction is provided as a rung import for installation. Click the link to access the Reference Manual: PROCES-RM003	Add-On Instruction No visualization elements are supplied with the L_ChangeDet instruction. 
Logix Controller CPU Utilization (L_CPU) The L_CPU instruction monitors a Logix controller, and provides information on controller CPU utilization, communication usage, memory usage, task scan times, and other information. Data provided by this instruction is useful in diagnosing communication or control responsiveness issues. This instruction is provided as a rung import for installation. Click the link to access the Reference Manual: PROCES-RM003	Global Object  Add-On Instruction  Faceplate 
Logix Redundant Controller Monitor (L_Redun) The L_Redun instruction monitors one redundant pair of Logix controllers, checking primary and secondary controller status that can impact the ability of the system to switch to the back-up controller on a failure of the primary. This instruction is provided as a rung import for installation. Click the link to access the Reference Manual: PROCES-RM003	Global Object  Add-On Instruction  Faceplate 

Table 10 - Diagnostic Objects

Process Object Description	Object Elements
Logix Task Monitor (L_TaskMon) The L_TaskMon instruction monitors one task running in a Logix controller to provide task statistics, such as task scan time and overlap count. This instruction also provides task configuration settings, task 'plan' execution time, and alarm if the planned execution time is exceeded. Click the link to access the Reference Manual: PROCES-RM003	Global Object  Add-On Instruction  Faceplate 

64-Bit Math

See Long Integer and Time Instructions on page 165 to complete math and time functions with library objects.

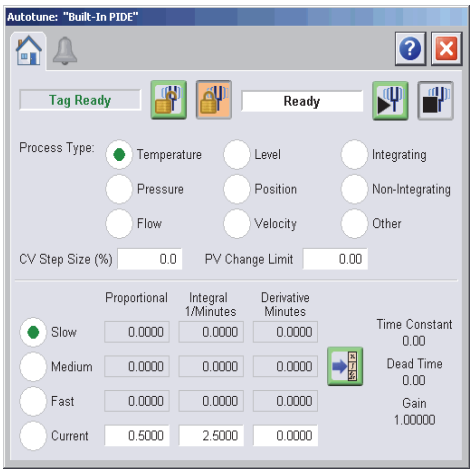
Time and Date Math

See the Time and Date Instructions on page 169 to complete date and time functions with library objects,

Graphics for Built-in Instructions

The faceplates in this section are designed so the built-in Logix5000 controller instructions can interface with the Process Library Add-On Instructions.

Table 11 - Built-in Autotuner

Process Object Description	Object Elements
Built-in Autotuner The RSLogix™ 5000 PIDE autotuner provides an open-loop autotuner that is built into the PIDE instruction. The autotune function is accessed from the PIDE faceplate. You can autotune from PanelView terminals or any other operator interface devices as well as RSLogix 5000 software. The PIDE block has an Autotune Tag (type PIDE_AUTOTUNE) that you specify for those PIDE blocks that you want to autotune. Click the links to access: • faceplates in Appendix F • Reference Manual, publication 1756-RM006	Global Object  – – – The autotuner is only supported in function block programming; it is not available in relay ladder or structured text programming. Faceplate 

Advanced Process Control Function Blocks (CC, IMC, MMC)

The three advanced process control (APC) function blocks can be used in place of PID instructions for loops with long dead-times and interacting loops.

Table 12 - APC Function Blocks

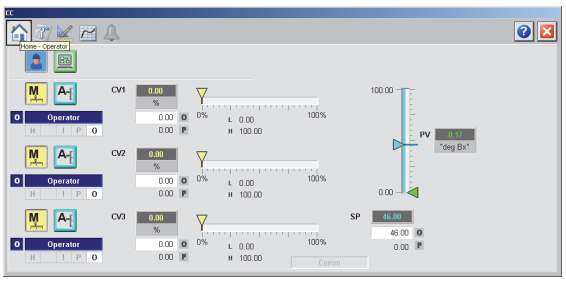
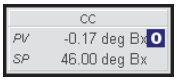
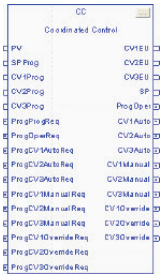
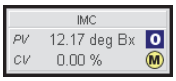
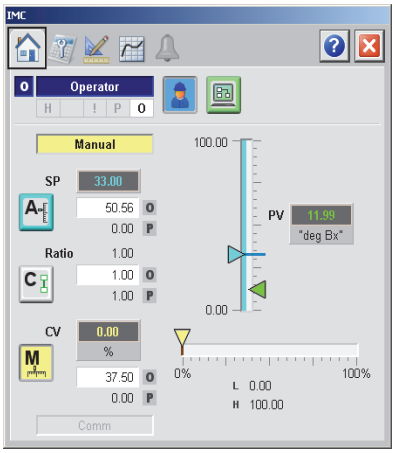
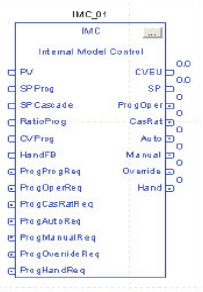
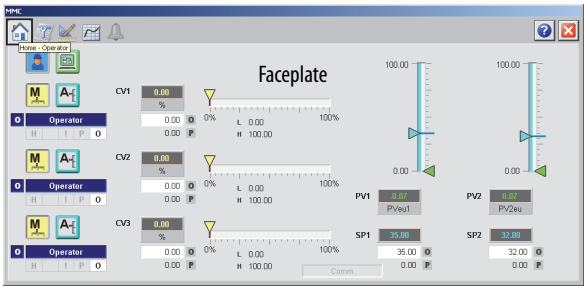
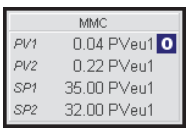
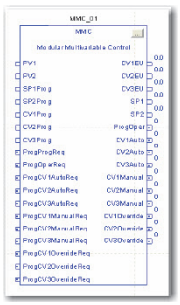
Process Object Description	Object Elements
Coordinated Control (CC) The Coordinated Control (CC) function block controls a single process variable by manipulating as many as three different control variables. As an option, any of the three outputs can be used as an input to create feed forward action in the control variable. The CC function block calculates the control variables (CV1, CV2, and CV3) in the Auto mode based on the PV - SP deviation, internal models, and tuning. The CC function block is a flexible model-based algorithm that can be used in various configurations. Click the links to access: • faceplates in Appendix F • Reference Manual, publication 1756-RM006	Faceplate  Global Object  Add-On Instruction 

Table 12 - APC Function Blocks

Process Object Description	Object Elements
Internal Model Control (IMC) The IMC function block controls a single process variable by manipulating a single control-variable output. This function block performs an algorithm where the actual error signal is compared against that of an internal first-order lag plus deadtime model of the process. The IMC function block calculates the control variable output (CV) in the Auto mode based on the PV - SP deviation, internal model, and tuning. Click the links to access: • faceplates in Appendix F • Reference Manual, publication 1756-RM006	Global Object  Faceplate  Add-On Instruction 
Modular Multivariable Control (MMC) The Modular Multivariable control (MMC) function block controls two process variables to their setpoints by manipulating up to three control variables. The MMC function block calculates the control variables (CV1, CV2, and CV3) in the Auto mode based on the PV1 - SP1, PV2 - SP2 deviation, internal model, and tuning. The MMC function block is a flexible model-based algorithm that can be used in two basic configuration modes: • Three control variables used to control two interacting process variables • Two control variables used to control two interacting process variables Click the links to access: • faceplates in Appendix F • Reference Manual, publication 1756-RM006	 Global Object  Add-On Instruction 

Advanced Process Control Instructions (PIDE, RMPS, TOT)

These APC instructions regulate control of process variables and values.

Table 13 - APC Instructions

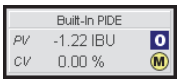
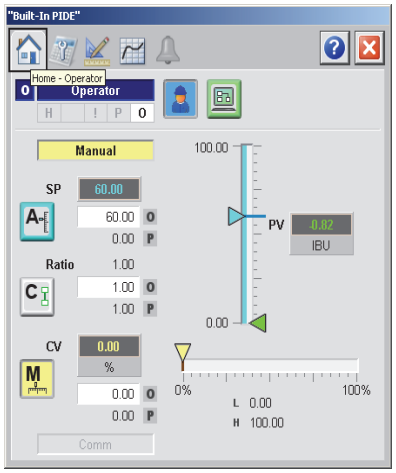
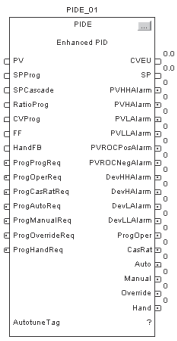
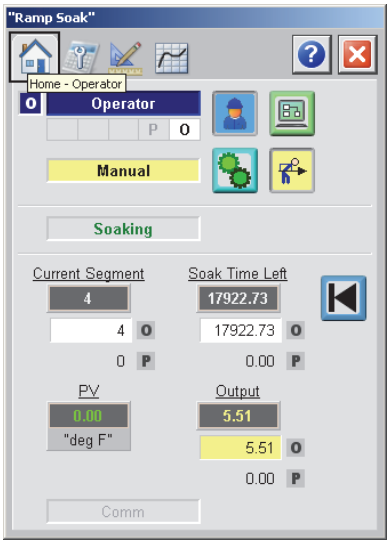
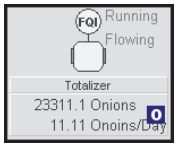
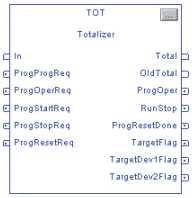
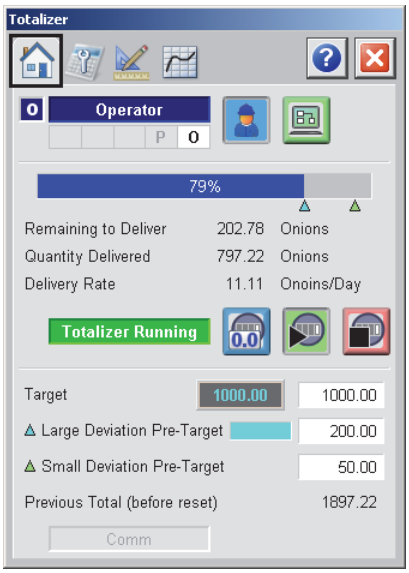
Process Object Description	Object Elements
Enhanced PID (PIDE) The PIDE instruction provides enhanced capabilities over the standard PID instruction. The instruction uses the velocity form of the PID algorithm. The gain terms are applied to the change in the value of error or PV, not the value of error or PV. The PIDE instruction uses a velocity form PID algorithm similar to that used in most DCS systems. An advantage to a velocity form algorithm includes a bumpless adaptive gain change – changing gains on the fly without initializing the algorithm. Click the links to access: • faceplates in Appendix F • Reference Manual, publication 1756-RM006	Global Object  Faceplate  Add-On Instruction 
Ramp/Soak (RMPS) The RMPS instruction provides for a number of segments of alternating ramp and soak periods. The RMPS instruction is typically used to provide a temperature profile in a batch heating process. The output of this instruction is typically the input to the setpoint of a PID loop. The RMPS instruction can be controlled by either Program mode or Operator mode. Control can be changed any time. Click the links to access: • faceplates in Appendix F • Reference Manual, publication 1756-RM006	Global Object  Faceplate  Add-On Instruction 

Table 13 - APC Instructions

Process Object Description	Object Elements
Totalizer (TOT) This instruction typically totals the amount of a material added over time, based on a flow signal. Support for the TOT instruction includes the following: • Time-base selectable as seconds, minutes, hours, or days. • Specify a target value and up to two pre-target values. Pre-target values are typically used to switch to a slower feed rate. Digital flags show the reaching of the target or pre-target values. • Low flow input cutoff that eliminates negative totalization due to slight flowmeter calibration inaccuracies when the flow is shut off. • Operator or program capability to start/stop/reset. Click the links to access: • faceplates in Appendix F • Reference Manual, publication 1756-RM006	Global Object  Add-On Instruction  Faceplate 

Standard Symbols and Indicators

The Library of Process Objects uses a standard set of symbols and indicators across its HMI objects. The following section illustrates these items.

Table 14 - Tab Navigation Icons

Graphic Symbol	Description
	Operator (Home) Page
	Maintenance Page
	Engineering Page
	Trends Page
	Diagnostics Page
	Alarms Page
	Alarms Page (with active alarm)
	Snapshot Page

Table 15 - Breadcrumbs

Graphic Symbol	Description
	Invalid Configuration
	Information Available
	A Maintenance Bypass is active
	Operator Attention
	Maintenance Required

Table 16 - Mode Symbols and Indicators

Graphic Symbol	Description
	No Mode (out of service)
	Hand (Local)
	Device in Maintenance mode
	Override
	Device locked in Program mode
	Device locked in Operator mode
	Device in Program mode
	Device in Operator mode

Table 17 - Status Symbols

Graphic Symbol	Description
	Device has been disabled
	At target speed
	Communication failure
	Accelerating
	Decelerating
	Input or PV uncertain
	The device is not ready to operate
	value is being initialized
	Input has been disabled
	value has not changed (stuck)

Table 17 - Status Symbols

Graphic Symbol	Description
	Input has been rejected
	Value infinite or not a number
	Input CV clamped to minimum/maximum
	Output CV clamped to minimum/maximum (information)
	Output CV clamped to minimum/maximum
	Speed reference limited to the minimum/maximum
	Value clamped to minimum/maximum
	Value is being held at the last good value
	Value is being replaced
	Input matches target
	Input does not match target

Table 18 - Alarm Symbols

Graphic Symbol	Description
	Urgent
	High
	Medium
	Low
	Out of alarm - Acknowledge required
	Alarm inhibit (suppressed, shelved, or disabled)
	In alarm (alarm active)

Table 18 - Alarm Symbols

Graphic Symbol	Description
	In alarm and acknowledged
	Out of alarm but not acknowledged
	Alarm suppressed (by Program logic)
	Alarm disabled (by Maintenance)
	Alarm shelved (by Operator)

Table 19 - Interlock and Permissive Indicators

Graphic Symbol		Description
		One or more conditions not OK
		Non-bypassed conditions OK
		All conditions OK, bypass active
		All conditions OK
		Enable checking all interlock and permissive conditions
		Bypass interlocks and permissives that can be bypassed

Table 20 - Level and Deviation Threshold Indicators

Graphic Symbol	Description
	High-high level exceeded
	High level exceeded
	Low level exceeded
	Low-low level exceeded

Table 20 - Level and Deviation Threshold Indicators

Graphic Symbol	Description
	High rate of change exceeded
	High-high deviation exceeded
	High deviation exceeded
	Low deviation exceeded
	Low-low deviation exceeded

Table 21 - PID Symbols

Graphic Symbol	Description
	Manual loop mode
	Auto loop mode
	Auto loop mode (cascade enabled)
	Cascade loop mode
	PV within SP deadband (no control action occurs)
	The CV has reached a high limit and cannot control the loop
	The CV has reached a low limit and cannot control the loop

Standard Buttons

The Library of Process Objects uses a standard set of buttons across its HMI objects. The following section illustrates these items.

Table 22 - Enable and Disable Buttons

Button	Description	Button	Description
	Enable Device		Disable Device

Table 23 - Alarm Buttons

Button	Description	Button	Description
	Acknowledge Alarm		Acknowledge and Reset all alarms for an object
	Shelve Alarm		Unshelve Alarm
	Enable Alarm		Disable Alarm

Table 24 - Mode Buttons

Button	Description	Button	Description
	Lock Operator Mode		Unlock Operator Mode
	Request Operator Mode		Request Program Mode
	Request Maintenance Mode		Release Maintenance Mode
	Navigate to Mode Configuration Display		

Table 25 - PID Buttons

Button	Description	Button	Description
	Request Cascade Loop mode		Request Auto Loop mode
	Request Manual Loop mode		Request 'Normal' Loop mode

Table 26 - Miscellaneous Command Buttons

Button	Description	Button	Description
	Move to state		Move to position
	Clear Counter		Capture snapshot. Captures the current state of the object.

Table 27 - Mix-proof Valve Buttons

Button	Description	Button	Description
	Open Valve		Close Valve
	Lift Valve Lower Seat		Lift Valve Upper Seat
	SIP/CIP Valve Lower Seat		SIP/CIP Valve Upper Seat
	SIP/CIP Valve Cavity		

Table 28 - Motor and Drive Buttons

Button	Description	Button	Description
	Start		Stop
	Run Motor Forward		Run Motor Reverse
	Run Motor at Slow Speed		Run Motor at Fast Speed
	Operator Command to Trip Motor		Jog
	Request Reverse Motion		Request Forward Motion
	Restart Inhibit Navigation Button		Runtime Accumulator Navigation Button
	Overload Navigation Button		

Table 29 - Valve Buttons

Button	Description	Button	Description
	Open Valve		Close Valve
	Stop Valve Motion		Operator Command to Trip Valve
	Bump Valve Close		Bump Valve Open
	Valve Stats Navigation Button		

Table 30 - Overload Buttons

Button	Description	Button	Description
	Operator Command to Trip Overload		Operator Command to Reset the Overload Trip

Table 31 - Digital Output Buttons

Button	Description	Button	Description
	Output ON		Output OFF
	Pulse Output ON (Once)		Pulse Output OFF (Once)
	Continuous Pulse Output		

Table 32 - Analog Input Buttons

Button	Description	Button	Description
	Use Input PV		Use Substitute PV
	Select Sensor A Input PV		Select Sensor B Input PV
	Select the Maximum of Sensor A and Sensor B Input PV		Select the Average of Sensor A and Sensor B Input PV
	Select the Minimum of Sensor A and Sensor B Input PV		Operator Command to Reset Min and Max capture Values

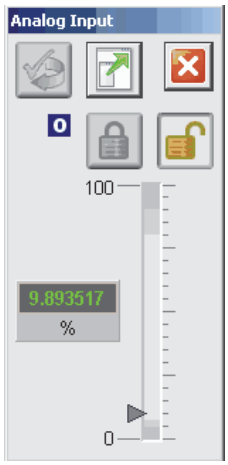
Table 33 - Dosing Buttons

Button	Description	Button	Description
	Clear Totalizer		Tar Scale
	Start Totalizer		Stop Totalizer
	Start Flow		Stop Flow
	Bump Flow		Check Tolerances

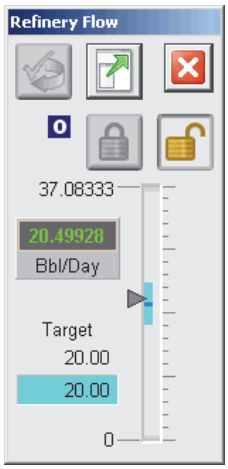
Quick Displays

Quick displays provide means for operators to perform simple interactions with Add-On Instruction instances. From the Quick Display, you can navigate to the faceplate for full access for operation, maintenance, and configuration.

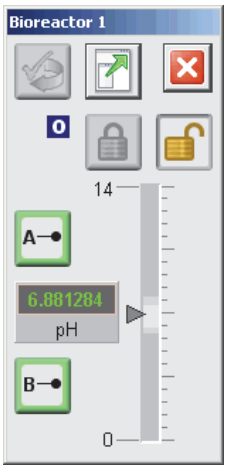
Basic Analog Input
(P_AIn)



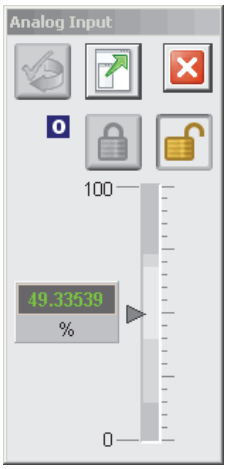
Advanced Analog Input
(P_AInAdv)



Dual Sensor Analog Input
(P_AInDual)



Multiple Analog Input
(P_AInMulti)



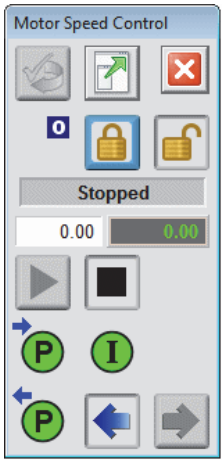
Single-speed Motor
(P_Motor)

Two-speed Motor
(P_Motor2Spd)

Hand-operated Motor
(P_MotorHO)

Reversing Motor
(P_MotorRev)

PowerFlex 523/525 Drive
(P_PF52x)

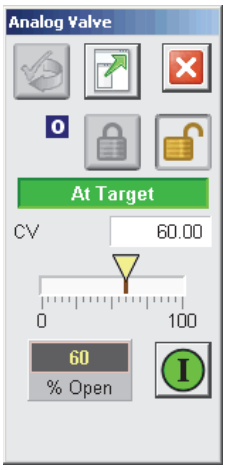


PowerFlex 753 Drive
(P_PF753)

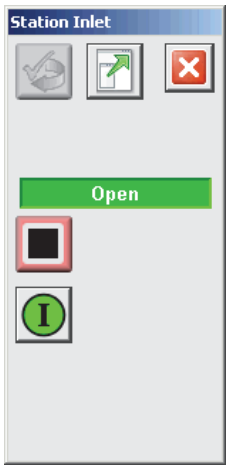
PowerFlex 755 Drive
(P_PF755)

Variable-speed Drive
(P_VSD)

Analog/Pulsed Control Valve
(P_ValveC)

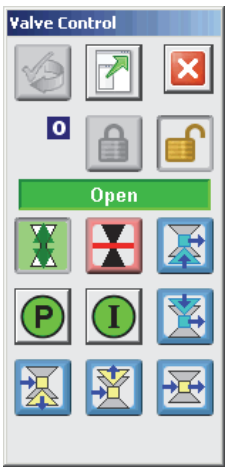


Hand-operated Valve
(P_ValveHO)



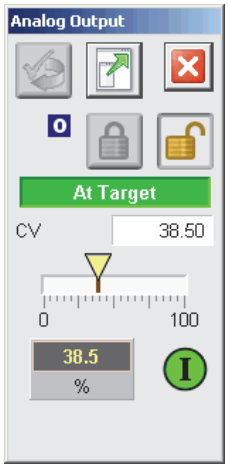
Motor-operated Valve
(P_ValveMO)

Mix-proof Valve
(P_Valve MP)

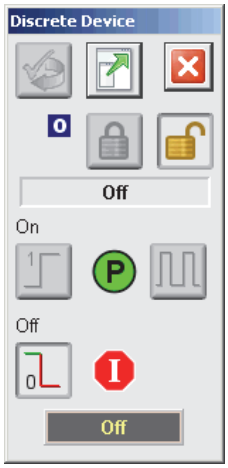


Solenoid-operated Valve
(P_ValveSO)

Analog Output
(P_AOut)

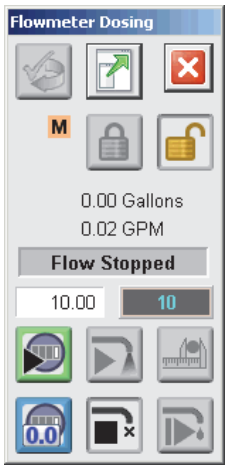


Discrete Output
(P_DOut)

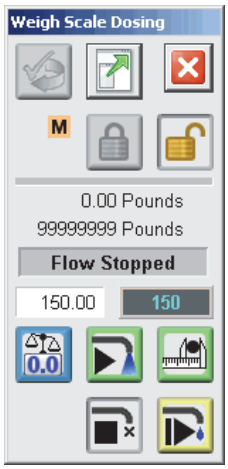


Discrete 2-, 3-, 4-state Device
(P_D4SD)

Flowmeter Dosing
(P_DoseFM)

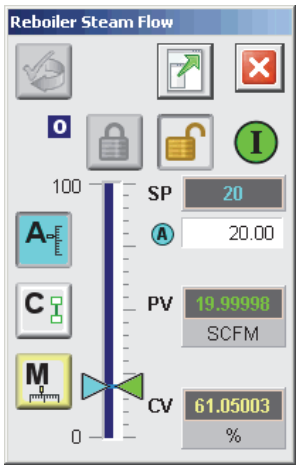


Weigh Scale Dosing
(P_DoseWS)



n-Position Device
(P_nPos)

Proportional + Integral + Derivative Enhanced (P_PIDE)

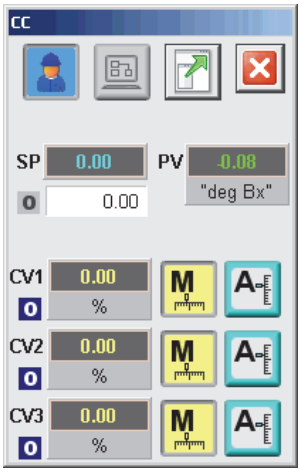


Sequencer Object (P_Seq)

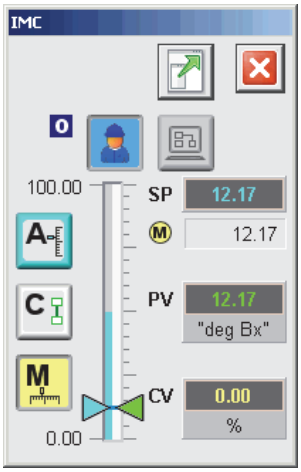
Built-in Quick Displays

The following Quick Displays provide access to faceplates that are designed to work with built-in instructions.

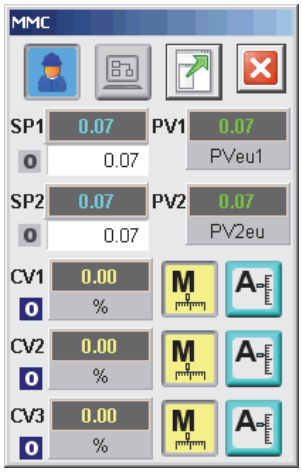
Coordinated Control (CC)



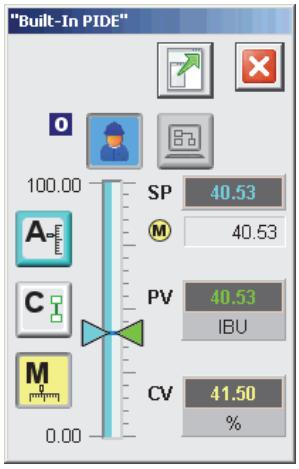
Internal Model Control (IMC)



Modular Multivariable Control (MMC)



Proportional + Integral + Derivative (PIDE)



How to Install the Library

This chapter describes procedures for installing the HMI and controller elements that comprise the Rockwell Automation Library of Process Objects. There are two methods:

- Using predefined application templates
- Importing individual library objects

If you are starting a new project, it is recommended that you begin from our Quick Start application templates. The Quick Start manual guides you through the necessary steps to install the library components, and provides guidance on configuring basic system diagnostics.

For details, see the PlantPAx System Application Templates Quick Start, publication PROCES-QS001.

The table describes the topics in this chapter.

Download the Library

For the latest compatible software information and to download the Rockwell Automation Library, see the Product Compatibility and Download Center at http://www.rockwellautomation.com/rockwellautomation/support/pcdc.page.

IMPORTANT To download virtual templates, see the Virtual Image Templates User Manual, publication, 9528-UM001.

Using HMI and Controller Templates

There is a Templates folder within the Rockwell Automation Library download. This folder has HMI and controller templates for creating a PlantPAx system.

However, the HMI templates do **not** include Library objects. You must import the Library objects into the FactoryTalk View SE and ME templates, respectively.

Contrarily, some of the controller templates have pre-loaded library content, if the Library is to be used with your application.

For more information, see the following:

- PlantPAx System Application Templates Quick Start, publication PROCES-QS001
- Importing visualization files on page 72

IMPORTANT	You can use library objects other than the Rockwell Automation Library objects. However, we suggest that you follow the instructions in this section to make sure of properly configuring the visualization files in your project.

Adding Controller Logic

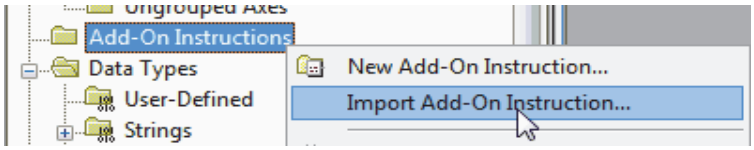
An Add-On Instruction is defined once in each controller project, and can be instantiated multiple times in your application code. To use the Add-On Instructions, you import them into a controller project.

Do these steps for each Add-On Instruction.

1. In RSLogix 5000 software, open a new or existing project.

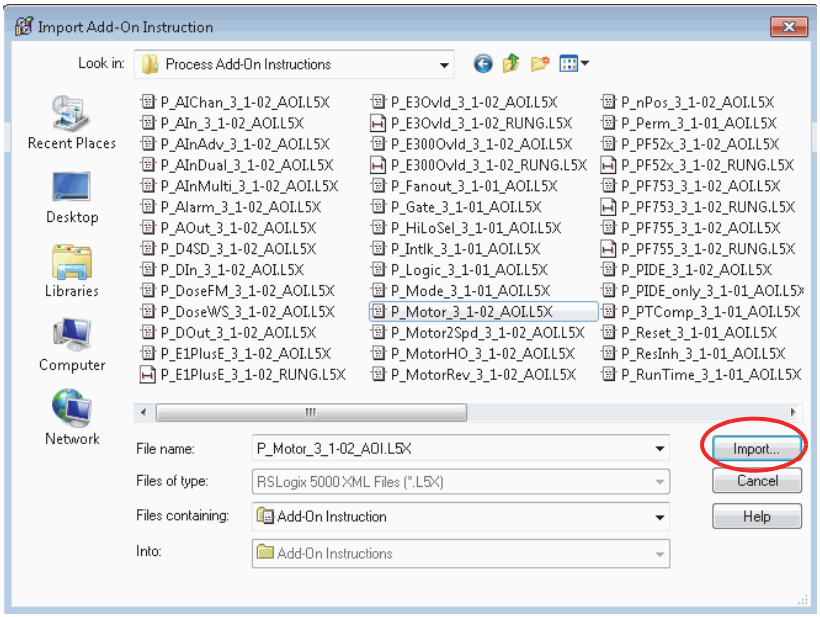
IMPORTANT	Add-On Instruction definitions can be imported, but not updated, online.

2. Right-click the Add-On Instructions folder in the Controller Organizer and choose Import Add-On Instruction.



The Import Add-On Instruction dialog box appears.

3. Select the Add-On Instruction and click Import.

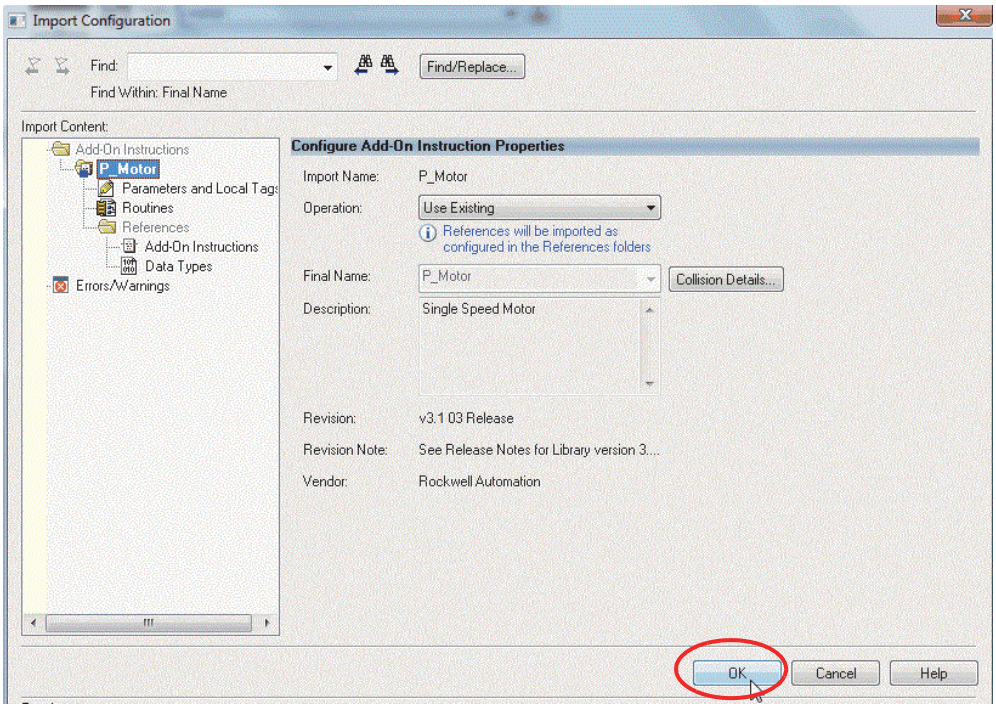


TIP The P_Mode, P_Alarm, and P_Gate Add-On Instructions are used within many of the other instructions. We recommend that you import these three instructions first.

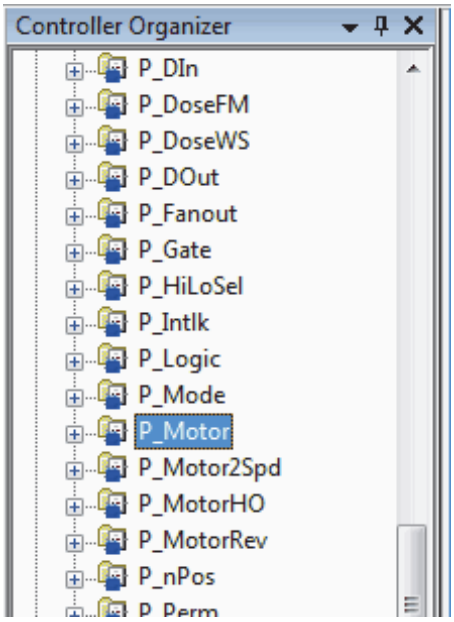
Some Add-On Instructions are provided in RUNG import files.

TIP If a RUNG import file is provided, import the rung into a Ladder Diagram routine to get all the required additional tags, data types, and Message configurations.

4. On the Import Configuration dialog box, click OK.



Once the import is complete, the Add-On Instructions are visible in the Controller Organizer.

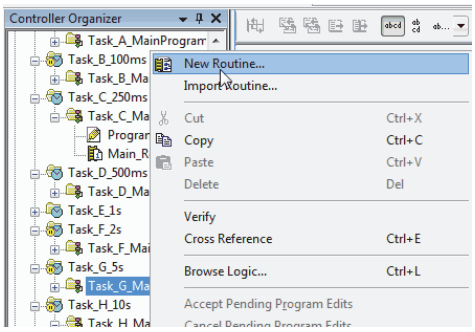


Add an Add-On Instruction to a Routine

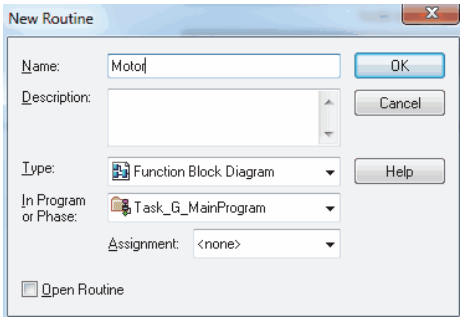
Do these steps to add an Add-On Instruction to a routine, create a backing tag, and connect I/O.

Add-On Instructions can be used in any of the Logix languages: Ladder Diagram, Function Block Diagram, or Structured Text. In this example, we show how to add an instruction instance to a Function Block Diagram routine.

1. In the Controller Organizer, right-click the Task and choose New Routine.



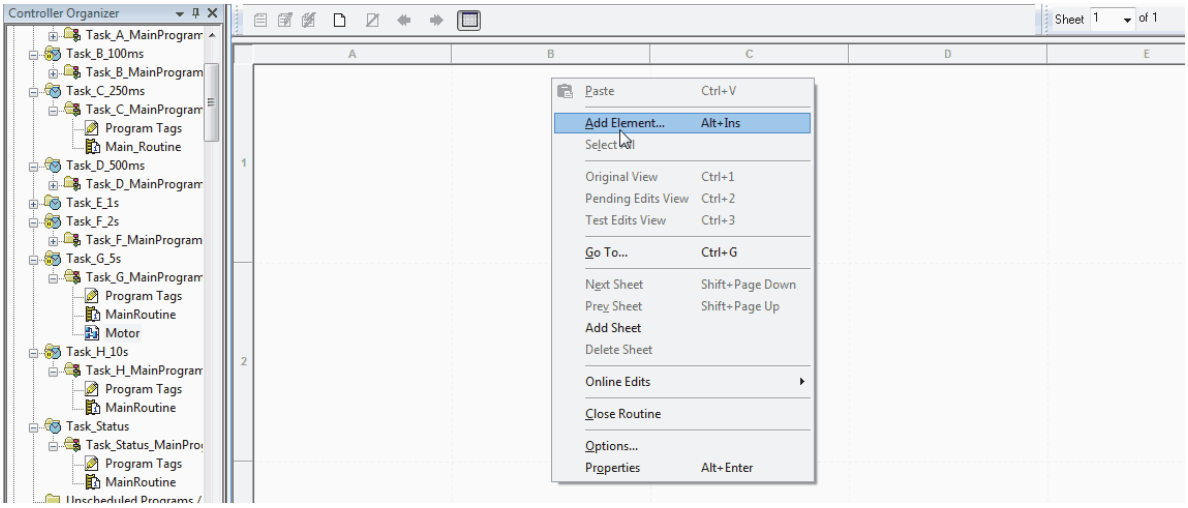
The New Routine dialog box appears.



2. Type a name for the routine.
3. Click the Type pull-down to select a Logic language, such as Function Block Diagram, and click OK.
4. Double-click the routine name in the Controller Organizer

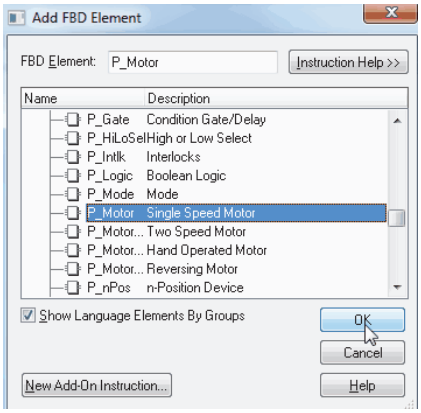
   A blank sheet appears in the right pane.

5. Right-click the blank sheet and choose Add Element.



The Add FBD Element window appears.

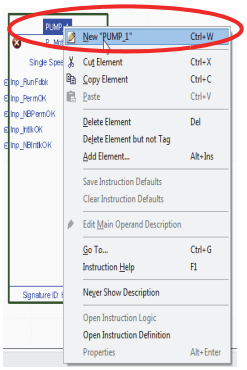
6. Browse to the Add-On Instruction folder, select the Add-On Instruction, and click OK.



A function block (or the language type you selected) appears.

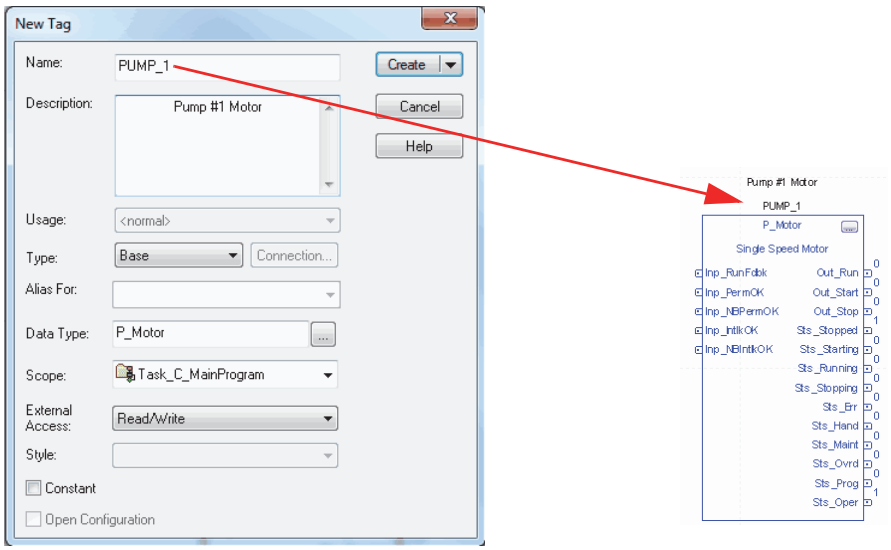
7. Double-click the name and type a new name, for example Pump_1.

8. Right-click the new name and choose New 'name' (Pump_1 is our example).



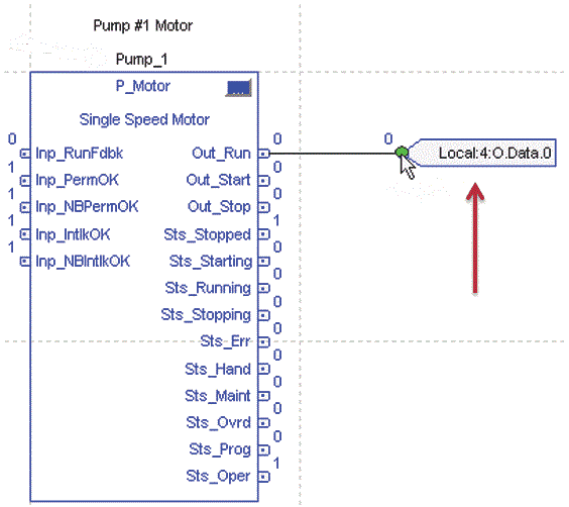
The New Tag dialog box appears.

9. Type a description for your new backing tag and click Create.



The tag's description and name appear at the top of the function block.

10. Connect the pins to add I/O.



11. Repeat this process for each required Add-On Instruction.

Import Visualization Files

Each Add-On Instruction has associated visualization files that provide a common user interface. You must import these files in the following order:

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

A global object is an HMI display element that is created once and referenced multiple times on multiple displays in an application. When changes are made to the original (base) object, the instantiated copies (reference objects) are automatically updated. Use of global objects, in conjunction with tag structures in the ControlLogix® system, aid in consistency and save engineering time.

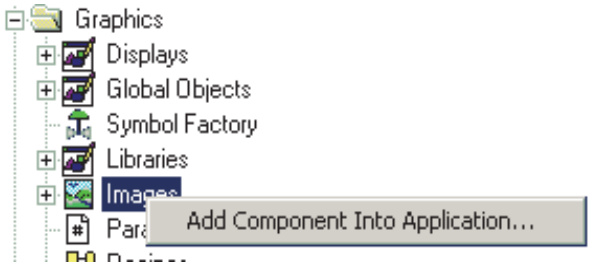
IMPORTANT If you are using FactoryTalk View ME software, you also must import the HMI tags to support switching tabs on any Process Object faceplate.
See FactoryTalk View ME Configuration on page 77 for procedures.

The import procedures in this section are to be followed in the **exact order** as documented to add the visualization files to your project.

Import Images

Do these steps to use the common icons for the global objects and faceplates for all Process objects.

1. In your FactoryTalk View SE or ME software program (depending on which one you are using), click the '+' to open the Graphics folder.

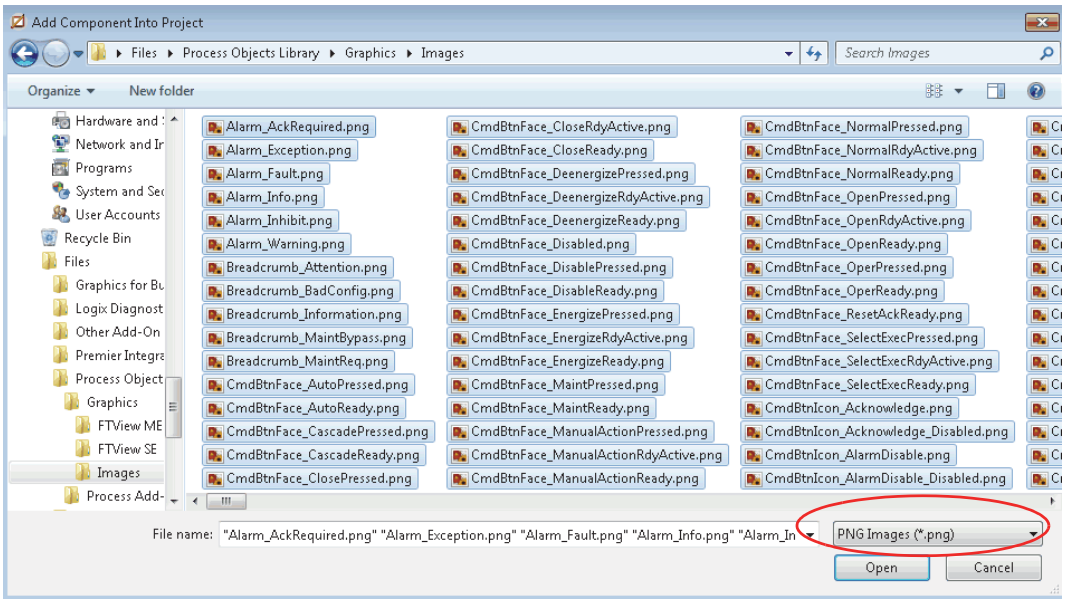


2. Right-click Images and choose Add Component Into Application.

   The Add Component Into Project dialog box appears.

3. Browse to your downloaded Rockwell Automation library files.
4. Click the graphics folder.

5. Click the Images folder.



IMPORTANT You need to change the path to the image folder and the file type to PNG. PNG files provide more control with transparency.

6. Click the pull-down menu (as circled) and select a file type.

   For example, PNG Images (*.png)

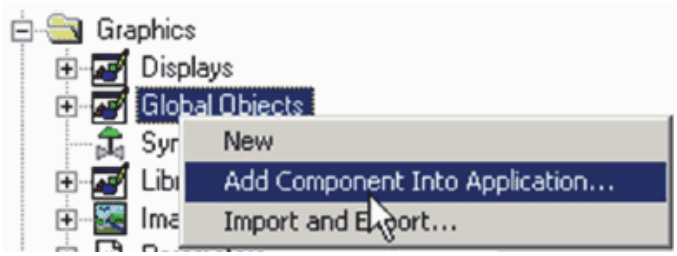
7. Click Ctrl-A to highlight all of the .png files.
8. Click Open to import the images.

Import Global Object Files

Global objects serve two purposes:

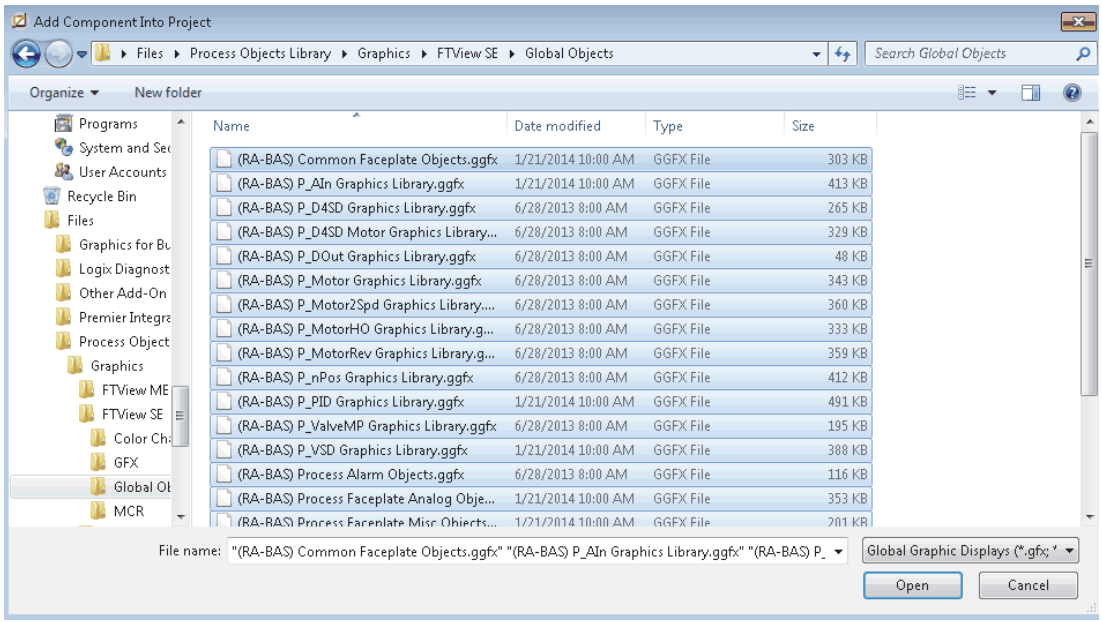
- Faceplate objects files contain common elements that are used in building faceplate displays.
- Graphics Library files contain device symbols that you can use to build your application displays. Clicking the symbol opens the corresponding faceplate display.

1. Right-click Global Objects and choose Add Component Into Application.



   The Add Component Into Project dialog box appears.

2. Browse to your downloaded Rockwell Automation Library files.
3. Click the Graphics folder.
4. Click the FactoryTalk View SE or FactoryTalk View ME folder depending on your application.
5. Click the Global Objects folder.

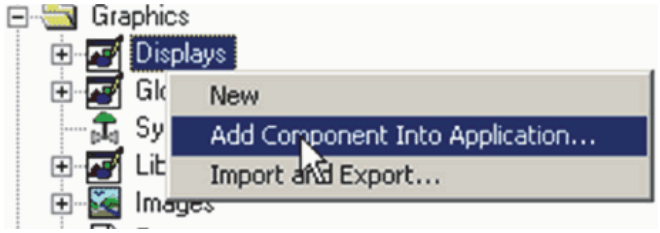


6. Click Ctrl-A to highlight all the global object (.ggfx) files.
7. Click Open to import the objects.

Import HMI Faceplates

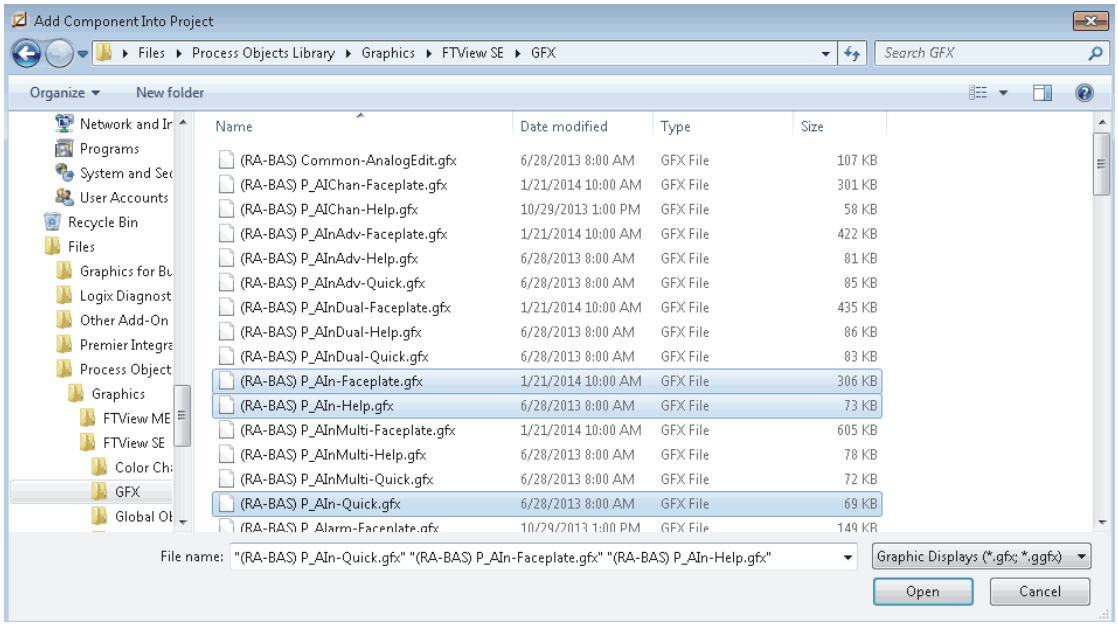
Faceplates provide operators, maintenance workers, engineers, and others to interact with instrument data. Do these steps to import faceplates.

1. Right-click Displays and choose Add Component Into Application.



   The Add Component Into Project dialog box appears.

2. Browse to your downloaded Rockwell Automation library files.
3. Click the graphics folder.
4. Click the FactoryTalk View SE or FactoryTalk View ME folders depending on your application.
5. Click the GFX folder.



6. Click only displays that you need; do not import all of them.

   The highlighted example with P_AIn shows that each Add-On Instruction requires a Faceplate.gfx, Help.gfx, and Quick.gfx. Most motors, valves, and other devices need displays for Mode Configuration and Help and Alarm Configuration and Help.

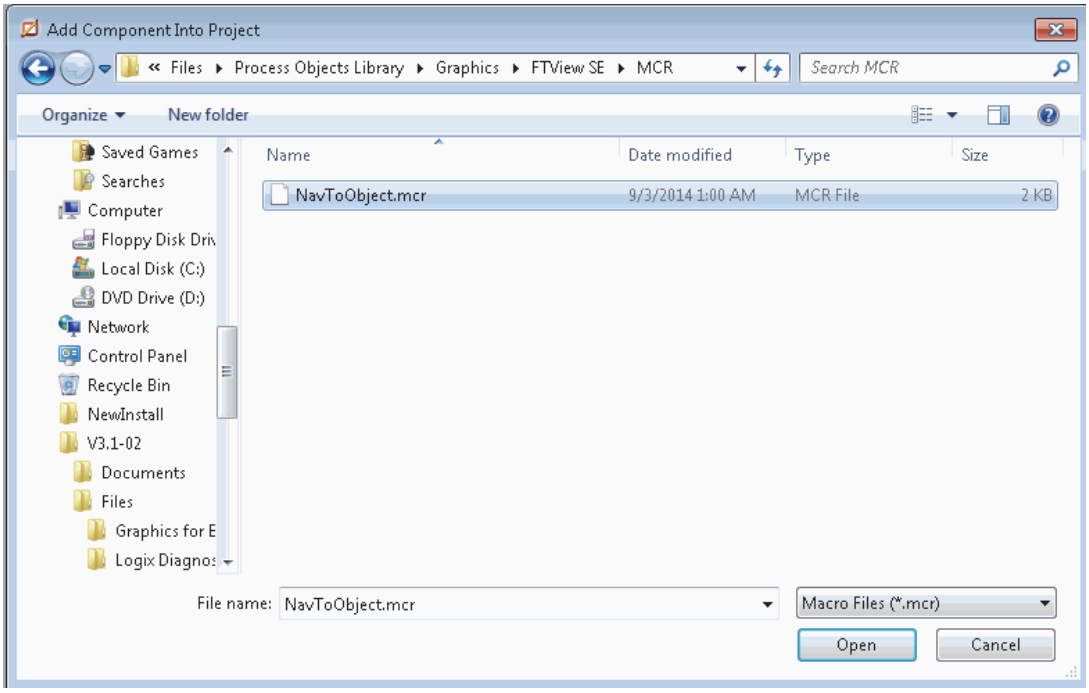
   See the Reference Manual for the respective Add-On Instruction for a list of required .gfx files.

7. For FactoryTalk View SE applications, also import the Common Analog Edit display.

Import the Macro

These instructions are for FactoryTalk View SE projects only. A macro must be imported to support faceplate-to-faceplate navigation by tag name.

1. Right-click Macro and select Add Component Into Application.

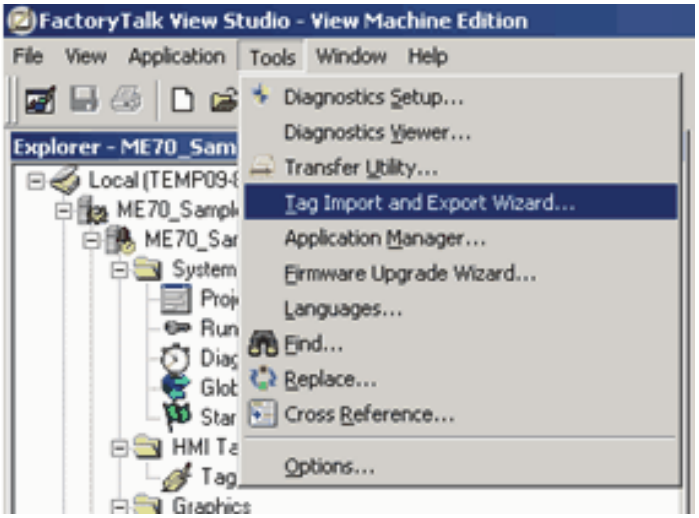


2. Select the NavToObject.mcr file and click Open.

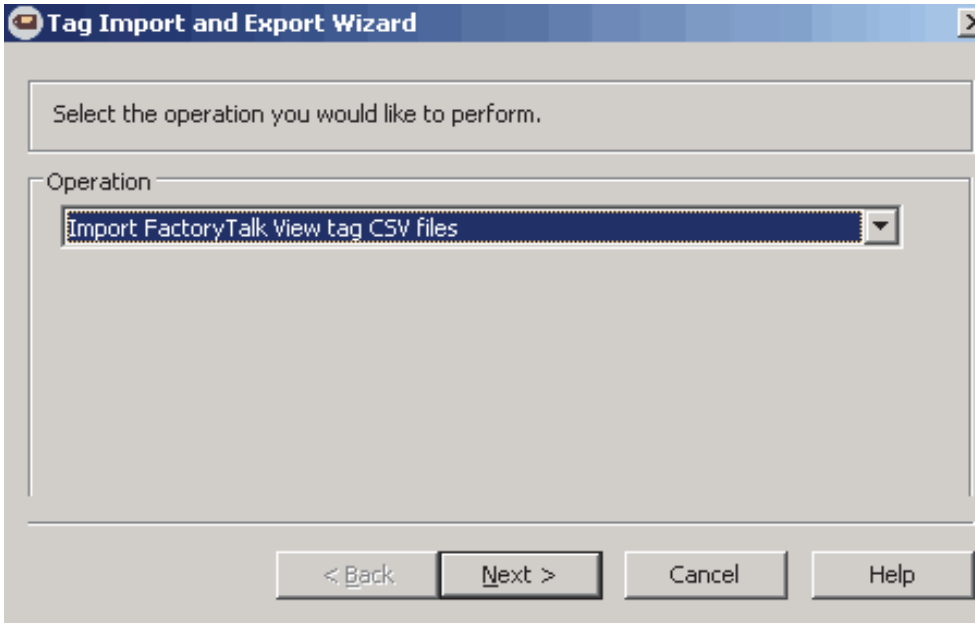
FactoryTalk View ME Configuration

For FactoryTalk View ME configurations **only**, follow these steps to import HMI tags so you can switch between tabs on the faceplates.

1. From the Tools pull-down menu, choose Tag Import and Export Wizard.

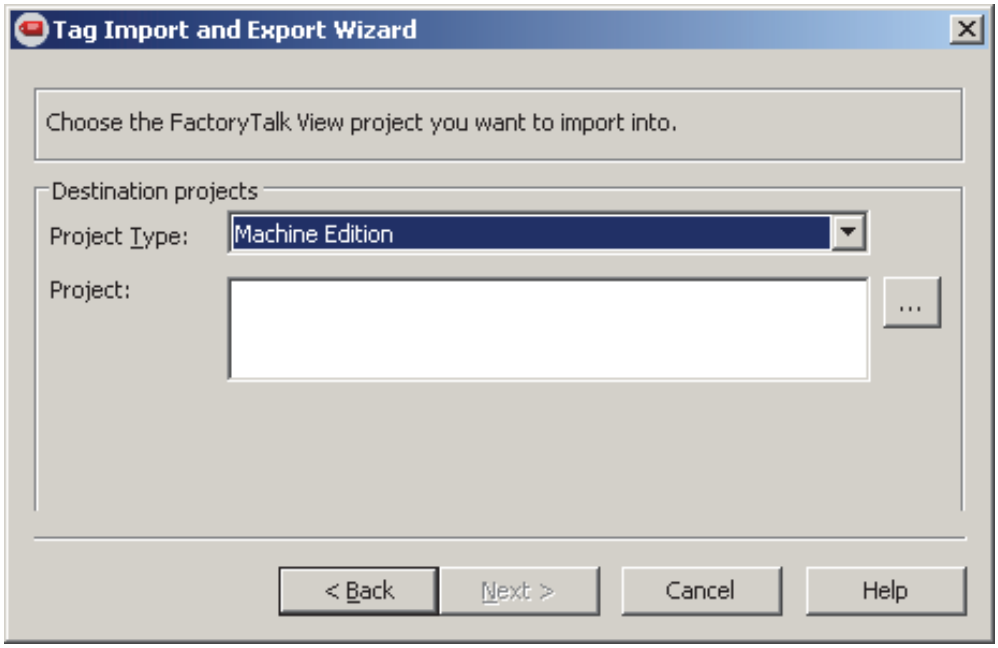


The Tag Import and Export Wizard dialog box appears.



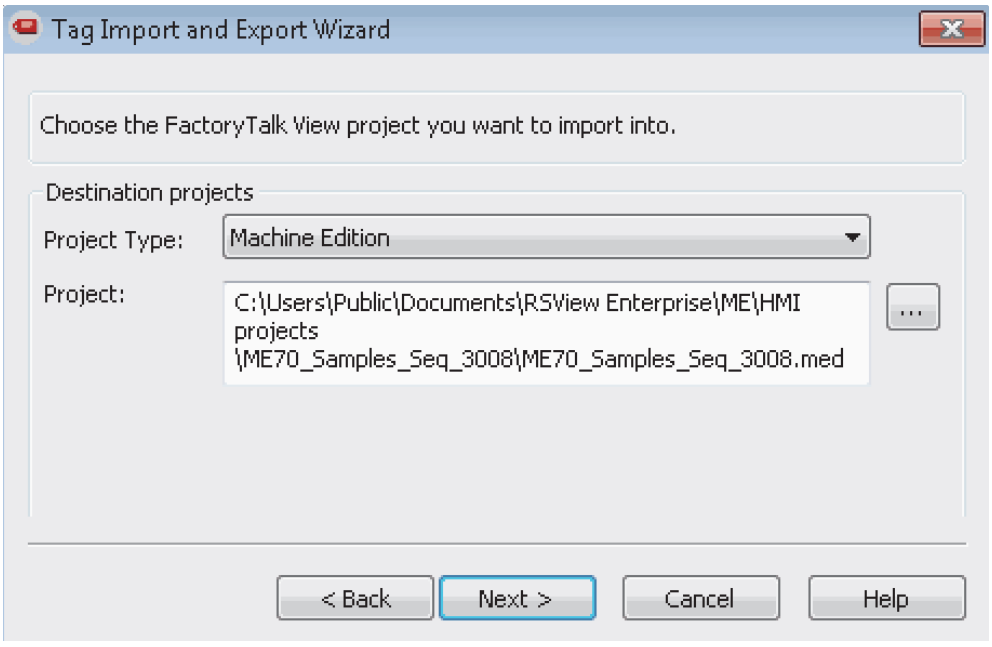
2. From the Operation pull-down menu, choose Import FactoryTalk View tag CSV files and click Next.

The Tag Import and Export Wizard dialog box reappears with a blank Project text box.



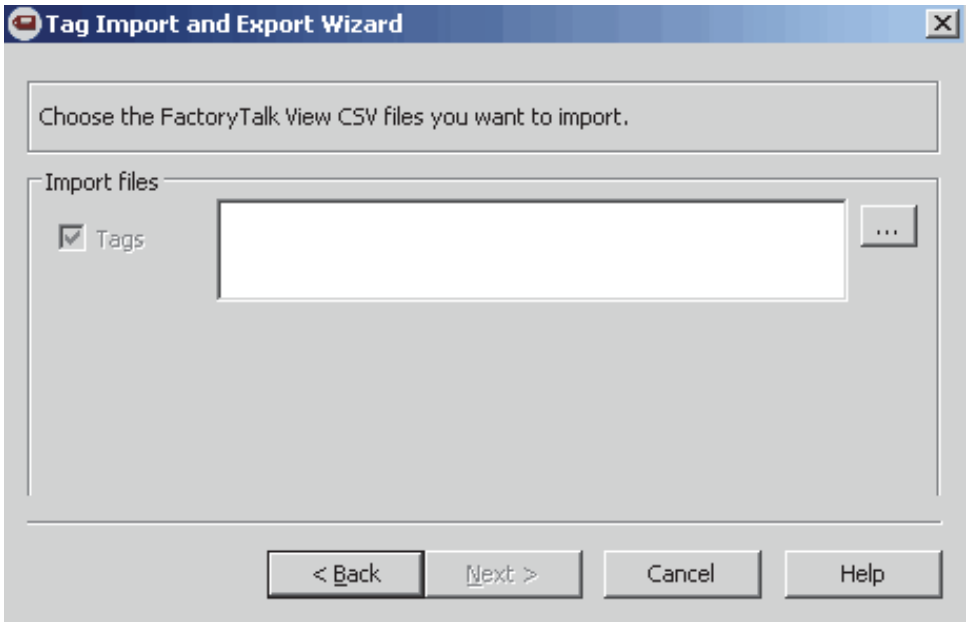
3. From the Project text box, click Browse (...) and select the .med project file that you want the HMI tags imported into and click Open.

   The Tag Import and Export Wizard dialog box reappears with the .med file in the Project text box.



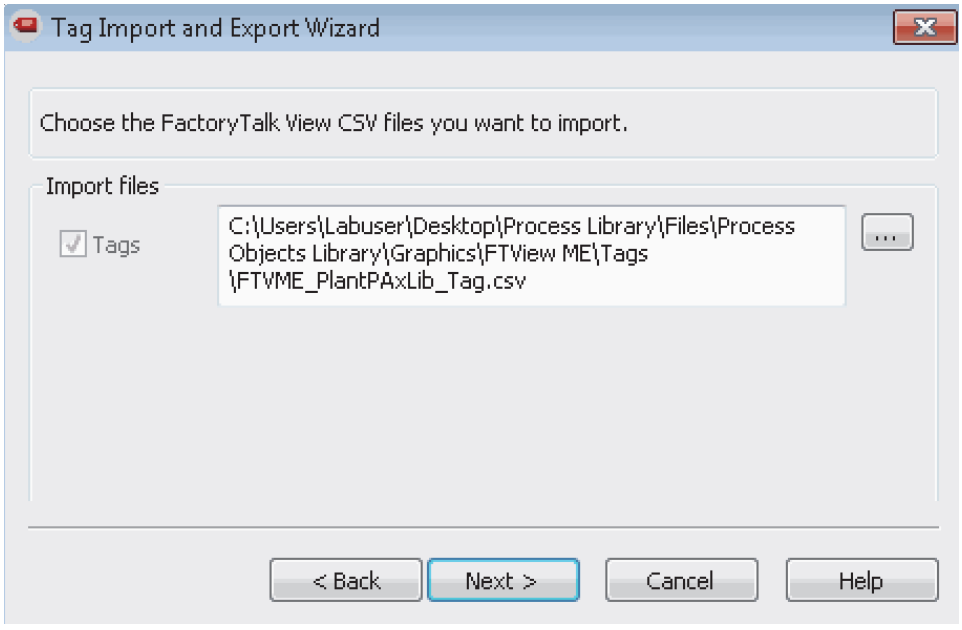
4. Click Next.

The Tag Import and Export Wizard dialog box reappears with a blank Import Files text box.



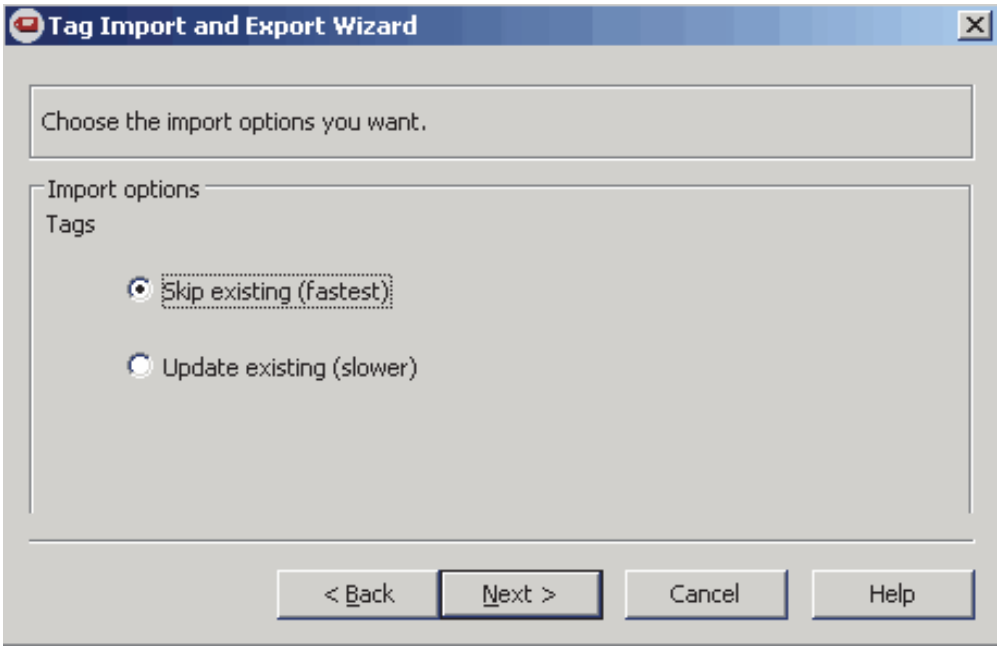
5. From the Import files text box, click Browse (...) and select the .csv file that is contained within the downloaded Library zipped file.
6. Click Open.

   The Tag Import and Export Wizard dialog box reappears with the selected .CSV file.



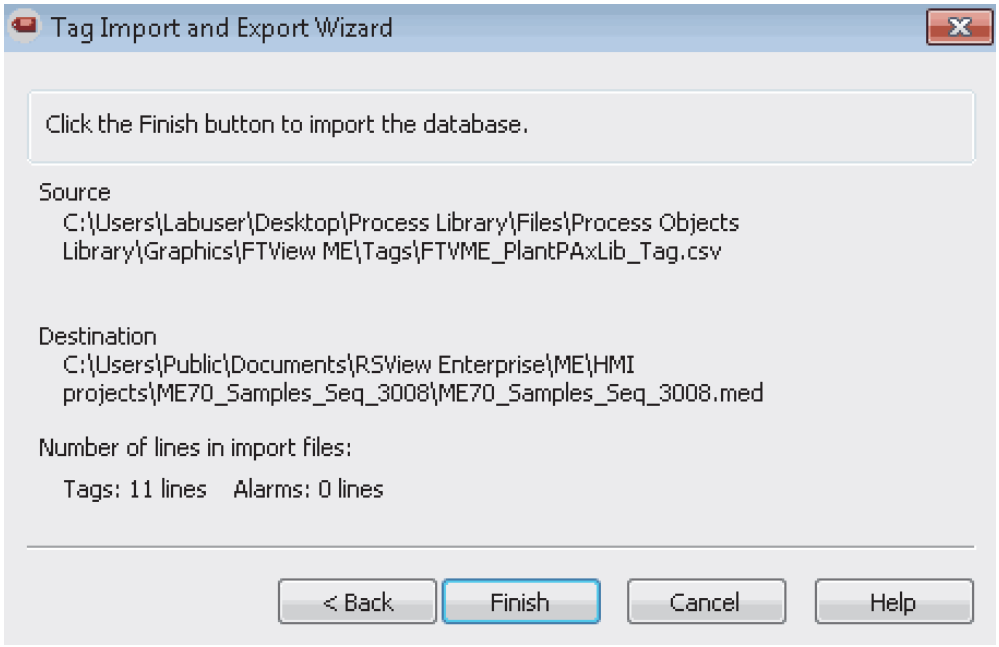
7. Click Next.

The Tag Import and Export Wizard dialog box reappears.



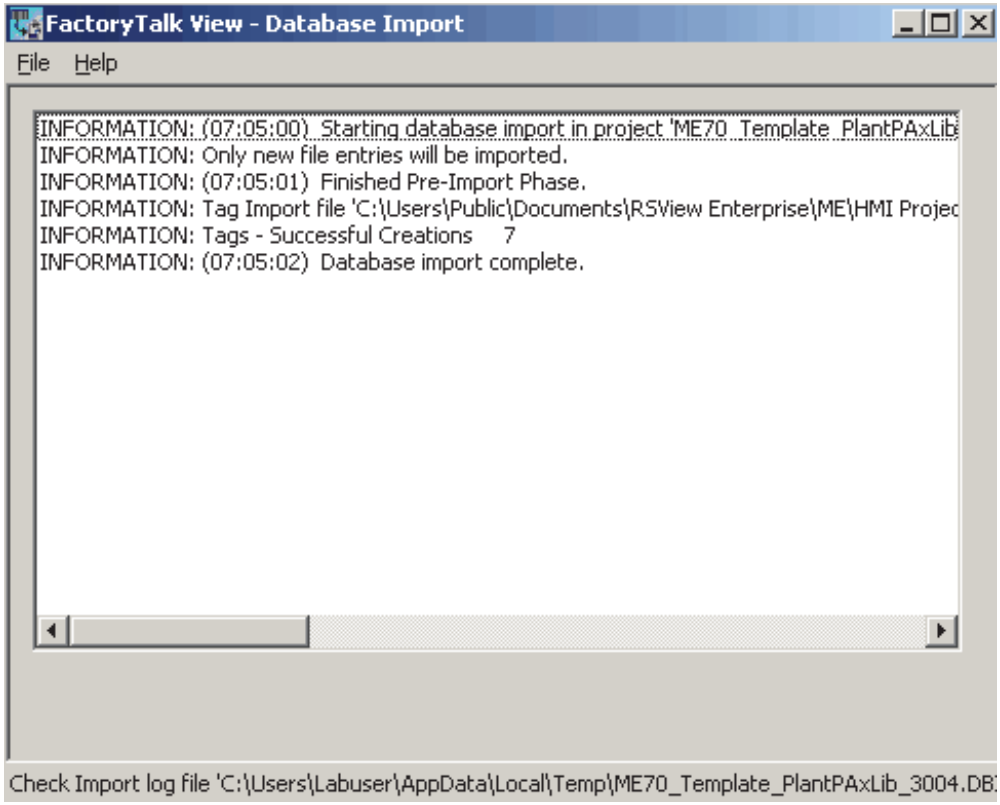
8. Use the Skip existing (fastest) option and click Next.

   The Tag Import and Export Wizard dialog box reappears.



9. Click Finish to import the HMI files.

The FactoryTalk View - Database Import dialog box appears with the information that the import is complete.



10. Click the 'X' in the upper, right corner of the window to close the window and complete the import.

Notes:

Common Configuration Considerations

This chapter includes programming considerations that are common for all Process Objects.

The table describes the topics in this chapter.

Library Programming Considerations

Multiple programming languages are available for your Library Add-On Instructions based on the type of application that you are creating. The Add-On Instruction logic can be used with Ladder Diagram, Function Block Diagram, and Structured Text languages.

Review the following programming language examples for the P_Motor instruction. Use whatever language is 'best' for your application.

Figure 3 - Ladder Diagram

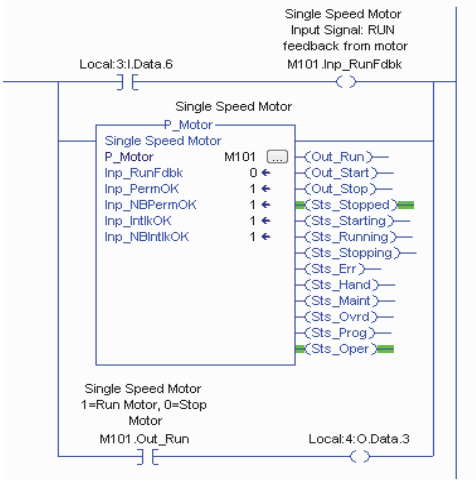


Figure 4 - Function Block Diagram

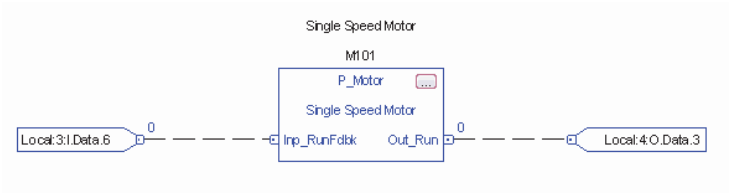


Figure 5 - Structured Text

```
M101.Inp_RunFdbk:= Local:3:I.Data.6;
P_Motor(M101);
Local:4:O.Data.3:= M101.Out_Run;
```

Once created, an Add-On Instruction can then be used in any of the RSLogix 5000 software routines without any additional effort on your part. This provides the flexibility of interfacing to the library through the programming method that you use for developing control strategies in the application code.

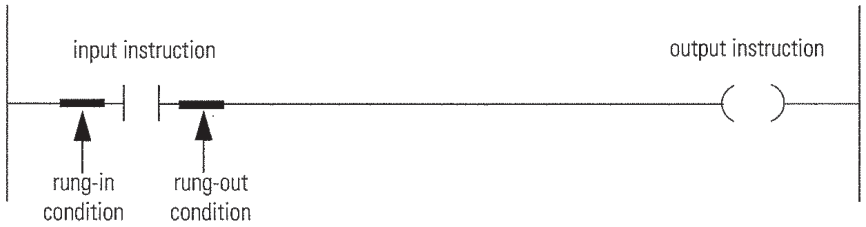
Ladder Diagram logic executes simple boolean logic, timers, and counters the fastest. Function Block Diagrams and Structured Text can give you an advantage of the more advanced process and drives instructions available in those languages.

You cannot compare execution times for the same Add-On Instruction written in different programming languages. There are fundamental differences on how the different languages execute and are compiled.

Ladder Diagram Considerations

Although multiple programming languages are available to be used with the Library Add-On Instructions, Ladder Diagram has differences in behavior to consider. The controller evaluates Ladder Diagram instructions based on the rung condition preceding the instruction (rung-in condition).

Based on the rung-in condition and the instruction, the controller sets the rung condition following the instruction (rung-out condition), which affects any subsequent instruction.



If the rung-in condition to an input instruction is true, the controller evaluates the instruction and sets the rung-out condition based on the results of the instruction. If the instruction evaluates to true, the rung-out condition is true; if the instruction evaluates to false, the rung-out condition is false.

IMPORTANT The rung-in condition is reflected in the EnableIn parameter and determines how the system performs each Process Add-On Instruction. If the EnableIn signal is **true**, the system performs the instruction's main logic routine. Conversely, if the EnableIn signal is **false**, the system performs the instruction's EnableInFalse routine.

The instruction's main logic routine sets/clears the EnableOut parameter, which then determines the rung-out condition. The EnableInFalse routine cannot set the EnableOut parameter. If the rung-in condition is **false**, then the EnableOut parameter and the rung-out condition also is **false**.

Prescan

During the transition into Run mode, the controller performs a Prescan before the first logic scan. Prescan is a special scan of all routines in the controller. The controller scans all main routines and subroutines during Prescan, but ignores jumps that could skip the execution of instructions. The controller executes all FOR loops and subroutine calls. If a subroutine is called more than once, it is executed each time it is called. The controller uses Prescan of built-in instructions to reset non-retentive data values.

During Prescan, input values are not current and outputs are not written. The following conditions generate Prescan:

- Toggle from Program to Run mode
- Automatically enter Run mode from a power-up condition

Prescan does not occur for a program when the following occurs:

- Program becomes scheduled while the controller is running
- Program is unscheduled when the controller enters Run mode

IMPORTANT The Prescan performs the Process Add-On Instruction's logic routine as all **false** and then performs its Prescan routine as **true**.

TIP When hard-coding configuration bits in Ladder Diagram instances of Add-On Instructions, we recommend using OTL (output latch) instructions for writing configuration bits to 1 and OTU (output unlatch) instructions for writing configuration bits to 0. Avoid using OTE (output energize) instructions for writing to Add-On Instruction bits. Because OTE is a non-retentive instruction, when the controller logic prescan occurs (on Powerup or controller Program to Run transition), the bit referenced in the OTE is cleared to zero.

Example: You want the mode of a device to default to Program and you don't want anyone to be able to change this from the faceplate, so you write code to set the Cfg_ProgDefault bit. The prescan logic of the P_Mode Add-On Instruction examines this bit and sets the Program/Operator selection accordingly on Powerup. Suppose you use an OTE instruction to hard-code this configuration to 1 (Program default). The prescan of the OTE sets Cfg_ProgDefault to 0 for the duration of prescan, and the instruction powers up in Operator mode, not the desired Program mode. Using an OTL instruction to hard-code this configuration, the result is the desired action: Powering up in Program mode.

Mode Configuration

Mode indicators provide a visual reference to the current owner of the process device. The modes available are listed on the Operator tab, as shown in the example.

Standard modes are implemented in each object by using an embedded instance of the P_Mode Add-On Instruction. The available modes are Operator, Program, Override, Maintenance, and Hand. Not all modes are used in every object. The particular modes available for a given object are listed in the Reference Manual for that object.

The Mode buttons on the faceplate show the behavior to expect when clicked.

Function	Action	Function	Action
	Clicking the button acquires Operator mode and takes from Program.		Clicking the button locks the Operator mode and prevents Program from acquiring.
	Clicking the button releases Operator mode and returns to Program.		Clicking unlocks the Operator mode and lets Program acquire.

Alarm Considerations

Effective alarm management is an important function of a process control system. This section describes how to use FactoryTalk View Alarm and Event software to create alarms for library objects to help safeguard personnel and plant assets.

IMPORTANT Version 3.1 of the Rockwell Automation Library requires FactoryTalk View software version 7.0 or later. FactoryTalk Alarm and Event software within View 7.0 supports new features for operator shelving of alarms, improves alignment with ANSI/ISA-18.2 (2009) alarm management standards, and gives operators new tools for dealing with alarm conditions to improve their response to abnormal process conditions.

FactoryTalk Alarm and Event software within View 8.0 supports additional features, such as dynamic alarm security, and is recommended for new applications.

The Library uses instances of a dedicated Add-On Instruction, P_Alarm, for alarm handling. For example, the P_Motor instruction uses four P_Alarm instances for the following four alarms:

- Fail to Start (used in the examples of this section)
- Fail to Stop
- Interlock Trip
- I/O Fault

The methods described in this section can be used to configure any alarm for objects in the Library of Process Objects. Recommended procedures for connecting the Add-On Instruction instance alarms are presented in the following sub-sections:

- Alarm with FactoryTalk Alarm and Event Server on page 92
- Alarm with FactoryTalk View ME Software on page 98

If you are using FactoryTalk View SE software, Version 3.1 of the Library of Process Objects uses a P_Alarm instance to communicate alarm conditions to the tag-based server. Digital alarm tags for P_Alarm instances can be configured manually from within FactoryTalk View Studio. We also provide the PlantPAx Alarm Builder tool (see Appendix B) that helps to streamline the digital alarm tag definition process and facilitates bulk configuration.

Table 34 summarizes the P_Alarm alarm types that are used with the Library, and indicates which objects use the alarm.

Table 34 - P_Alarm Types by Library Objects

Alarm Type	Alarm Description	Library Objects
ActuatorFault	Raised if the Inp_ActuatorFault input is true. This is provided for use by valves that generate a fault contact, such as actuator motor overload trip	Valve objects, including: • P_ValveC • P_ValveMO
AnyReject	At least one input signal has been rejected because of the following: • It's outside the configured failure range • It's outside of two standard deviations from the mean • Has its Bad Quality input bit set • Has a floating-point value that is infinite or not a number (floating-point exception)	I/O Processing objects, including: P_AInMulti
DeviceFault	Device fault from device via an input	Valve objects, including: P_D4SD
Diff	The difference between the two input signals exceeds the configured high difference warning limit	I/O Processing objects, including: P_AInDual
DriveFault	Raised if the Inp_Faulted input is true. Enables display of the Drive Fault code	Motor objects, including: • P_PF52x • P_PF753 • P_PF755 • P_VSD
EqpFault	Raised if the controlled equipment asserts the Inp_CtrldEqpFault input or if the equipment feedback signals fail to track the commanded state of the equipment. If configured as a shed fault, this also stops flow if it is running	Additional objects, including: • P_DoseFM • P_DoseWS
Fail	PV quality is bad (Inp_PVBad is true) or PV is beyond configured Fail High and Fail Low thresholds. Thresholds are set in configuration	I/O Processing, Valve, and Regulatory Control objects, including: • P_AIChan • P_AIn • P_AInAdv • P_AInDual • P_AInMulti
	Device failed to reach commanded state	• P_D4SD • P_ValveMP
	Loop failure: PV bad, SP bad or CV communication failure or bad	• P_PIDE
FailToStart	Raised if the motor has and is using run feedback, an attempt is made to start the motor, and the run feedback does not indicate the motor running within the configured time	Motor objects, including: • P_Motor • P_Motor2Spd • P_MotorRev • P_PF52x • P_PF753 • P_PF755 • P_SMC50 • P_SMCFlex • P_VSD
FailToStop	Raised if the motor has and is using run feedback, an attempt is made to stop the motor, and the run feedback does not indicate the motor stopped within the configured time	Motor objects, including: • P_Motor • P_Motor2Spd • P_MotorRev • P_PF52x • P_PF753 • P_PF755 • P_SMC50 • P_SMCFlex • P_VSD
FullStall	Raised if the Valve has and is using Open and/or Closed feedback, an attemptis made to open or close the valve, and the valve position feedback indicates the valve did not move off its original position at all during the configured time	Valve objects, including: • P_ValveMO • P_ValveSO
Hi	PV above High threshold. Threshold is set by Operator or Program. Deadband, gating, timing, and severity are set in configuration	I/O Processing objects, including: • P_AIn • P_AInAdv • P_AInDual • P_AInMulti

Table 34 - P_Alarm Types by Library Objects

Alarm Type	Alarm Description	Library Objects
HiDev	PV exceeds SP by High threshold. Threshold is set by Operator or Program. Deadband, gating, timing, and severity are set in configuration	I/O Processing objects, including: • P_AInDual • P_PIDE
HiHi	PV above High-High threshold. Threshold is set by Operator or Program. Deadband, gating, timing, and severity are set in configuration	I/O Processing objects, including: • P_AIn • P_AInAdv • P_AInDual • P_AInMulti
HIHIDev	PV exceeds SP by High-High threshold. Threshold is set by Operator or Program. Deadband, gating, timing, and severity are set in configuration	Regulatory Control objects, including: P_PIDE
HiRoC	Absolute value of PV rate of change above High Rate of Change limit. Limit set by Operator or Program. Deadband and severity in configuration	I/O Processing objects, including: P_AInAdv
IntlkTrip	Triggered when an Interlock not OK causes the device to transition state	I/O Processing, Motor, and Valve objects, including: • P_D4SD • P_DOut • P_Motor • P_Motor2Spd • P_MotorHO • P_MotorRev • P_nPos • P_PF52x • P_PF753 • P_PF755 • P_PIDE • P_SMC50 • P_SMCFlex • P_ValveC • P_ValveHO • P_ValveMP • P_ValveSO • P_VSD
IOFault	Triggered by the Inp_IOFault Input and used to indicate an I/O communication failure	I/O Processing, Motor, and Valve objects, including: • P_D4SD • P_DOut • P_E1PlusE • P_E300Ovld • P_E3Ovld • P_Motor • P_Motor2Spd • P_MotorHO • P_MotorRev • P_nPos • P_PF52x • P_PF753 • P_PF755 • P_SMC50 • P_SMCFlex • P_ValveC • P_ValveHO • P_ValveMO • P_ValveMP • P_ValveSO • P_VSD
Lo	PV below Low threshold. Threshold is set by Operator or Program. Deadband, gating, timing, and severity are set in configuration	I/O Processing objects, including: • P_AIn • P_AInAdv • P_AInDual • P_AInMulti
LockFail	A device with a locking or sealing feature was commanded to a new position, but the lock/seal feedback failed to show the device unlocked before the move, or locked after the move, within the configured time allowed (Cfg_LockChkT). The device can be configured to either alarm only or to 'shed' to a de-energized state on a Lock Fail	Valve objects, including: P_nPos

Table 34 - P_Alarm Types by Library Objects

Alarm Type	Alarm Description	Library Objects
LoDev	PV falls below SP by Low threshold. Threshold is set by Operator or Program. Deadband, gating, timing, and severity are set in configuration	I/O Processing and Regulatory Control objects, including: • P_AInDual • P_PIDE
LoLo	PV below Low-Low threshold. Threshold is set by Operator or Program. Deadband, gating, timing, and severity are set in configuration	I/O Processing objects, including: • P_AIn • P_AInAdv • P_AInDual • P_AInMulti
LoLoDev	PV falls below SP by Low-Low threshold. Threshold is set by Operator or Program. Deadband, gating, timing, and severity are set in configuration	Regulatory Control objects, including: P_PIDE
MinGood	At least one input signal has been rejected, and the remaining unrejected signals are the minimum number configured as required for a good PV. The next input failure causes the PV to be flagged as Bad	I/O Processing objects, including: P_AInMulti
MotorFault	Occurs when a fault is detected by the smart motor controller	Motor objects, including: P_SMC50 P_SMCFlex
NoneGood	Warning if neither PV input's quality is good. Raised if both PV inputs have bad or uncertain quality	I/O Processing objects, including: P_AInDual
OffFail	Triggered when the device is commanded Off, but device feedback does not confirm the device is Off within the configured time. The OffFail Status/ Alarm has a configurable time (Cfg_OffFailT) so the device feedbacks can show the device reached the Off state before declaring a failure	I/O Processing objects, including: P_DOut
OneGood	Warning if only one of the two PV inputs is good quality. Raised if one input's quality is bad or uncertain	I/O Processing objects, including: P_AInDual
OnFail	Triggered when the device is commanded. On, but device feedback does not confirm the device is On within the configured time. The OnFail Status/ Alarm has a configurable time (Cfg_OnFailT) so the device feedbacks can show the device reached the On state before declaring a failure	I/O Processing objects, including: P_DOut
OverTol	Raised when tolerance check is performed if the quantity delivered exceeds the setpoint by more than the high tolerance limit	Procedural Control objects, including: • P_DoseFM • P_DoseWS
PosFail	The device was commanded to a new position, but the position feedbacks failed to show the device reached the target position within the configured time allowed (Cfg_PosChkT). For target positions other than Position 1, the device can be configured to retry the move a configurable number of times (returning to Position 1 for each try) before declaring a Position Fail. The device can be configured to either alarm only or to shed to a de-energized state on a Position Fail	Valve objects, including: P_nPos
TareFault	Raised if the scale is tared and then drifts away from the tare weight before the start flow command is received	Procedural Control objects, including: P_DoseWS
TgtDisagree	Activated when the Input Process Variable is not equal to the Target Process Variable and the Gate input is Enabled	I/O Processing objects, including: P_DIn
TransitStall	Raised if the valve has and is using Open and Closed feedback, an attempt is made to open or close the valve, and the valve position feedback indicates the valve moved from its original position but did not reach its target position within the configured time	Valve objects, including: • P_ValveHO • P_ValveMO • P_ValveSO
Trip	The Overload Trip alarm occurs when the overload has tripped	Motor objects, including: P_E1PlusE P_E300Ovld P_E3Ovld
TripFail	Raised if using the optional trip function when an attempt is made to trip the motor/valve and the feedbacks do not indicate that the trip state was achieved	Motor and Valve objects, including: • P_MotorHO • P_ValveHO

Table 34 - P_Alarm Types by Library Objects

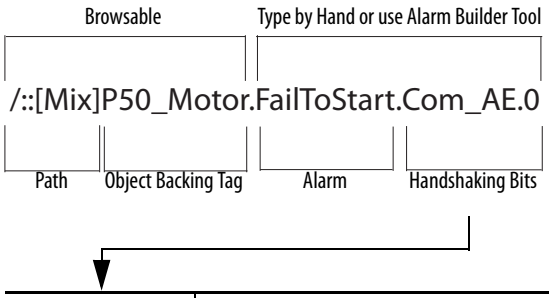
Alarm Type	Alarm Description	Library Objects
UnderTol	Raised when the tolerance check is performed if the quantity delivered falls short of the setpoint by more than the low tolerance limit	Procedural Control objects, including: • P_DoseFM • P_DoseWS
Warn	The Pending Trip Warning alarm is triggered when a motor overload condition is occurring and an overload trip is imminent	Motor objects, including: P_E1PlusE P_E300Ovld P_E3Ovld
ZeroFault	Raised if the flow total fails to clear, or the flow is cleared and still accumulates flow before the start flow command is received	Procedural Control objects, including: P_DoseFM

Alarm with FactoryTalk Alarm and Event Server

FactoryTalk View SE alarms use digital alarms; one digital alarm per P_Alarm instance. This section describes how to configure digital alarm parameters.

As shown in Figure 6, each object with alarms has a P_Alarm instance for each alarm, and each P_Alarm instance has a Local Tag (.Com_AE.*x*) specifically for communicating all alarm status and commands with the FactoryTalk Alarm and Event server.

Figure 6 - FactoryTalk View SE Alarm Path



Local INT Tag with Bit Number

Communication Bits	Alarm Description	AE Status	AE Control
Com_AE.0	Alarm to AE (alarm condition required for digital alarm)	No	No
Com_AE.1	Acknowledged from AE Status, Acknowledge to AE Control	Yes	Yes
Com_AE.2	Not used	N/A	N/A
Com_AE.3	Shelved from AE Status	Yes	No
Com_AE.4	Shelve to AE Control	No	Yes
Com_AE.5	Unshelve to AE Control	No	Yes
Com_AE.6	Suppressed from AE Status (see Important on page 93)	Yes	No
Com_AE.7	Suppress to AE Control	No	Yes
Com_AE.8	Unsuppress to AE Control	No	Yes
Com_AE.9	Disabled from AE Status (see Important on page 93)	Yes	No
Com_AE.10	Disable to AE Control	No	Yes
Com_AE.11	Enable to AE Control	No	Yes
Com_AE.12	Not used	N/A	N/A
Com_AE.13	Not used	N/A	N/A
Com_AE.14	Not used	N/A	N/A
Com_AE.15	Not used	N/A	N/A

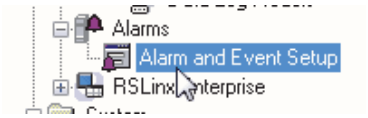
As shown in the chart with Figure 6, by using a single INT (Com_AE) tag for the interface and bits for status and control, the number of tags and elements on scan in the FactoryTalk Alarm and Event server are kept to a minimum.

IMPORTANT Status and control bits for Suppress and Disable have been separated for Version 3.1-02 to improve FactoryTalk Alarm and Event import behavior. If you update the P_Alarm Add-On Instruction it is not necessary to modify an existing FactoryTalk Alarm and Event configuration. However, certain alarms can remain disabled following an alarm import. It's important to make sure that this situation does not occur. If you update to the new bit assignments shown in Figure 6, this situation does not occur but you must update the P_Alarm Add-On Instruction to the latest system version.

Adding a Digital Alarm

Complete these steps to add a digital alarm.

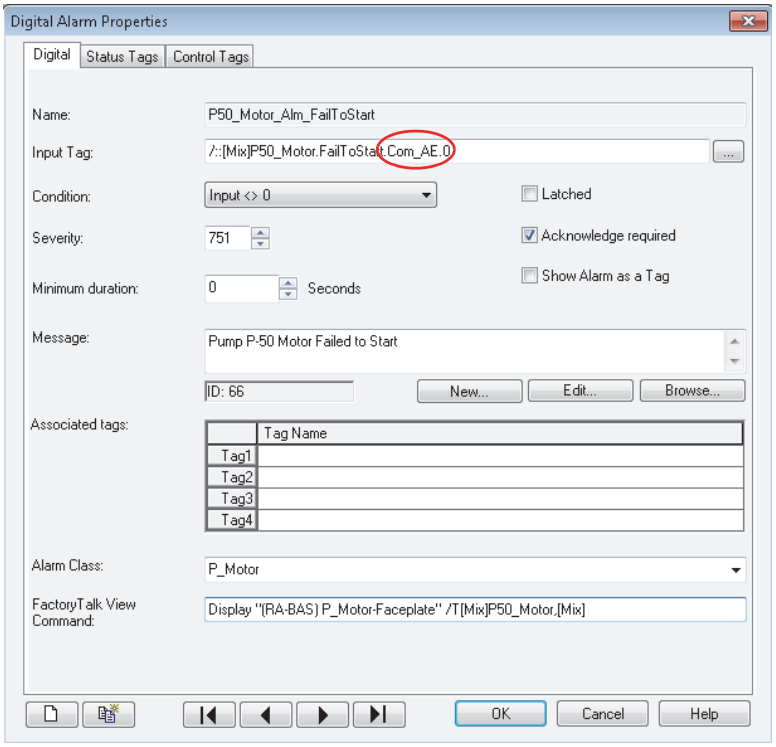
1. Open the FactoryTalk View Studio software.
2. In the Explorer window, click Alarms and double-click Alarm and Event Setup.



The Alarm and Event Setup dialog box appears with the All Alarms tab.

3. Click the New toolbar icon and choose Digital.

The Digital Alarm Properties dialog box appears empty for a new alarm. The example shows data for instructional purposes.



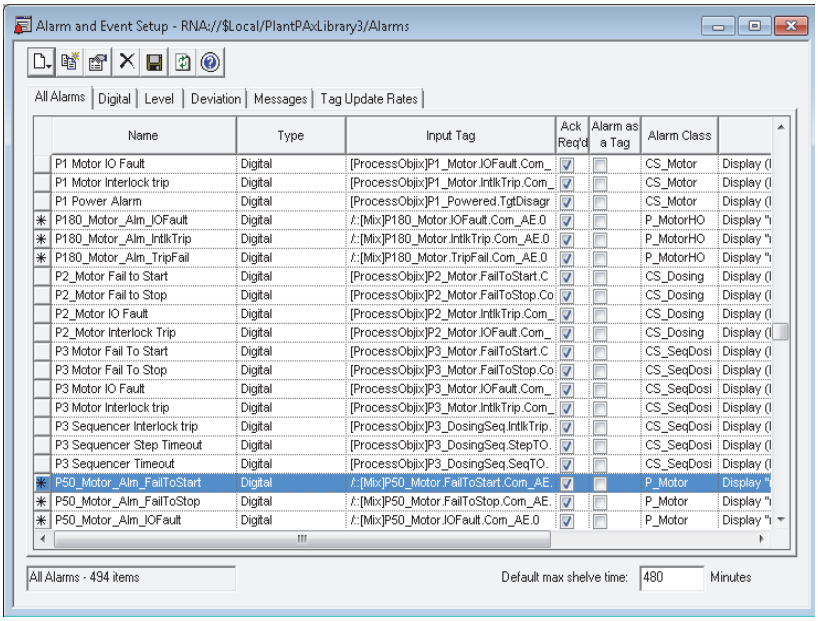
IMPORTANT In the 'Com_AE' local tag, bit .0 (circled in the example on page 93) is the alarm condition required for the digital alarm in the FactoryTalk Alarm and Event setup.

4. Complete the Digital Alarm Properties dialog box.

Topic	Description
Name	Type a unique designator for the object as well as a short description of the alarm. In the example, 'P50_Motor' identifies the P_Motor object and 'Alm_FailToStart' identifies the alarm.
Input Tag	Type bit .0 of the Com_AE local tag in the P_Alarm instance. You can click Browse (…) to select the path and object but not for local tags. Therefore, you must type the last part of the input tag (Com_AE.0). **IMPORTANT:** A simple method to generate the alarm name, especially in an editing tool like Microsoft Excel, is to remove the communication path (shortcut) and replace the dot (".") separating the tag name and the alarm status parameter name with an underscore ('_'). For example, the input tag: /::[Mix]P50_Motor.Alm_FailToStart becomes alarm name: P50_Motor_Alm_FailToStart
Condition	From the Condition pull-down menu, choose Input <> 0.
Severity	Choose an alarm severity that aligns with the severity in the controller and is shown on the HMI faceplate. Severity values include the following: **Severity on Faceplate** / **A & E Severity** 1…250 = Low / 1…250 251…500 = Medium / 251…500 501…750 = High / 501…750 751…1000 = Urgent / 751…1000 **IMPORTANT:** The Alarms Builder tool works with Library versions 2.0, 3.0, and 3.1. The mapping shown above is for version 3.1 that has a range from 1…1000 (INT data type). For Add-On Instructions version 3.0 and earlier, the severity values are in the range of 1…4 (SINT data type). When an ACD file that contains version 3.0 Add-On Instructions is updated with version 3.1 Add-On Instructions, the Add-On Instruction tags retain their existing severity values (1…4).
Minimum duration	Set to zero. Alarm on-delay and off-delay timing is handled by the controller.
Latched	Leave blank. Alarm latching (reset required) is handled by the controller.
Acknowledge required	Make sure there is a check in the checkbox. Alarm auto-acknowledgement for alarms configured as 'Ack Not Required' from the faceplate is handled by the controller.
Show Alarm as a Tag	Leave blank.
Message	Type a message (optional) into the text box.
Associated tags	Type any associated tags (optional) into the text box.
Alarm Class	Choose a classification (optional) from the pull-down menu for alarm grouping and filtering.
FactoryTalk View Command	Type a FactoryTalk View command that accesses a faceplate when you double-click an alarm for a particular object on an A&E Alarm Summary dialog box. The command is configured as this example: 'Display {faceplate display name} /TPath object}' The name of the display must be inside double-quotes because it contains a space.

5. Click OK.

The digital alarm configuration information appears on the Alarm and Event Setup dialog box.

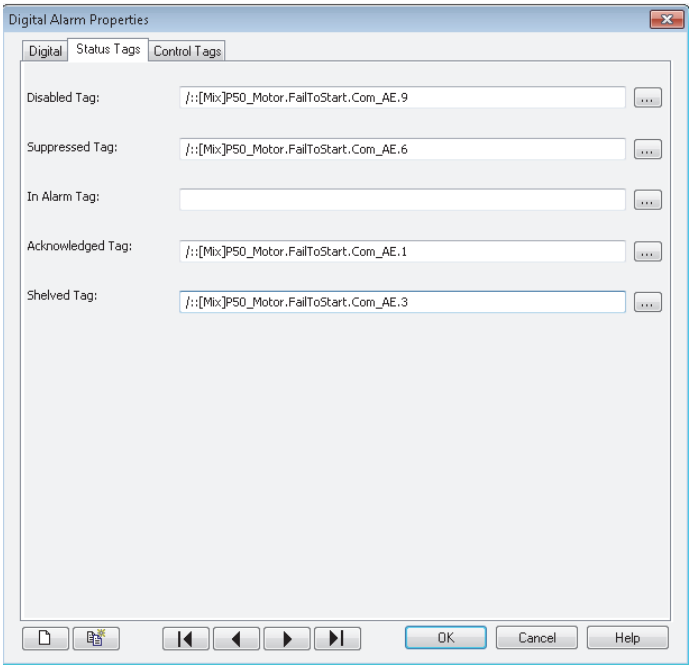


Add Digital Alarm Status Tags

The status tags configuration sends to the controller any shelved/unshelved, disabled/enabled, suppressed/unsuppressed and acknowledged status updates from the FactoryTalk Alarms and Events Alarm Summary and Alarm Status Explorer dialog boxes.

1. From the Alarms Properties dialog box, click the Status Tags tab.

   The Status Tags dialog box appears.



2. Complete the status tag dialog boxes.

Table 35 - Status Tag Dialog Box

Field	Description
Disabled Tag	Type "'Path Object.Alarm.Com_AE.9" for the disabled tag. Our example is /::[Mix]P50_Motor.FailToStart.Com_AE.9
Suppressed Tag	Type 'PathObject.Alarm.Com_AE.6 for the suppressed tag. Our example is /::[Mix]P50_Motor.FailToStart.Com_AE.6
In Alarm Tag	Leave blank.
Acknowledged Tag	Type 'Path Object.Alarm.COM_AE.1' for the acknowledged tag. Our example is /::[Mix]P50_Motor.FailToStart.Com_AE.1
Shelved Tag	Type 'Path Object.Alarm.COM_AE.3' for the shelved tag. Our example is /::[Mix]P50_Motor.FailToStart.Com_AE.3

IMPORTANT Even though Com_AE is a Local Tag in the Add-On definition, it's configured to be writable (Read/Write, not Read Only) so the FactoryTalk Alarm and Event server status is sent to the bits identified for the tags above.

3. Click OK.

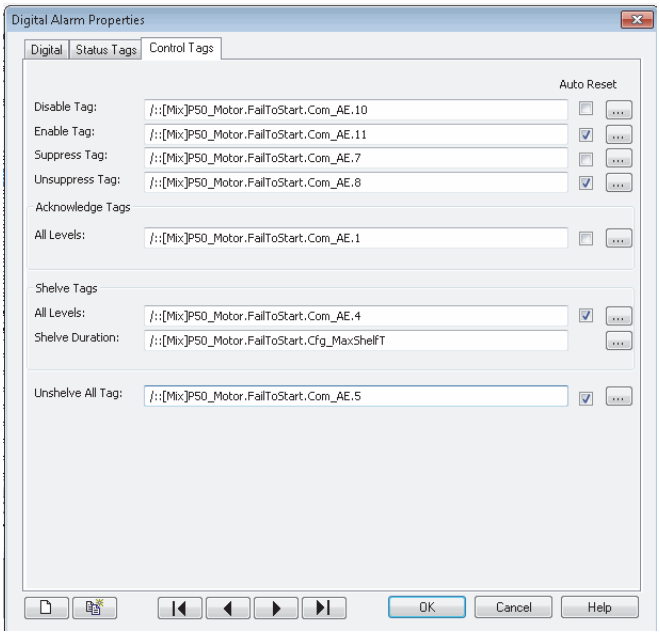
Add Digital Alarm Control Tags

The Control Tags configuration lets the FactoryTalk Alarm and Event server to access alarm acknowledgment, disable, enable, shelve, and unshelve actions performed via the faceplates, and suppress and unsuppress actions performed in controller logic.

Some of the status bits are reused as control bits with the single INT (Com_AE) tag to reduce the number of tags and elements on scan in the FactoryTalk Alarm and Event server.

1. From the Alarms Properties dialog box, click the Control Tags tab.

The Control Tags dialog box appears.



2. Complete the control tag dialog boxes.

Field	Description
Disable Tag Auto Reset	Type "Path Object.Alarm.Com_AE.10" for the disable tag. Leave the Auto Reset checkbox blank. Our example is /::[Mix]P50_Motor.FailToStart.Com_AE.10
Enable Tag Auto Reset	Type "Path Object.Alarm.Com_AE.11" for the enable tag. Check the Auto Reset checkbox (so the alarm automatically resets when acknowledged.) Our example is /::[Mix]P50_Motor.FailToStart.Com_AE.11
Suppress Tag Auto Reset	Type "Path Object.Alarm.Com_AE.7" for the suppress tag. Leave the Auto Reset checkbox blank. Our example is /::[Mix]P50_Motor.FailToStart.Com_AE.7
Unsuppress Tag Auto Reset	Type "Path Object.Alarm.Com_AE.8" for the unsuppress tag. Check the Auto Reset checkbox (so the alarm automatically resets when acknowledged.) Our example is /::[Mix]P50_Motor.FailToStart.Com_AE.8
Acknowledge Tags All Levels Auto Reset	Type "Path Object.Alarm.Com_AE.1" for the acknowledge tag. Leave the Auto Reset checkbox blank. Our example is /::[Mix]P50_Motor.FailToStart.Com_AE.1
Shelve Tags All Levels Auto Reset	Type "Path Object.Alarm.Com_AE.4" for the shelve tag. Check the Auto Reset checkbox (so the alarm automatically resets when acknowledged.) Our example is /::[Mix]P50_Motor.FailToStart.Com_AE.4
Shelve Duration	Type "Path Object.Alarm.Cfg_MaxShelfT" for the shelve duration tag. Our example is /::[Mix]P50_Motor.FailToStart.Cfg_MaxShelfT
Unshelve All Tags	Type "Path Object.Alarm.Com_AE.5" for the unshelve tag. Check the Auto Reset checkbox (so the alarm automatically resets when acknowledged.) Our example is /::[Mix]P50_Motor.FailToStart.Com_AE.5

3. Click OK.

Alarm with FactoryTalk View ME Software

This section describes how to configure a library alarm in FactoryTalk View ME software for use with PanelView Plus and other FactoryTalk View ME terminals. Because FactoryTalk View ME software does not support features for suppression, shelving, and disabling, this procedure has not changed from the method used for the Library Version 2.0.

This method does support acknowledgement of alarms from the FactoryTalk View ME terminal. In addition, because the FactoryTalk View ME faceplates are the same as the FactoryTalk View SE faceplates, operator actions for shelve, unshelve, disable, enable, and acknowledgement are recognized by the P_Alarm Add-On Instruction, and by the FactoryTalk SE Alarm and Event server.

The missing capabilities in FactoryTalk View ME include the following:

- Ability to shelve/unshelve and disable/enable from the Alarm Status Explorer (there is none) or Alarm Summary
- Ability to provide lists of shelved, suppressed, or disabled alarms

An operator on a FactoryTalk View ME terminal can shelve alarms, or maintenance personnel can disable alarms (and the P_Alarm Add-On Instruction prevents new alarms from being generated when shelved). The shelved or disabled status is reflected on the FactoryTalk View SE Alarm Status Explorer.

The examples use the following attributes:

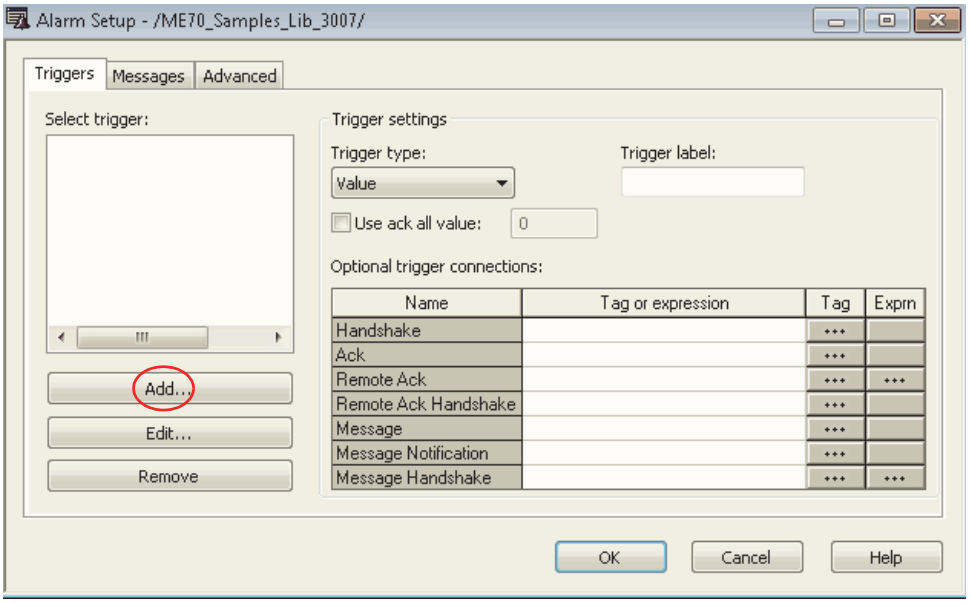
- Path is /::[Mix]
- Object is P50_Motor
- Alarm is FailToStart

Create a New Alarm Trigger

Complete these steps.

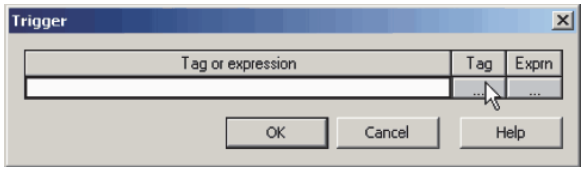
1. Open the FactoryTalk View Studio software.
2. In the Explorer window, click Alarms and double-click Alarm Setup.

The Alarm Setup dialog box appears with the Triggers tag open.



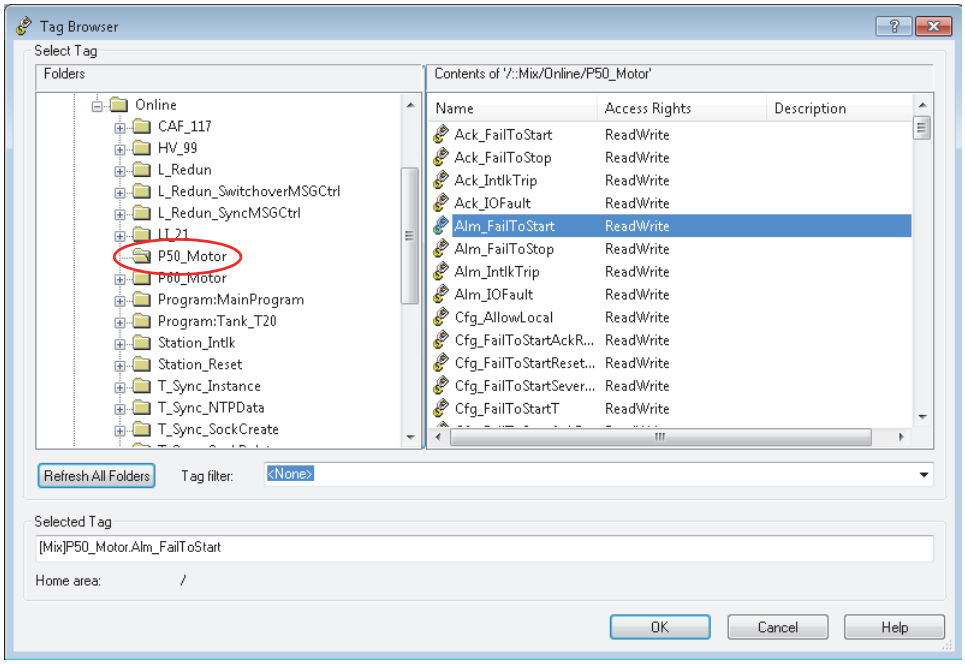
3. Click Add.

   The Trigger dialog box appears.



4. Click Browse (...) under Tag.

   The Tag Browser appears.



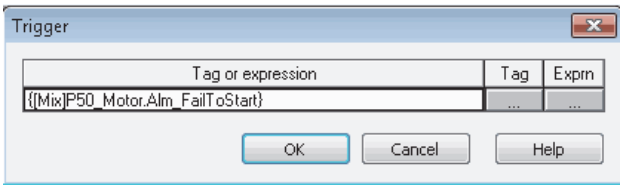
5. In the left pane of the Tag Browser, click the folder representing the Add-On Instruction instance. Our example is P50_Motor.

You can use the folder for the offline controller. If you are connected to the actual hardware controller on a network and the controller application is leaded, you can use the online folder.

6. In the right pane of the Tag Browser, double-click the alarm status parameter for the alarm.

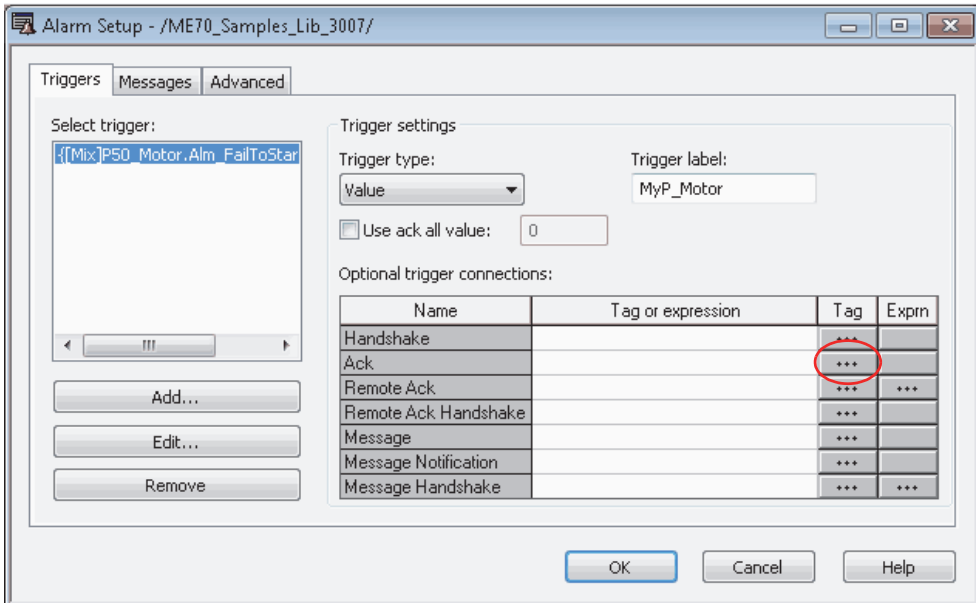
   TIP The alarm status parameter's name starts with 'Alm_'. For example, the parameter is 'Alm_FailToStart'.

   The alarm status parameter appears in the Trigger pop-up window.



7. Click OK.

   The alarm status parameter appears in the Select trigger box of the Alarm Setup dialog box.



8. Click Browse (...) in the Tag column for the Ack row.

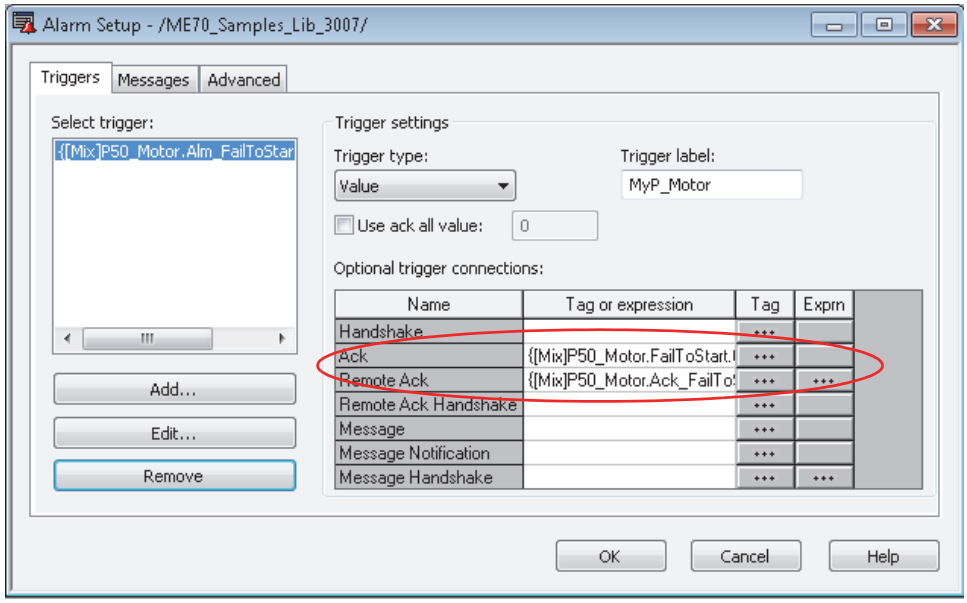
   'Ack' is an abbreviation for acknowledge.

   The Tag Browser appears when you click Browse.

9. Using the Tag Browser, choose the acknowledge status parameter.

10. Repeat step 8 and step 9 to choose the acknowledge status parameter for the Remote Ack row.

Your setup now looks like the example.



The 'Ack' parameter must be changed so it triggers the Operator Acknowledge Command, which is part of the alarm's Local Tag within the Add-On Instruction. Because this parameter is in a Local Tag, it cannot be browsed. But, the FactoryTalk View ME HMI server still can write to the tag.

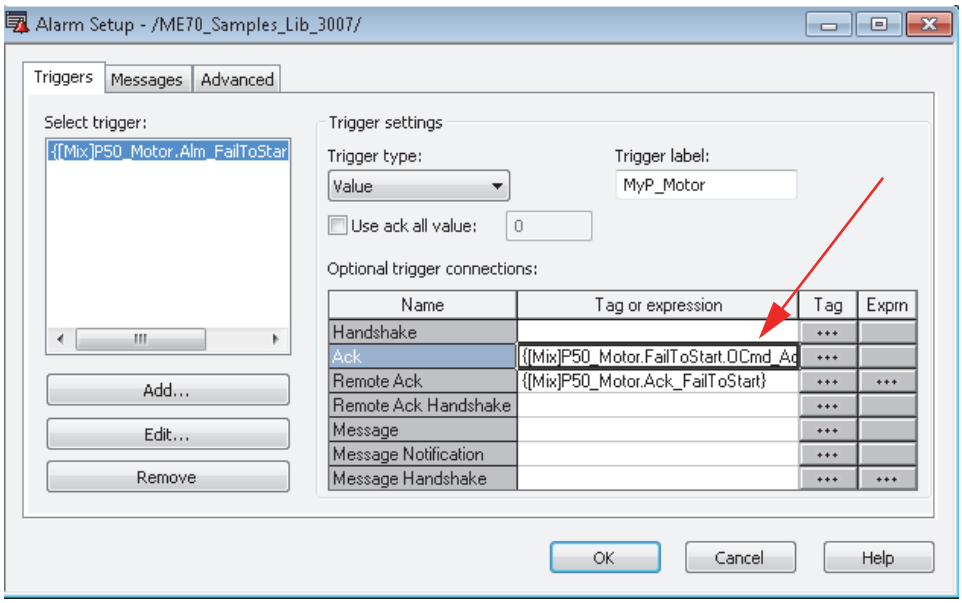
11. To change the 'Ack' parameter, click the tag in the Tag or expression box to access a text cursor and manually type the change.

    The format of the expression changes from:

    'Path Object.Ack_Alarm'

    to:

    'Path Object. Alarm.OCmd_Ack'



IMPORTANT Each 'Ack' tag occurrence can be changed in the alarm database by exporting the alarms and making the changes with an editing tool, then importing the changes. Simply Find and Replace every 'Ack' tag of:

.Ack_FailToStart

to:

.FailToStart.0Cmd_Ack

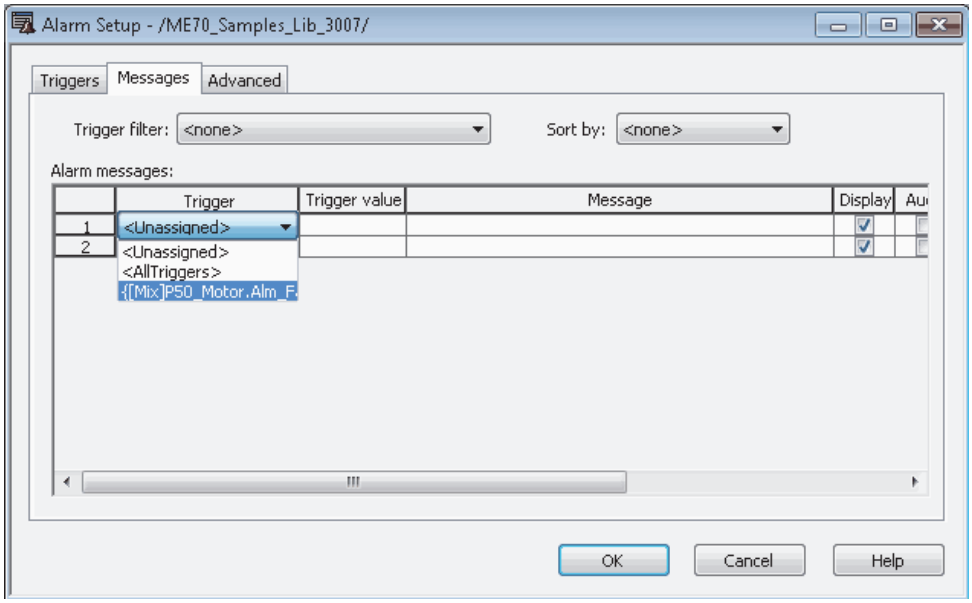
Perform a similar Find and Replace for each type of alarm: FailToStop, IOFault, and so forth.

Be careful when performing the Find and Replace procedure that you do not change the RemoteAck entries.

Configure the Alarm Message

Follow these steps to define a message that appears when the alarm occurs.

1. On the FactoryTalk View ME Alarm Setup dialog box, click the Messages tab.

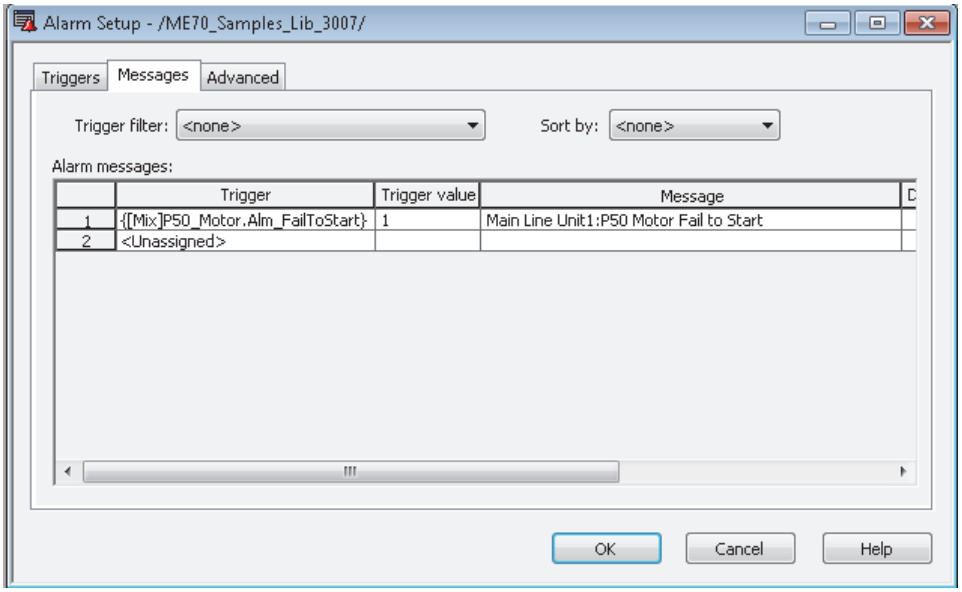


2. From the Trigger pull-down menu, choose the Trigger tag that you created.

   Our example is the {[Mix]P50_Motor.

3. In the Trigger Value column, type 1.

4. In the Message column, type the message to appear with the alarm.

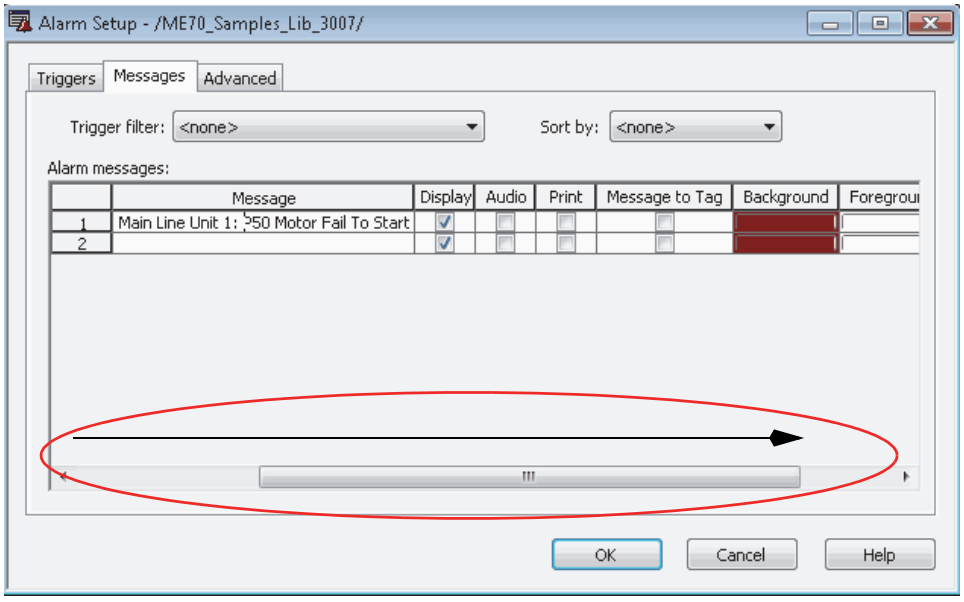


5. Click OK.

Configure Notification Options

Follow these steps for audio or visuals when an alarm occurs.

1. On the Message tab of the Alarm Setup dialog box, click and drag the bottom display bar to the right to show the rest of the columns for the alarm message.



2. Complete the notification options.

Field	Description
Display	Leave the check in the checkbox so you see the alarm display when there is an alarm occurrence.
Audio	Check the checkbox if you want a sound to play from the HMI terminal's speakers when an alarm occurs.
Print	Check the checkbox if you have a printer attached to the HMI terminal (directly or via a network) and you want the alarm to print when the alarm occurs.
Message to Tag	We recommend that you leave the box blank.
Background and Foreground	Select the colors that are used to display the alarm on the alarm summary. Colors selected must match the alarm severity.

Severity	Foreground	Background
1…250 (Low)	White	Blue
251…500 (Medium)	Black	Bright Yellow
501…750 (High)	Black	Bright Red
751…1000 (Urgent)	Black	Bright Magenta

3. Repeat these steps for additional messages.
4. Click OK.

Security Configuration

Runtime security must be set up to provide each account or user group with the correct FactoryTalk View security codes. The security codes verify that operators, maintenance personnel, and engineers have permission to run secured commands, open secured graphic displays, or write to secured HMI tags at runtime.

IMPORTANT See Appendix E for a list of security codes and descriptions.

FactoryTalk Directory stores information about which users have access to the parts of a control system. During the log on, FactoryTalk Security uses this information to verify the user's identity and then permissions assigned to the user. Authorized users can then access secured parts of the application.

IMPORTANT FactoryTalk Security settings are stored separately for a Local Directory and a Network Directory, even if both are in use on the same computer. You must set up security permissions twice—once for the Local Directory and once for the Network Directory—to give one user access to a local and a network distributed application on the same computer.

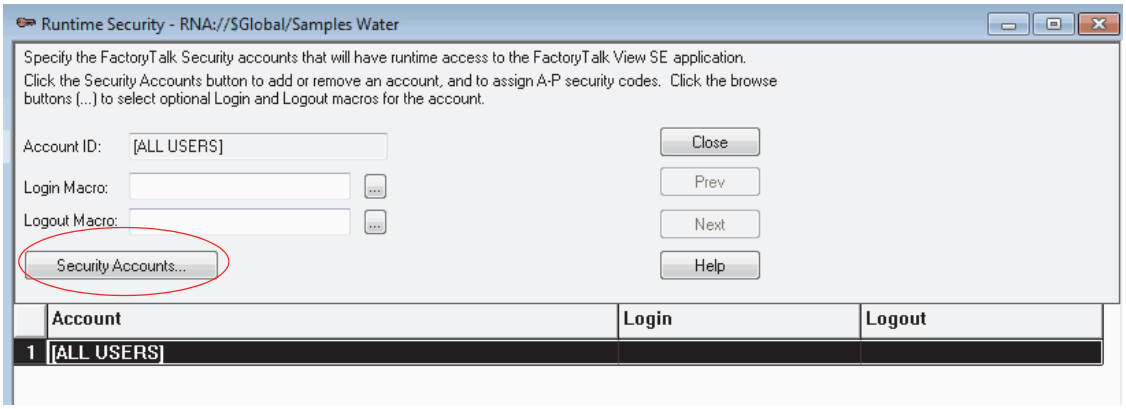
Add Users to Security Codes

The following steps are for adding a user or group account to an **existing** FactoryTalk security account.

If you are adding a new user, you must create the FactoryTalk account first, and then add the account in the Runtime Security editor. For procedures, see the FactoryTalk View Site Edition User's Guide, publication VIEWSE-UM006.

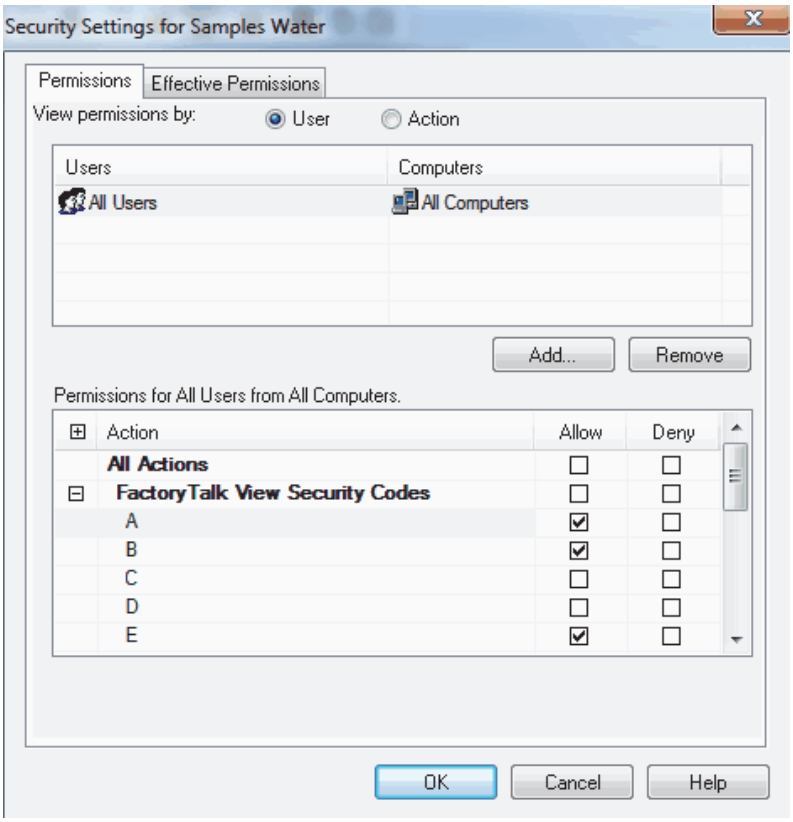
1. From the Settings menu, click Runtime Security.

The Runtime Security dialog box appears.



2. Click Security Accounts.

   The Security Settings dialog box appears.



3. Click Add to select an existing user or user group from the Select User and Computer dialog box.

   The selection appears under the Users and Computers columns at the top of the Security Settings dialog box.

4. Click the Allow checkbox beside the FactoryTalk View Security Codes that you want to allow permission for the selected account.

   To select all of the codes A...P, click the Allow checkbox for All Actions or the checkbox next to FactoryTalk View Security Codes.

IMPORTANT	Recommended settings for the Library do not use Deny, which takes precedence over an explicit Allow.

5. Click OK.

6. Repeat step 3 through step 5 for each user or group account that you want to set up with Runtime Security.

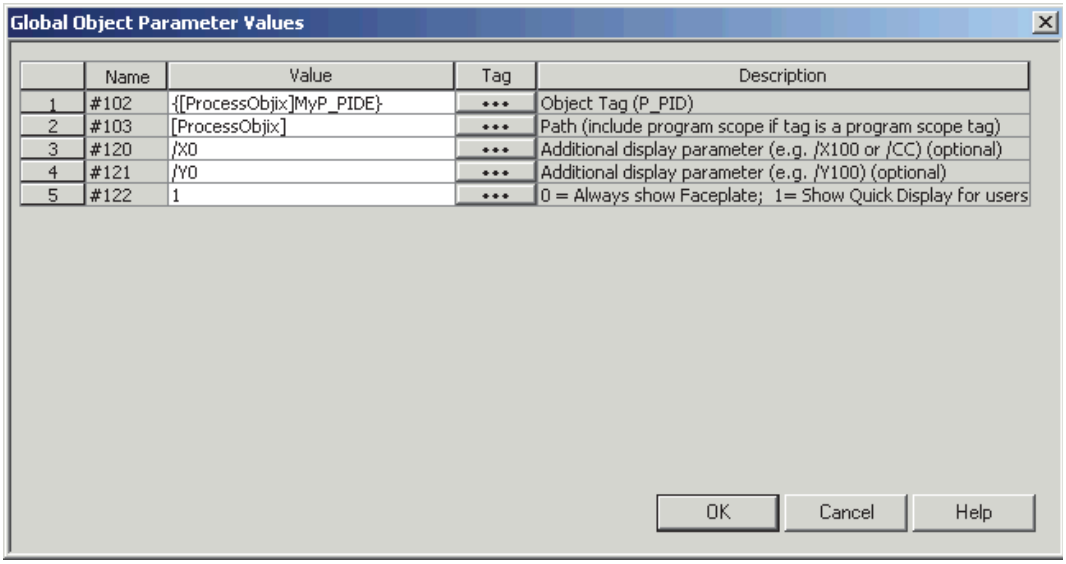
Global Object Configuration

Graphic displays, which consist of display elements (graphic objects), provide an operator with a visual, run-time outlook on a plant activity. The displays show system or process data, and provide operator's with a way to write values to external devices, such as a controller.

For details on using graphic objects, see page 113.

/X and /Y Positioning

The Global Object Parameter Values dialog box, which is used for specifying faceplate coordinates, has two optional parameters: #120 and #121.



Parameters #120 and #121 let you specify additional display command switches. These additional switches can be used to specify where the faceplate appears on your screen.

For example, '/x150 in parameter #120 and '/Y50' specify the faceplate appears 150 pixels from the left and 50 pixels from the top. These parameters are optional and do not need to be specified.

Table 36 and Table 37 describe the position parameters and how to use them with the #120 and #121 global object parameters, respectively.

Table 36 - Corner and Center Positioning

Global Object Parameter	Position Parameter	Description
#120	/Q1	Top, right corner
	/Q2	Top, left corner
	/Q3	Bottom, left corner
	/Q4	Bottom, right corner
	/CT	Centered on top edge
	/CB	Centered on bottom edge
	/CL	Centered on left edge
	/CR	Centered on right edge
	/CC	Center of screen
#121	Leave blank. See the tip.	

TIP The values of global object parameters #120 and #121 are passed directly to the Display command that opens the faceplate's display (.gfx). Nothing prevents you from putting any valid Display command parameters in these values. For more information, see the FactoryTalk View Site Edition User's Guide, publication VIEWSE-UM006, or the online Help for the Display command parameters in the FactoryTalk View Studio software.

Table 37 - Absolute (/X, /Y) Positioning

Global Object Parameter	Position Parameter	Description
#120 (SE)	/Xnnn	Left edge of faceplate nnn pixels from the left edge of the screen
#120 (ME)	Numeric value (for example, 100)	
#121 (SE)	/Ynnn	Top edge of the faceplate nnn pixels down from the top edge of the screen
#121 (ME)	Numeric value (for example, 100)	

Parameter #122 controls the display that appears when you click the display element. You can expand the row height for parameter #122 by typing a value and clicking Enter.

- '0' specifies the full faceplate always appears.
- '1' specifies the full faceplate appears if the user has security code 'C' (see page 105), and the 'Quick' display (see page 109) appears if you do not have security code 'C'.
- '2' specifies the 'Quick' display always appears.
- If parameter #122 is left blank, the full faceplate always appears.

See the 'Use Global Objects' section in each Add-On Instruction Reference Manual to configure these parameters, if applicable.

Quick Display

Most library objects that have faceplates have 'Quick' displays as well. The Quick display is much smaller than the PlantPAx faceplate, but still has all of the information and controls needed by the operator.

Clicking the View Faceplate icon  shows the full faceplate.

Saving Your Data in FactoryTalk View SE Software

When entering data into string Input fields in FactoryTalk View SE software, the data is not saved to the tag until you press Enter. When the Input field is active, its border changes based on the state of the input:

- When the Input field is active (the cursor is in the field), the Input field border is a solid line.

EXAMPLE Active input field (with cursor)

- If you modify the data in the Input field and move to a different field without pressing Enter, the border remains a solid line. This indicates that the data has not been saved to the tag.

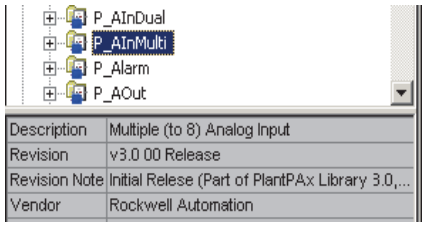
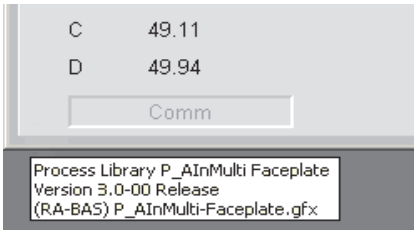
EXAMPLE Data entered but not saved

- If the data in the Input field has not changed or has been written to the controller tag, the border is a dashed line.

EXAMPLE Data entered and saved

Maintain Library Releases

Each library object has a revision xx.yy-zz where: xx is the Major Revision number, yy is the Minor Revision number, and zz is the service release. Each release of the Library comes with release notes describing the changes made since the last release.

Component	Example
The Add-On Instruction in RSLogix 5000 software has revision information visible when the instruction is selected in the Controller Organizer.	
The faceplate in FactoryTalk View software has revision information visible when the pointer is paused just inside the lower left corner of the faceplate when accessed in a running HMI Client.	

The instruction and faceplate are compatible if they have the same Major and Minor Revision numbers. In general, service releases can be loaded into your application with little impact, although we suggest that you review the release notes to make sure you understand the changes made in the release and assess the impact to your application.

Using two different Major/Minor revisions of the Library in a single application is not supported. When updating major or minor releases, there are several considerations to take into account:

- The new revision can have differences in functionality that requires adjustment of your application code written around the library instruction instances.
- There can be differences in the look-and-feel that requires an update of operator training and manuals.
- If you have made any customizations, they likely need to be repeated on the new release of the library after it is loaded.

It is important that these considerations are weighed properly against the benefits you gained from the new Library release. The release of the Library typically includes documentation on how to upgrade from the previous Major/Minor release.

Customize the Library

The Rockwell Automation Library of Process Objects can be customized for project or customer-specific reasons. However, we recommend that you consider the following:

- Weigh the value of the customization against the value of using the library as issued. For example, you must document any customization so it can be reapplied, if necessary, on top of a subsequent maintenance release.
- Rockwell Automation provides a varied schedule of maintenance releases for the library. If the library object has not been altered, updating the latest maintenance release can be done more easily. If customized, the customization has to be reapplied manually on subsequent releases.

IMPORTANT There are some library objects, such as Logix Diagnostic objects and Steam Table instructions, that are not to be customized. These objects typically are not operator-facing and have functionality that must be fixed to provide the desired functionality.

- Customization could inhibit your ability to leverage the library documentation or standard training based on the library.
- The library is supported through Rockwell Automation Technical Support as long as the Add-On Instructions have not been modified from the original deployment. If customized, the library is supported similar to any other application code.

Notes:

Use the Library

This chapter shows how to use Library Add-On Instructions, global objects, and faceplates to build your control application. FactoryTalk View SE software provides visualization of instruments connected to the network interface.

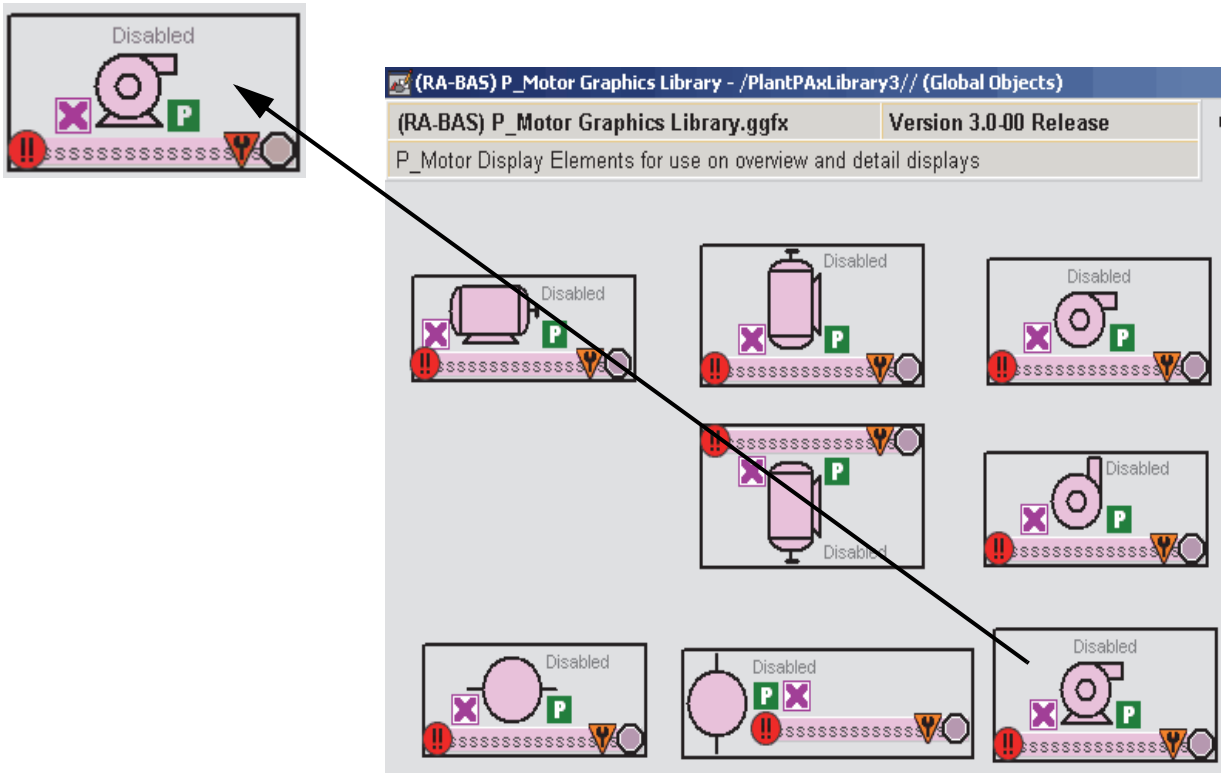
Global objects (display elements) provide touch areas from which faceplates are launched, link tag names to the faceplates, and display the process variables and alarms.

The procedures in this section require that you have installed global objects (.ggfx file types). See Import Visualization Files on page 72 for installation details, if necessary.

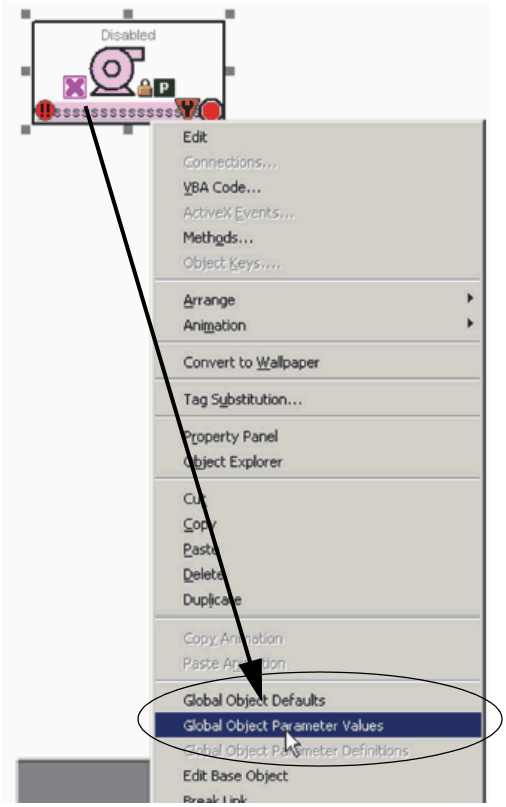
Create a Library Instance

Follow these steps to configure a library instance.

1. In the FactoryTalk View Studio software program, open the global objects (.ggfx) file that contains the graphics library for the instruction.
2. Click and drag a global object onto the Studio software display file.



3. In the display file, right-click the global object file and choose Global Object Parameter Values.

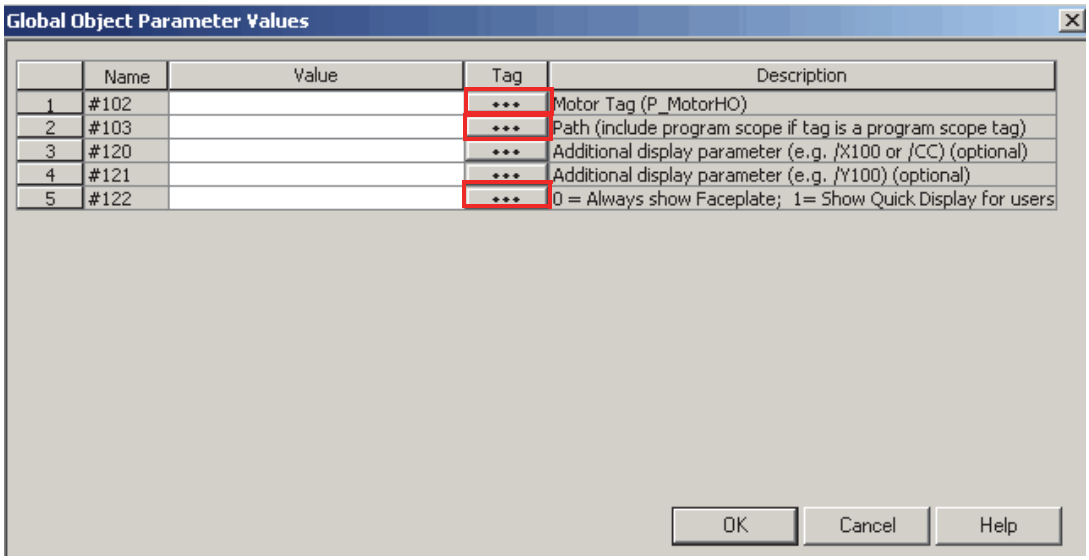


The Global Object Parameter Values dialog box appears.

You need to configure the first, second, and fifth parameters.

The third and fourth parameters are 'optional' and differ in syntax depending on whether using a FactoryTalk SE or ME application.

See page 107 for details.



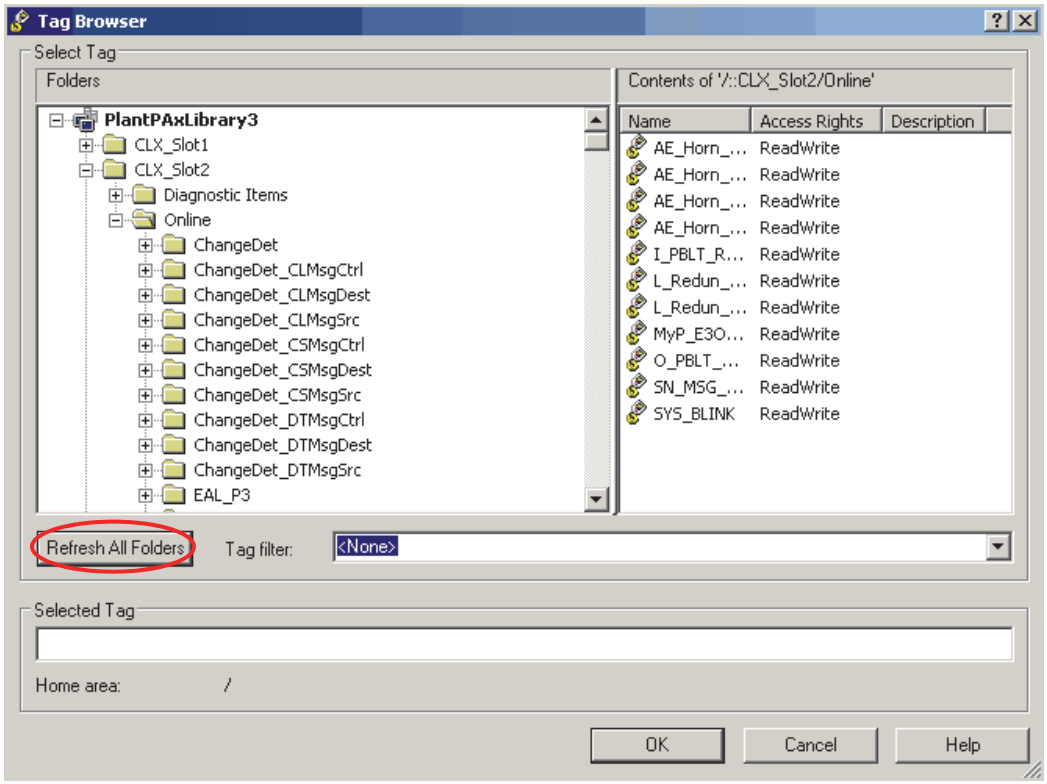
The global object parameters are as follows.

Parameter	Required	Description
#102	Y	Object tag to point to the name of the associated object Add-On Instruction in the controller.
#103	Y	Path used for display navigation features to other objects. Include program scope if tag is a program scope tag.
#120	N	Additional parameter to pass to the display command to open the faceplate. Typically used to define position for the faceplate.
#121	N	Additional parameter to pass to the display command to open the faceplate. if defining X and Y coordinate, separate parameters so that X is defined by #120 and Y is defined by #121. This lets the same parameters be used in subsequent display commands originating from the faceplate.
#122	Y	These are the options for the global object display: 0 = Always show faceplate 1 = Show Quick Display for users without Maintenance access (Code C) 2 = Always show Quick Display

4. To enter the backing tag for the #102 parameter, you can type a tag into the Value column or click the Browse button  .

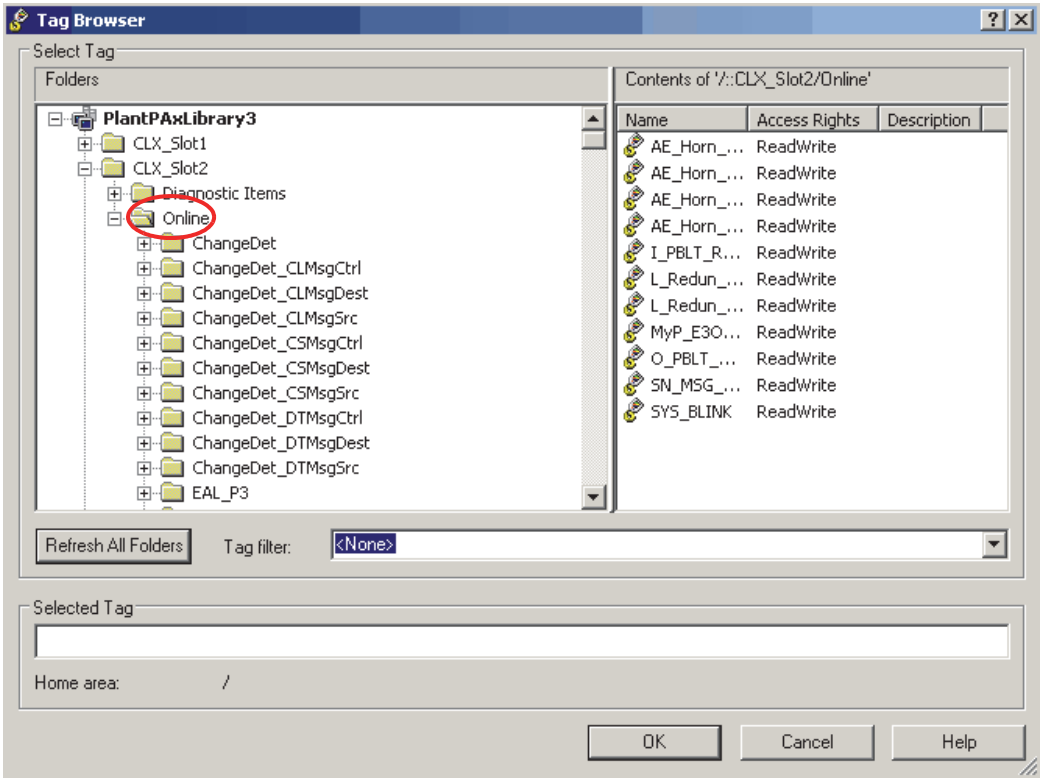
   For the following procedures, click Browse to enter the first parameter object tag.

   The Tag Browser dialog box appears.



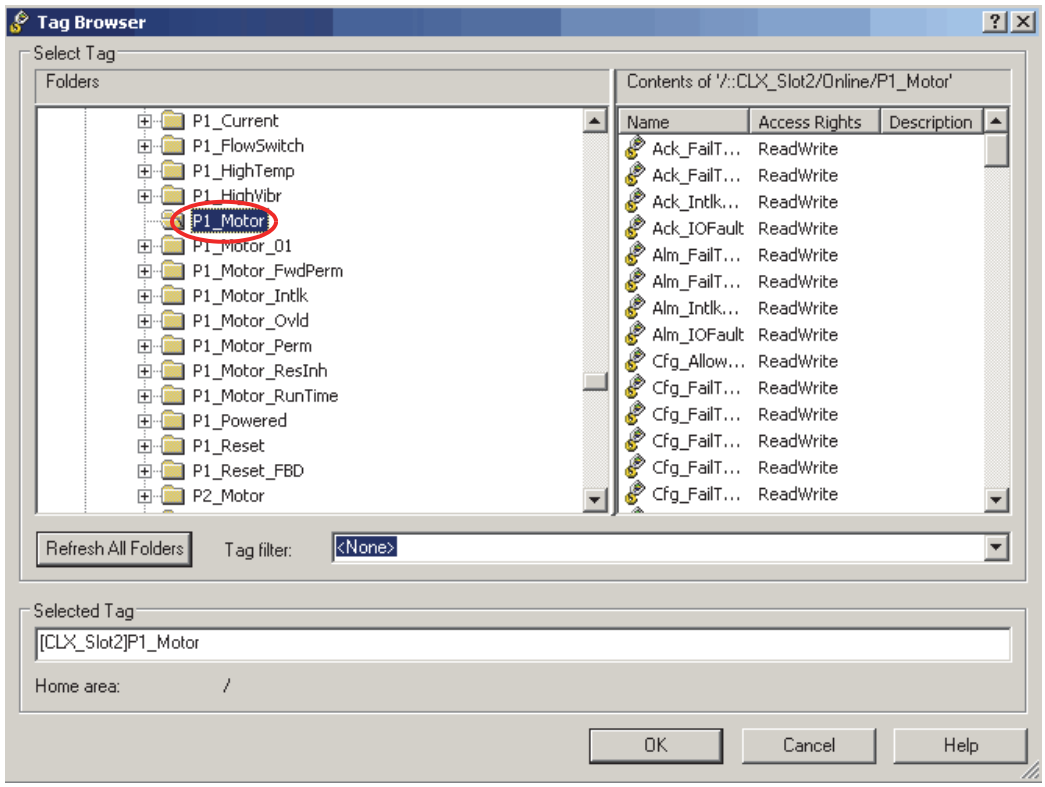
5. Click Refresh All Folders.

6. Expand the shortcut for the project controller and then click Online.



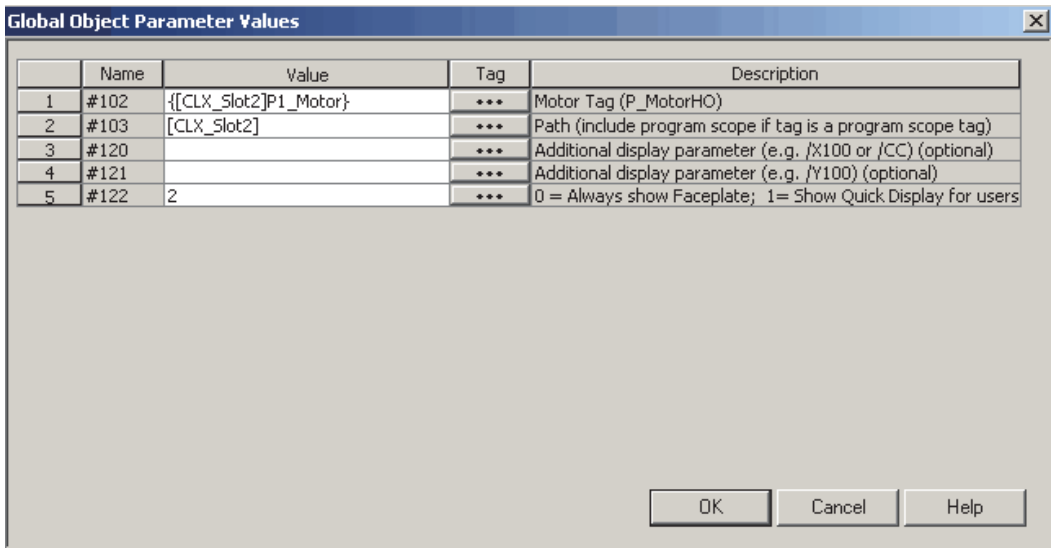
7. Expand and scroll down the Online menu to select the tag for the Add-On Instruction.

   The backing tag for the Add-On Instruction is a folder.



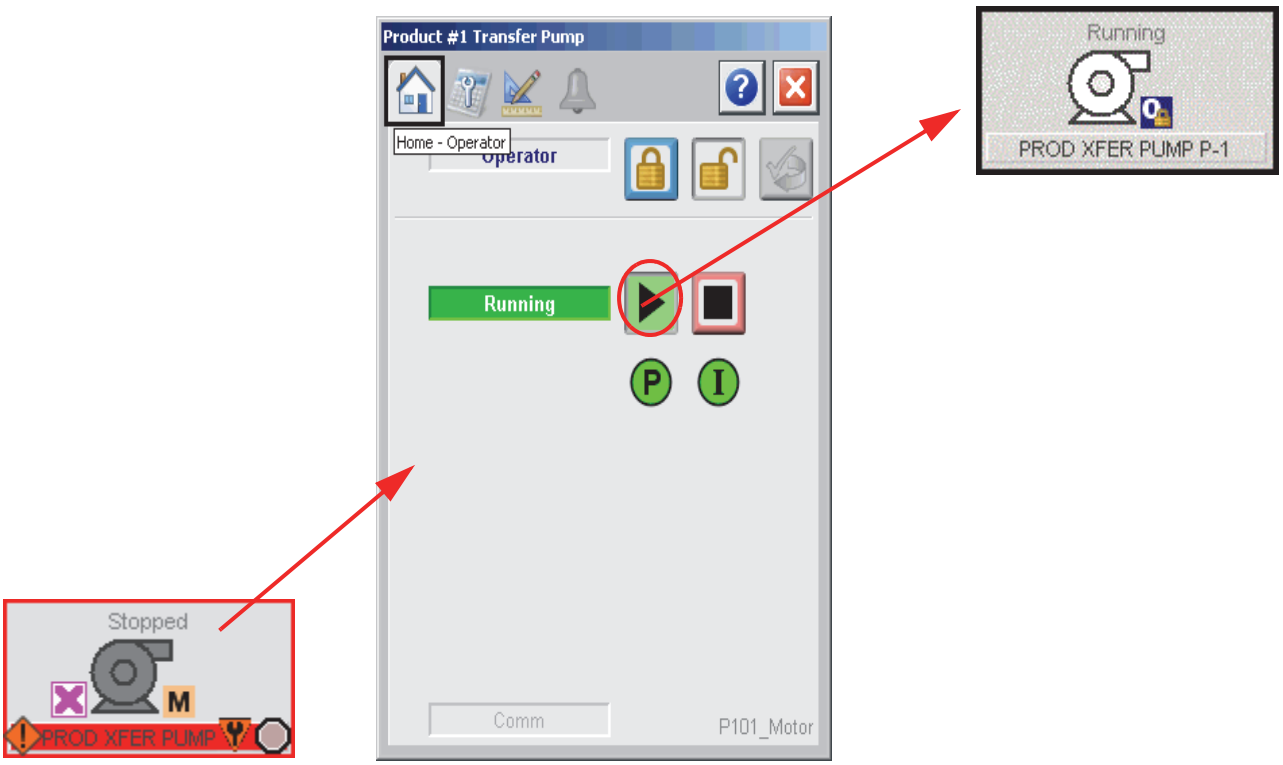
8. Click OK.
9. To configure the #103 parameter, copy the path part of #102 into the Value column:
   - No { } (curly braces)
   - No tag
   - Just /:[ ] (area and shortcut)
10. Click the Value column for the fifth parameter and enter a value (0, 1, 2) per the desired display.

    This is an example of parameter values.



11. Click OK.
12. Click Save.

13. On an HMI screen, access the display that contains the global object.
14. Click the global object to access a faceplate.



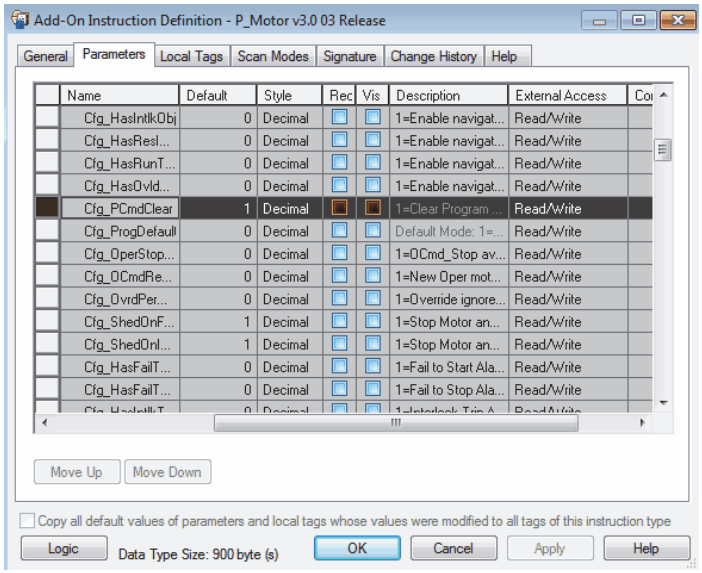
15. Click command buttons on the faceplate to operate the device.

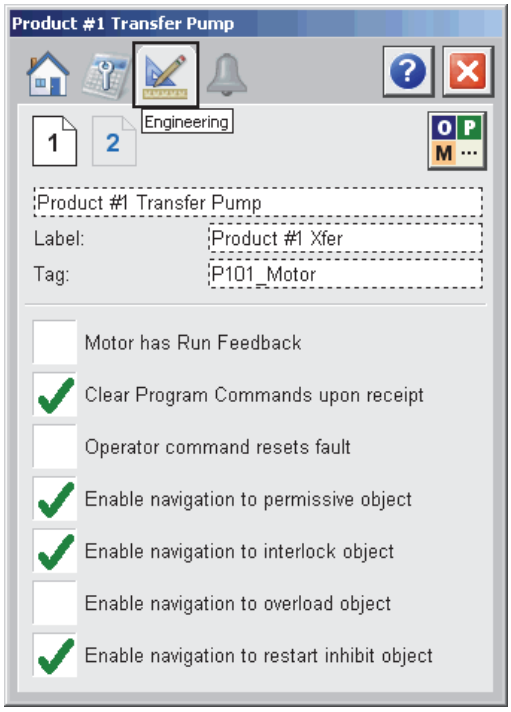
IMPORTANT See Appendix C to change the color on a display element or faceplate.

Device Configuration

You can configure the device parameters by doing one of the following steps:

- Type values in the Parameters tab of the RSLogix 5000 software program
- Check a checkbox for each command on the Engineering tab of the instruction's HMI faceplate.





Notes:

Online Configuration Tool

The Online Configuration Tool is a stand-alone, Microsoft Excel-based spreadsheet. The multi-tabbed spreadsheet lets you access configuration tags of Library objects (Add-On Instruction parameters and local tags) for multiple instances of a library instruction simultaneously by using OPC.

IMPORTANT The Configuration Tool is supported **only** in 32-bit editions of Microsoft Excel software because the tool uses RSLinx® Classic OPC/DA for its online communication, and RSLinx Classic OPC Server is a 32-bit application. (The tool works fine on 64-bit Microsoft Windows operating systems; it requires a 32-bit installation of Microsoft Excel/Microsoft Office.)

This tool lets you make bulk changes more easily, especially for local configuration tags like strings, instead of modifying each tag separately for each Add-On Instruction instance.

WARNING: The spreadsheet is used to modify parameters by using a controller. However, the controller **must not** be in a production environment controlling machinery or processes. The spreadsheet works with the controller in Program mode, so we strongly suggest it be used that way to avoid unintended control changes to running equipment.

Before You Begin

The spreadsheet uses RSLinx Classic software DDE/OPC, so you need to have a suitably licensed copy (OEM, Gateway) of the software. You cannot use the Lite version of RSLinx Classic software.

We suggest that you make a back-up copy of your application before following the steps below. When using the spreadsheet, all of your Add-On Instruction instances (backing tags) are to be created in your RSLogix 5000 software project (.acd file) and the project downloaded to your controller.

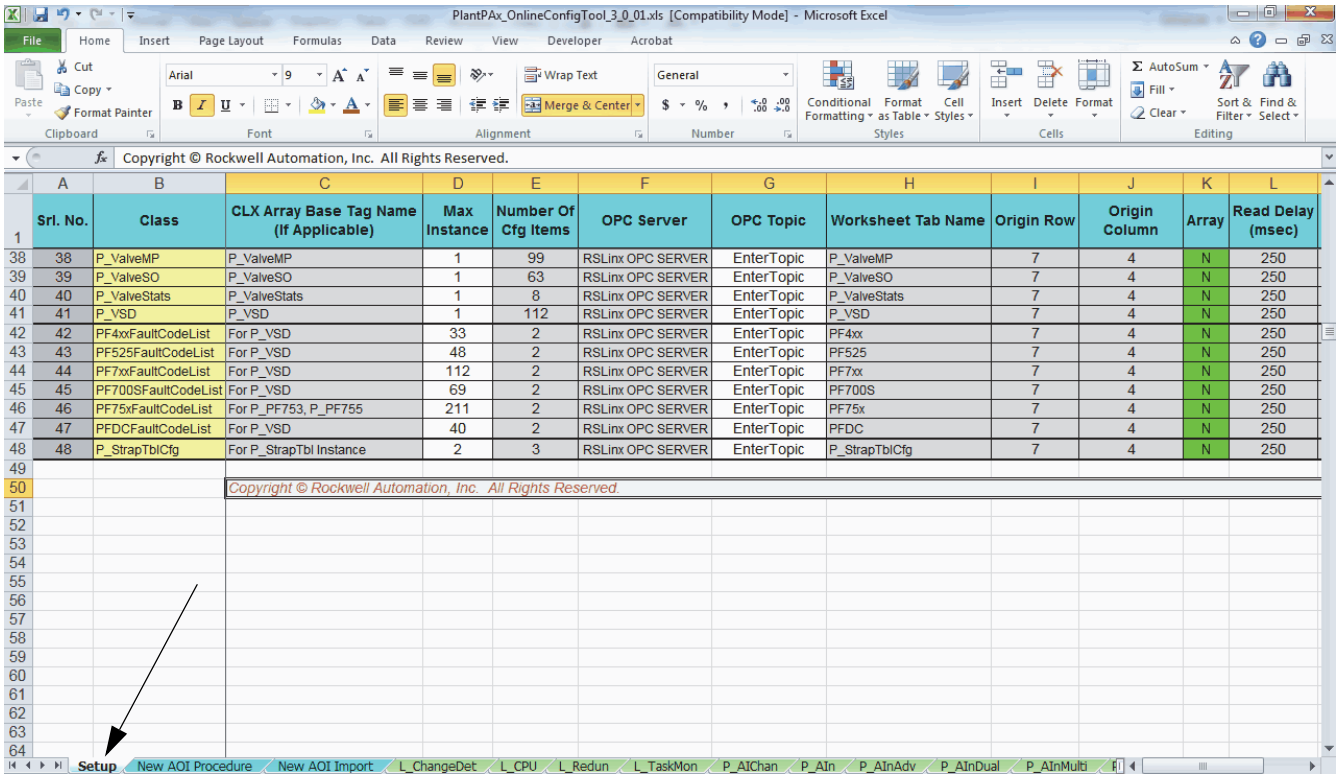
Configure Parameters By Using a Spreadsheet

Local tags can be configured through the HMI faceplates or in RSLogix 5000 software by opening the instruction logic of the Add-On Instruction instance and then opening the Data Monitor on a local tag.

The following procedure is for using a spreadsheet for uploading (saving) and downloading (restoring) the configuration (.Cfg) parameters from library instances in an online controller. See the Warning above that applies for the controller that is being used before starting these steps.

1. Download the Rockwell Automation Library from the Product Compatibility and Download Center at http://www.rockwellautomation.com/rockwellautomation/support/downloads.page.
2. Open Tools & Utilities and double-click PlantPAx Online Configuration Tool.
3. Open the Excel spreadsheet.

   The spreadsheet opens in the default Setup folder (at the bottom of the screen).

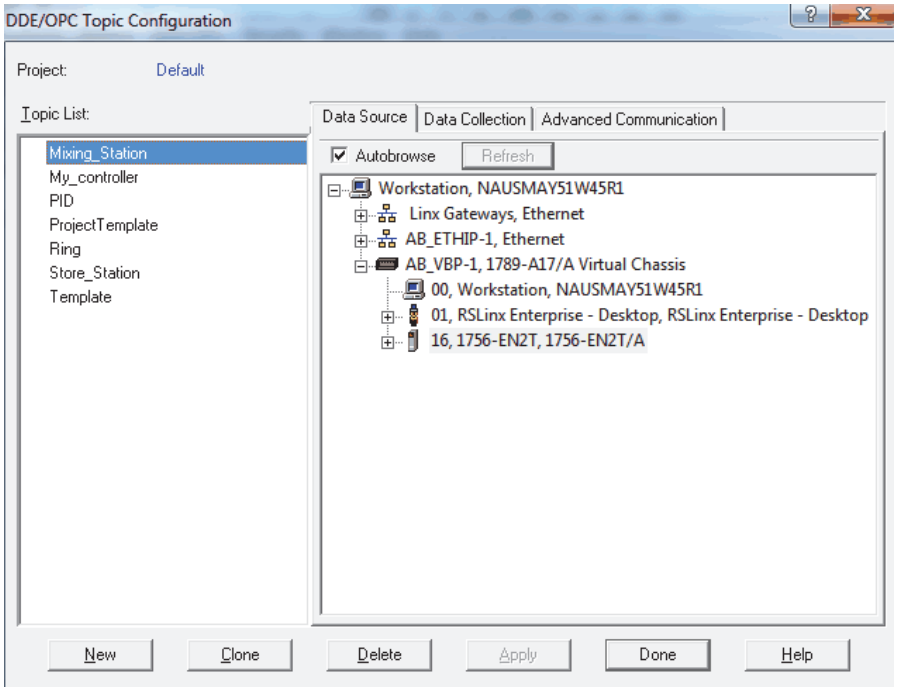


	A	B	C	D	E	F	G	H	I	J	K	L
1	Srl. No.	Class	CLX Array Base Tag Name (If Applicable)	Max Instance	Number Of Cfg Items	OPC Server	OPC Topic	Worksheet Tab Name	Origin Row	Origin Column	Array	Read Delay (msec)
38	38	P_ValveMP	P_ValveMP	1	99	RSLinx OPC SERVER	EnterTopic	P_ValveMP	7	4	N	250
39	39	P_ValveSO	P_ValveSO	1	63	RSLinx OPC SERVER	EnterTopic	P_ValveSO	7	4	N	250
40	40	P_ValveStats	P_ValveStats	1	8	RSLinx OPC SERVER	EnterTopic	P_ValveStats	7	4	N	250
41	41	P_VSD	P_VSD	1	112	RSLinx OPC SERVER	EnterTopic	P_VSD	7	4	N	250
42	42	PF4xxFaultCodeList	For P_VSD	33	2	RSLinx OPC SERVER	EnterTopic	PF4xx	7	4	N	250
43	43	PF525FaultCodeList	For P_VSD	48	2	RSLinx OPC SERVER	EnterTopic	PF525	7	4	N	250
44	44	PF7xxFaultCodeList	For P_VSD	112	2	RSLinx OPC SERVER	EnterTopic	PF7xx	7	4	N	250
45	45	PF700SFaultCodeList	For P_VSD	69	2	RSLinx OPC SERVER	EnterTopic	PF700S	7	4	N	250
46	46	PF75xFaultCodeList	For P_PF753, P_PF755	211	2	RSLinx OPC SERVER	EnterTopic	PF75x	7	4	N	250
47	47	PFDCFaultCodeList	For P_VSD	40	2	RSLinx OPC SERVER	EnterTopic	PFDC	7	4	N	250
48	48	P_StrapTblCfg	For P_StrapTbl Instance	2	3	RSLinx OPC SERVER	EnterTopic	P_StrapTblCfg	7	4	N	250
49												
50			Copyright © Rockwell Automation, Inc. All Rights Reserved.									

4. Name and save the file that you want to use for the controller.
5. Open RSLinx Classic software.

6. From the DDE/OPC menu, choose Topic Configuration.

   The DDE/OPC Topic Configuration dialog box appears.



7. In the left pane, find the Topic that points to your controller or create a new one.
8. Open your new spreadsheet file.

   Use the active content if you get a warning message.
9. Type the Topic name in all of the rows of column G of the Setup sheet.

   Be sure you are going to the correct controller.
10. Open an Add-On Instruction instance by clicking the respective tab name at the bottom of the screen.

11. In column C, starting in row 10, type the backing tag names for your Add-On Instruction instances.

	A	B	C	D	E
1		P_DIn: Discrete Input	Read From CLX:		
2			Send To CLX:		
3					
4			Description:		Description for display on HMI
5			Usage:		Local
6			Data Type:		STRING_40
7				(Origin)	.Cfg_
8	Unit	Tag Description	TagName	Instance	Desc
9			Default Values:	0	Discrete Input
10			FSL_101	1	
11			TSH_102	2	
12				3	
13				4	
14				5	
15				6	
16				7	
17				8	

12. On the Setup sheet, type the number of instances of each Add-On Instruction in column D.

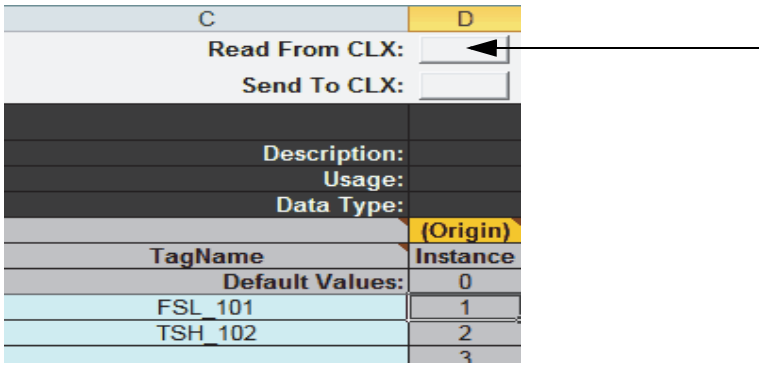
IMPORTANT Make sure you switched to the Setup sheet for step 12. Do **not** change the 'Number of Cfg Items' in Column E or any of the other data on the Setup sheet.

The example below shows 2 instances entered on the Setup sheet because we made 2 modifications in column C in the P_DIN instruction folder.

fx 1

	A	B	C	D	E
1	Srl. No.	Class	CLX Array Base Tag Name (If Applicable)	Max Instance	Number Of Cfg Items
4	3	L_Redun	L_Redun	1	37
5	4	L_TaskMon	L_TaskMon	1	15
6	5	P_AIChan	P_AIChan	1	48
7	6	P_AIn	P_AIn	1	96
8	7	P_AInAdv	P_AInAdv	1	148
9	8	P_AInDual	P_AInDual	1	138
10	9	P_AInMulti	P_AInMulti	1	145
11	10	P_AOut	P_AOut	1	46
12	11	P_D4SD	P_D4SD	1	85
13	13	P_DIn	P_DIn	2	24
14	14	P_DoseFM	P_DoseFM	1	73
15	15	P_DoseWS	P_DoseWS	1	72
16	16	P_DOut	P_DOut	1	69
17	17	P_Fanout	P_Fanout	1	110
18	18	P_HiLoSel	P_HiLoSel	1	46
19	19	P_Intlk	P_Intlk	1	100

13. Go back to the Add-On Instruction sheet, in our example it's P_DIn, and click the Read From CLX box near the top of the spreadsheet.



14. Click Yes to the overwrite message to populate the row for that instance.
15. Make your modifications, for example change the strings, and click the Send To CLX to send the new configuration to the controller.
16. Check the data in the controller and save the project (to a new .acd file is best).

    Make sure you upload tag values when you save the project.

TIP If your Add-On Instruction instances are in Program-scope tags, you can get to those as well. The TagName syntax is:

Program:<program_name>.<tagname>

Notes:

Alarms Builder Tool

This section provides basic procedures for using the Alarms Builder tool to expedite creating alarms in the PlantPAx system.

The stand-alone tool is used with FactoryTalk View software for creating the following types of alarm systems:

- Tag-based (polling) Alarm and Event (AE) server (page 128)
- FactoryTalk View Machine Edition (ME) HMI alarms (page 148)

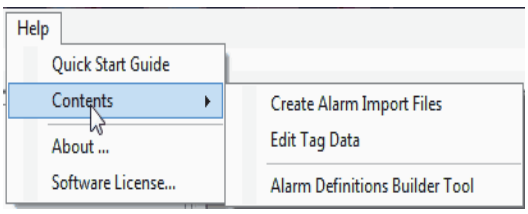
The Alarms Builder tool uses the data types in an RSLogix 5000 project and associates a FactoryTalk View Site Edition (SE) HMI faceplate display to each Logix alarm definition. The tool automatically configures a SE command with the alarm tag when the import files are built.

IMPORTANT If you work only with FactoryTalk View ME software, you can associate the controller to a ME application. The ME alarm system does not provide the functionality to execute FactoryTalk View commands.

Before You Begin

The Alarms Builder software is among the Library of Process Objects files that must be downloaded. The library zip files are available from the Product Compatibility and Download Center at http://www.rockwellautomation.com/rockwellautomation/support/downloads.page.

You also must have RSLogix 5000 software installed. RSLogix 5000 services, which are part of the software installation package, converts a controller ACD file to an XML file so the Alarm Builder software can read the data to create the alarm tags.



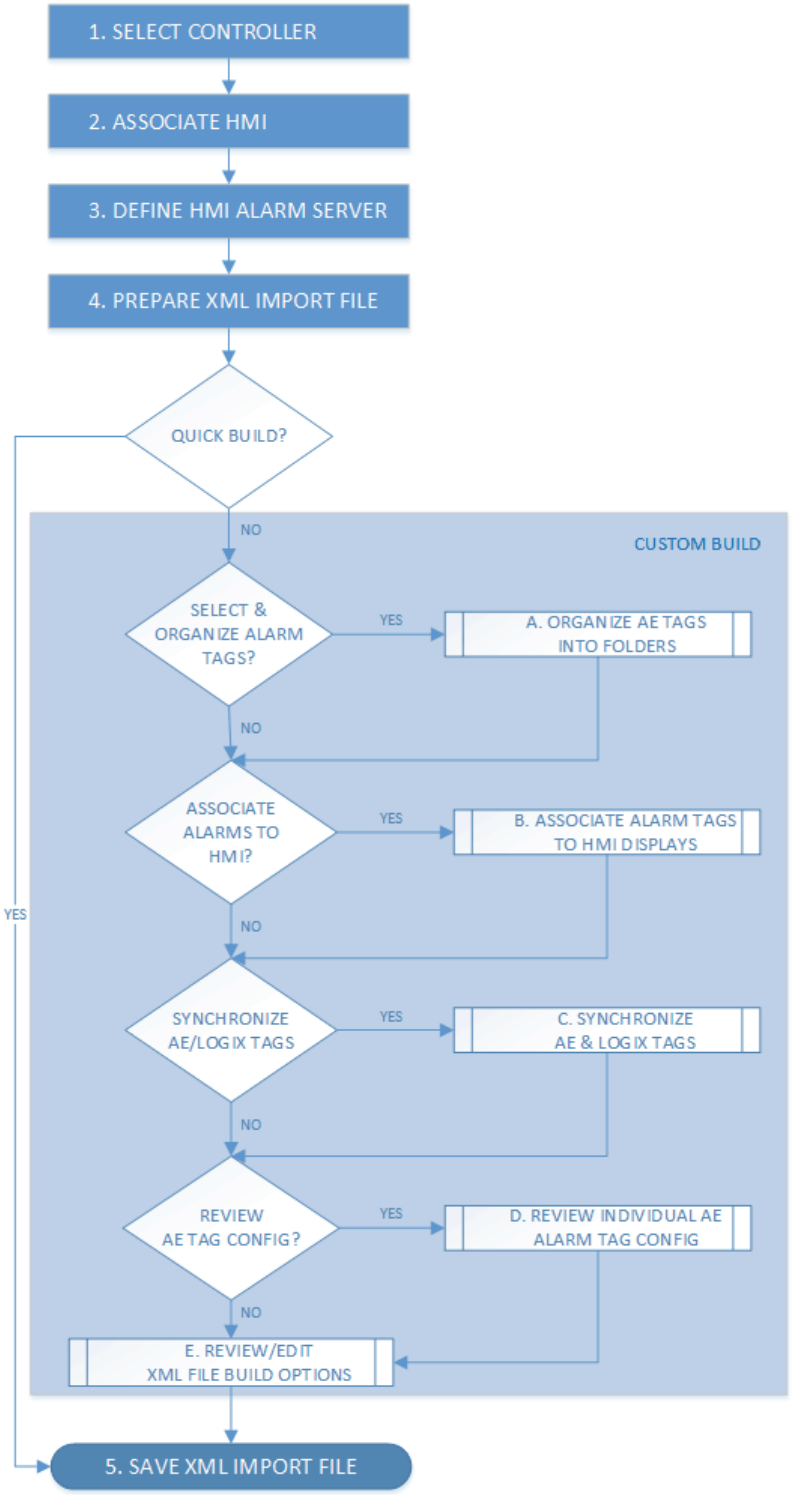
There are additional resources available that provide detailed information for configuring alarm tags and data types. From the Help menu on the PlantPAx Alarms Builder and Tag Data Edit Tool window, click Contents to access these documents:

- Create Alarm Import Files
- Edit Tag Data
- Alarm Definitions Builder Tool

Build AE Alarms

The diagram outlines the procedures for creating FactoryTalk View SE software alarms. The procedures in this section are in the same order as the headings in the diagram.

Figure 7 - Alarm Builder AE Workflow



1. Select Controller

This section describes how to associate a Logix controller to FactoryTalk View HMI servers and data servers so the Alarm Builder tool can create HMI alarm tags. The tool obtains server information from specified FactoryTalk View directories, which use default FactoryTalk View installation settings. The HMI server and data server information can be entered manually, if necessary.

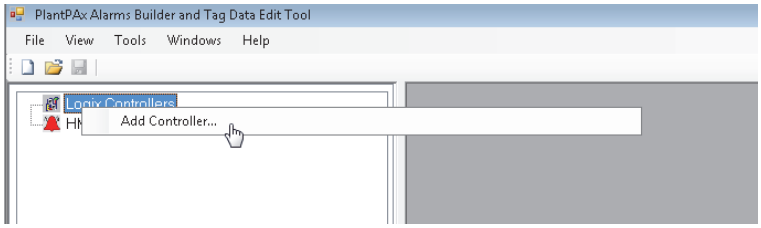
Follow these steps to add a controller to the Alarms Builder software tool.

1. From the Tools & Utilities folder in the downloaded library files, open the Alarms Builder tool.

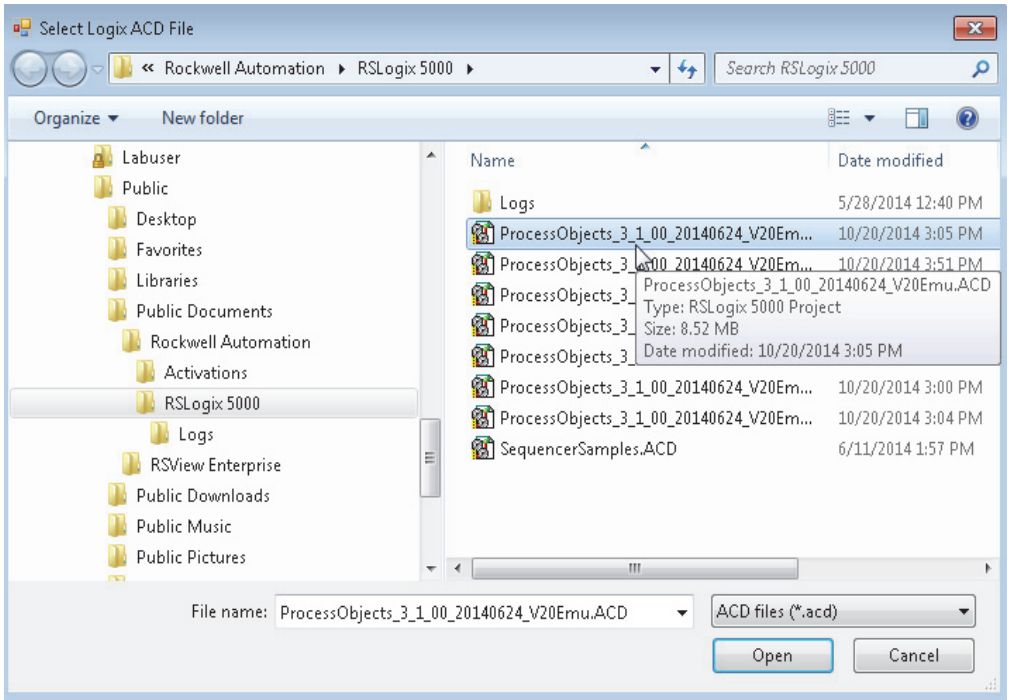
   A Quick Start PDF file opens in a separate Adobe PDF Reader window. Review the Quick Start guide and close the Adobe window.

   The PlantPAx Alarms Builder and Tag Data Edit Tool window appears.

2. Right-click Logix Controllers and choose Add Controller.



   The Select Logix ACD File dialog box appears.



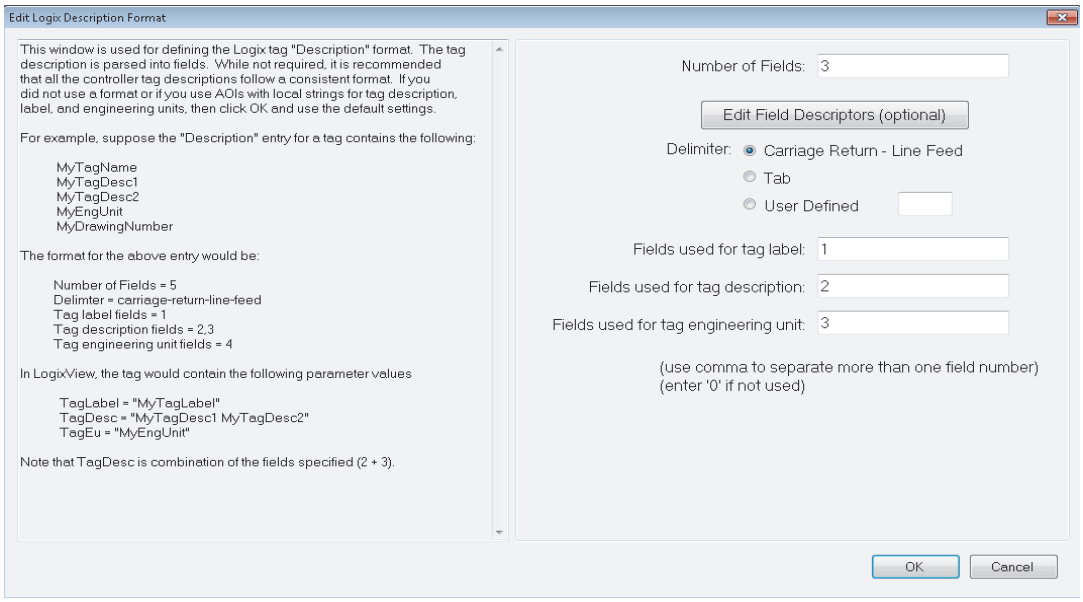
3. Navigate to the folder that stores the controller's project file.
4. Select the ACD file and click Open.

A message window displays if the tool cannot determine the RSLogix 5000 software version of the selected ACD file.

5. Type the RSLogix 5000 software version and click OK if prompted. Otherwise, proceed to step 6.

6. On the Edit Logix Description Format dialog box, click OK to use the defaults if the following conditions apply:

   - You do not use a format for describing controller tags
   - You use Add-On Instructions with local strings that describe the tag label and engineering units



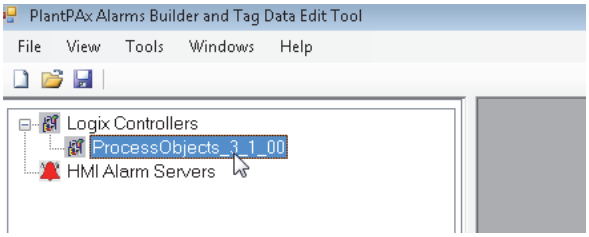
7. If the above conditions do **not** apply, type data into the text boxes to describe the tags and click OK.

   A dialog box appears to update severity values if you are replacing Version 3.0 Add-On Instructions with Version 3.1 instructions.

   See Replace Logix Tag Severity Values on page 131 for details.

   TIP The conversion of the controller project file to an XML format can take several minutes for each controller file.

   When the conversion is complete, the controller file is added to the configuration tree under Logix Controllers.



8. Repeat step 2 through step 5 to add multiple controllers to the project.

   Multiple controllers can be associated to different HMI alarm servers. For example, you can have an AE alarm server for several controllers in a SE project and a ME alarm server for skid-mounted controllers.

Replace Logix Tag Severity Values

Use the procedures in this section if you have upgraded an ACD file with Rockwell Automation Library Add-On Instructions, Version 3.0, to Version 3.1 and want to change the severity values. The severity values for the Version 3.1 instructions are now a range as shown in Table 38.

Table 38 - Logix Tag Severity Values

Version 3.0 and earlier		Version 3.1		Definitions	Alarm Color
Logix Severity	**A&E Severity**	**Logix Severity**	**A&E Severity**		
1	1	1…250	1…250	Low	Blue
2	251	251…500	251…500	Medium	Yellow
3	501	501…750	501…750	High	Red
4	751	751…1000	751…1000	Urgent	Magenta

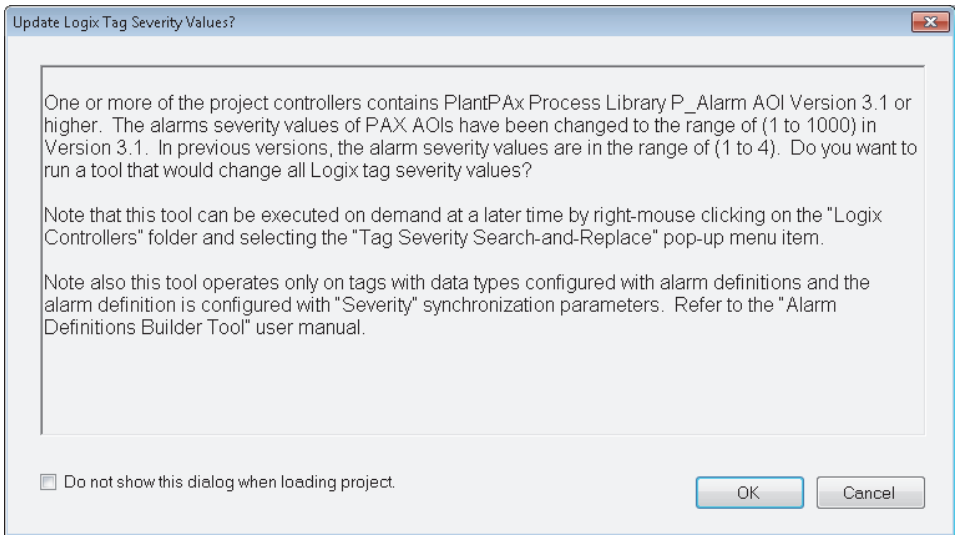
The Version 3.1 instruction range of 1...1000 (INT data type) is consistent with the default AE severity value range.

The Version 3.0 and earlier instructions have a severity range of 1...4 (SINT data type). When an ACD file that contains Version 3.0 instructions is updated with Version 3.1 instructions, the tags retain their existing severity values (1...4).

IMPORTANT For FactoryTalk View software, version 8 and later, the AE severity can be configured by using a Logix tag address. The alarm server reads the severity from the Logix tag during runtime. This function allows the severity value to be changed from the HMI faceplate; no number mapping is required.

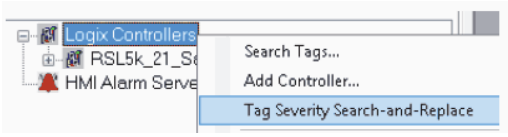
For FactoryTalk View software, version 7 and earlier, the AE severity value is a fixed numeric value. It cannot be changed from the tag faceplate; it can be changed only from the AE tag database.

If you load a controller that has the P_Alarm Add-On Instruction with Version 3.1 or later, a prompt appears to let you use a tool to change severity tag values.



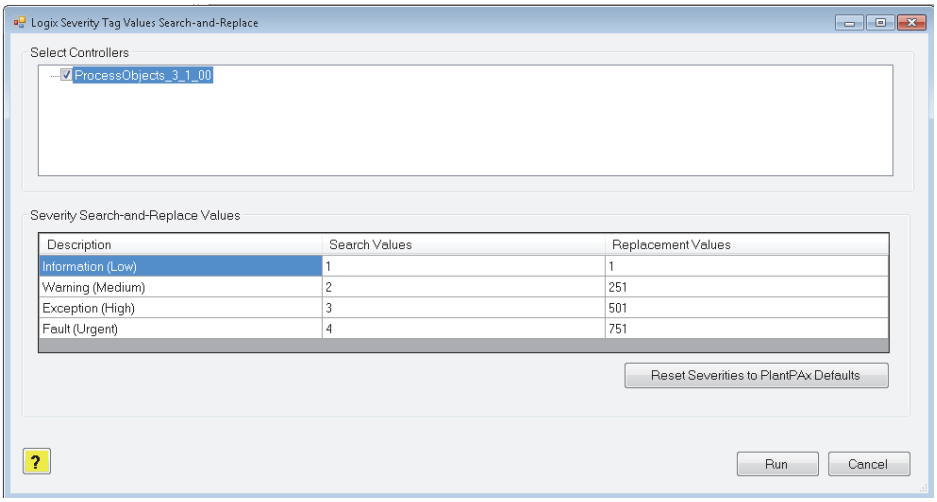
1. Do one of the following:
   a. Click Cancel to run the tool at a later time.

      To reaccess the prompt, right-click Logix Controllers and choose Tag Severity Search-and-Replace from the pull-down menu.



   b. Click OK to replace the severity values.

      The Replacement Severity Tool dialog box appears.



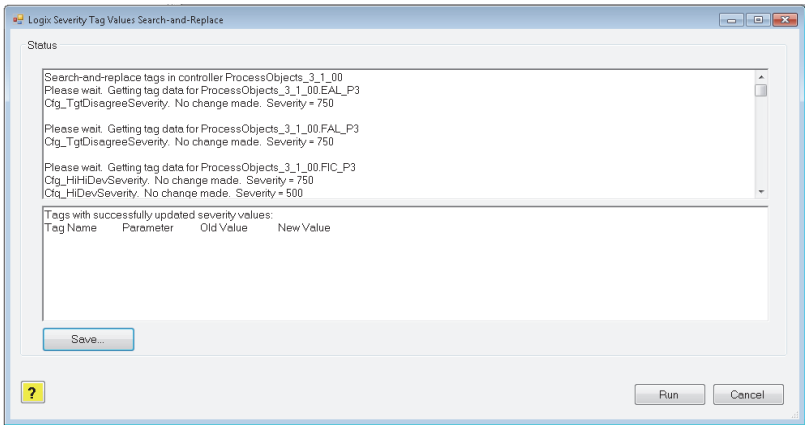
2. In the top box, select a controller.

   Controllers that contain Version 3.1 or later of the P_Alarm Add-On Instruction are automatically selected.

3. In the bottom box, change the values by typing in the respective cells.

4. Click Run.

   Two status boxes appear during the replacement operation.



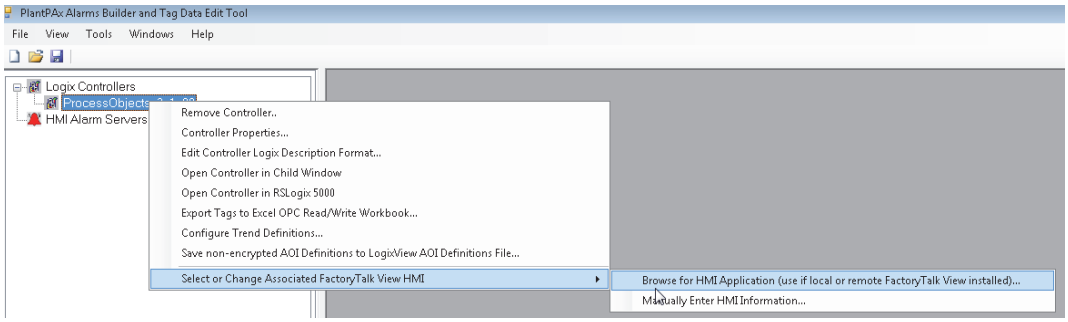
   – The top table provides status and error messages.
   – The bottom table provides a listing of tag values that are changed, if any.

5. Click Save.

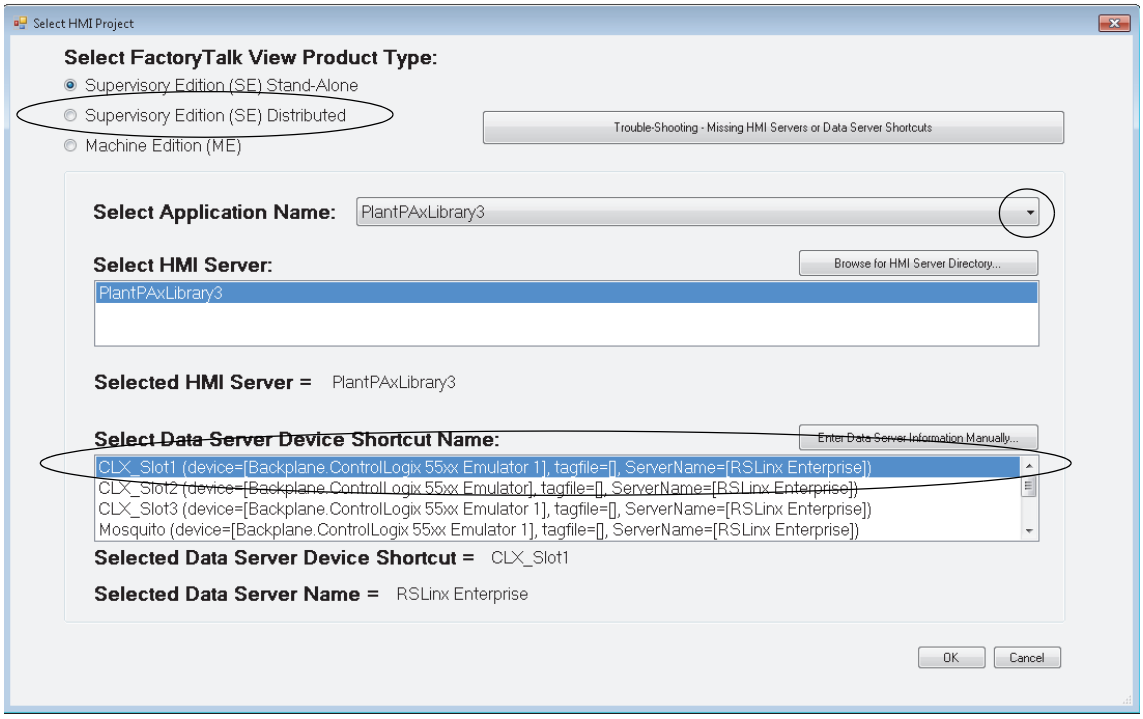
2. Associate HMI

The HMI association provides the information (data area name, device shortcut name) needed in the AE tag addresses.

1. If you have multiple controllers, click the '+' to expand the Logic Controllers node.
2. Right-click on a controller file, and from the pull-down menu choose Select or Change Associated FactoryTalk View HMI>Browse for HMI Application (use if local or remote FactoryTalk View installed).

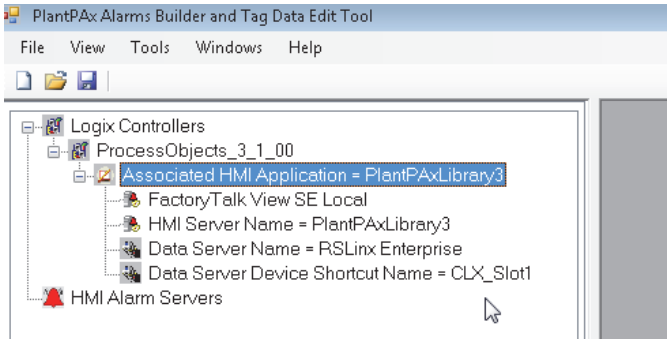


The Select HMI Project dialog box appears.



3. Click the network-scope (Distributed) product type.
4. From the Select Application Name pull-down menu, choose the application name.
5. Click a data server device shortcut name and click OK.

   The associated HMI application with its valid information appears under the selected controller.



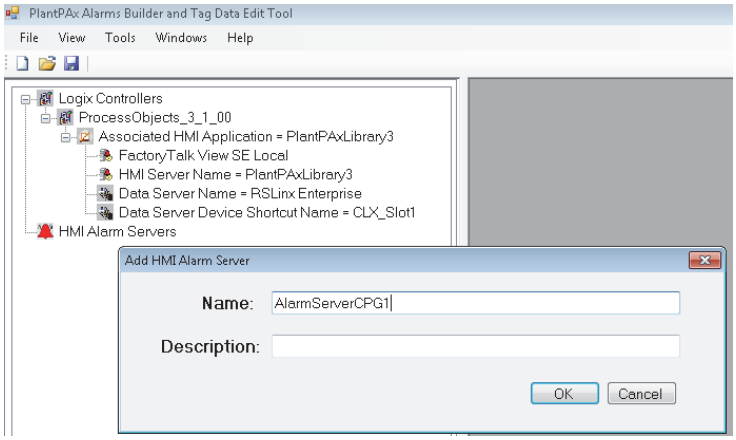
3. Define HMI Alarm Server

This section shows how to add an alarm server to organize the controllers that you want to create AE alarm polling tags. This alarm server has no association to the AE alarm server that you configure in FactoryTalk View Studio software.

Follow these steps.

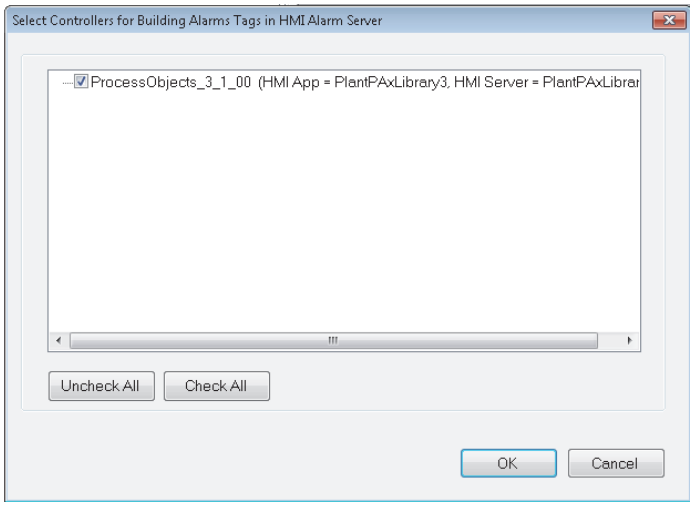
1. In the Alarms Builder configuration tree, right-click HMI Alarm Servers and choose Add Alarm Server.

   The Add HMI Alarm Server dialog box appears.



2. Type a name and a description (optional).
3. Click OK.

   The Select Controllers for Building Alarms Tags in HMI Alarm Server window appears.



4. Check the checkbox to select the controller that you added, and click OK.

   The alarm server name appears under HMI Alarm Servers in the configuration tree.

4. Prepare XML Import File

You have two options to build the XML import file:

- Quick Build
- Custom Build

We recommend that you use the Quick Build default options below to create the XML import files. For optional tag configurations (as shown in the flow chart), see Custom Build on page 138

Quick Build

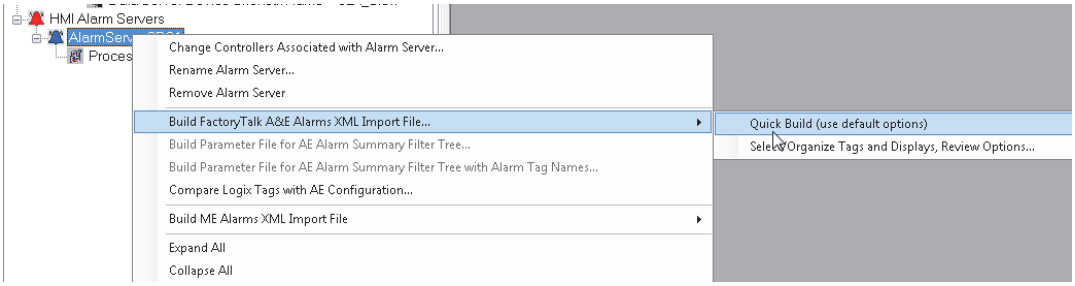
The AE alarms import file uses the following default settings:

- Discrete alarms are only added
- All Logix tags in the alarm server controllers with alarm definitions are used
- AE tag's AckRequired parameter is set to True
- AE tag's Severity value is set from the Logix tag's alarm severity value.

  See Table 38 on page 131 for mapping values.
- AE tag is added to the XML file only if the Logix tag's HasAlarm value is '1'
- Logix tag's Cfg_Desc local string tag value is added to the AE tag's alarm message
- AE tag name equals the Logix tag name and alarm element name. For example, 'TI123_Alm_HiHi'
- AE tag's View command is configured to display the corresponding Logix tag faceplate by using the parameter file passing. The parameter file containing the tag address is automatically added to the HMI project's 'par' folder
- AE tag's Alarm Class parameter is configured with the Logix tag's controller name, program name, tag name, and tag data type
- AE status and control tags for data types are automatically configured by using the PlantPAx P_Alarm Add-On Instruction
- All other AE parameters are configured based on the Logix data type alarm definition

Follow these steps to create the XML import file by using the default settings.

1. Right-click the alarm server name, and from the pull-down menu choose Build FactoryTalk A&E Alarms XML Import File>Quick Build (use default options).



   A dialog box appears that explains the QuickBuild default build options.

2. Click Yes after reading about the settings when the XML file is imported to the AE alarm database.

   The SaveAs AE Alarms Import XML File dialog box appears.

3. Choose where to save the file and type a file name.

   IMPORTANT Choose a file location that is easily accessible, such as your desktop, because you need this XML file to complete the alarms process.

4. Click Save.

   A message box appears that the file is created.

5. Click OK.

   The Alarms Builder software automatically configures the AE tag's View command to display the associated Logix tag faceplate by passing it a parameter file with the tag address. The parameter file is automatically generated by the software.

Custom Build

Custom Build lets you adjust settings by using a series of wizard-like dialog boxes to walk through the build procedure. You have the option to skip certain steps with the dialog boxes, that provide the following procedures:

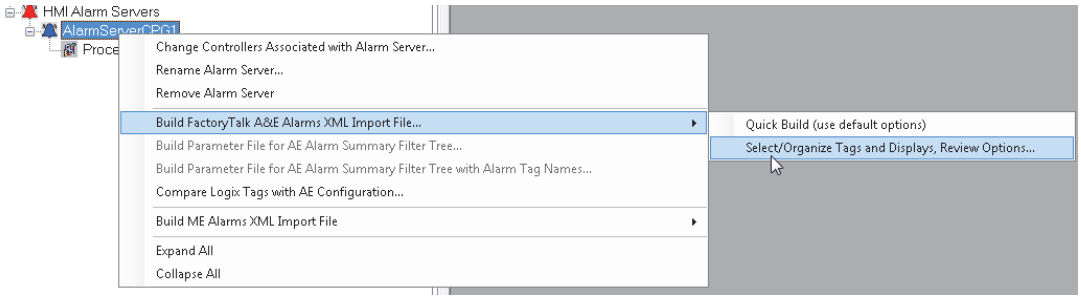
- Select the AE tags to synchronization with Logix tags
- Select which Logix tags to use for building the AE alarms import file (the Quick Build uses all controller tags that have alarm definitions)
- Organize Logix tags in user-defined folders, which can be used as prefixes in the AE tag names. (For example, AE_InAlmUnackedCount)
- Associate HMI displays to Logix tags to open the HMI via a user-configurable macro
- Review tag configuration before adding the tags to the alarms import file

A. Organize AE Tags into Folders

You can select individual Logix tags to use for building the alarm tags. The selected tags also can be organized in user-defined folders. This lets you use the alarm summary to filter alarms based on the folder names in the 'Alarm Class' string.

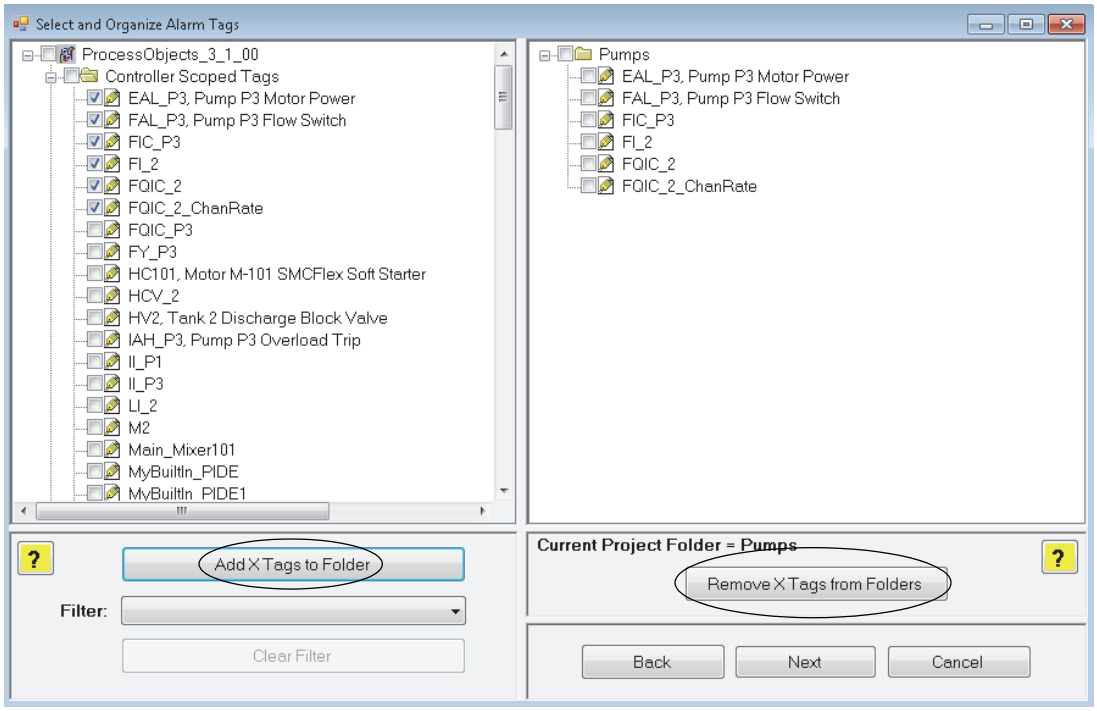
Follow these steps.

1. Right-click the alarm server name, and from the pull-down menu choose Build FactoryTalk A&E Alarms XML Import File>Select/Organize tags and Displays, Review Options.



A message box precedes each dialog box. Click OK to perform the task or click Skip to continue to the next dialog box.

If OK is clicked, the Select and Organize Alarm Tags dialog box appears with two panes. The left pane lists the controllers assigned to the HMI alarm server. The right pane lets you create folders to organize selected tags.



2. In the left pane, check each tag checkbox that you want to use.
3. Right-click in the right pane, and choose Add Folder.

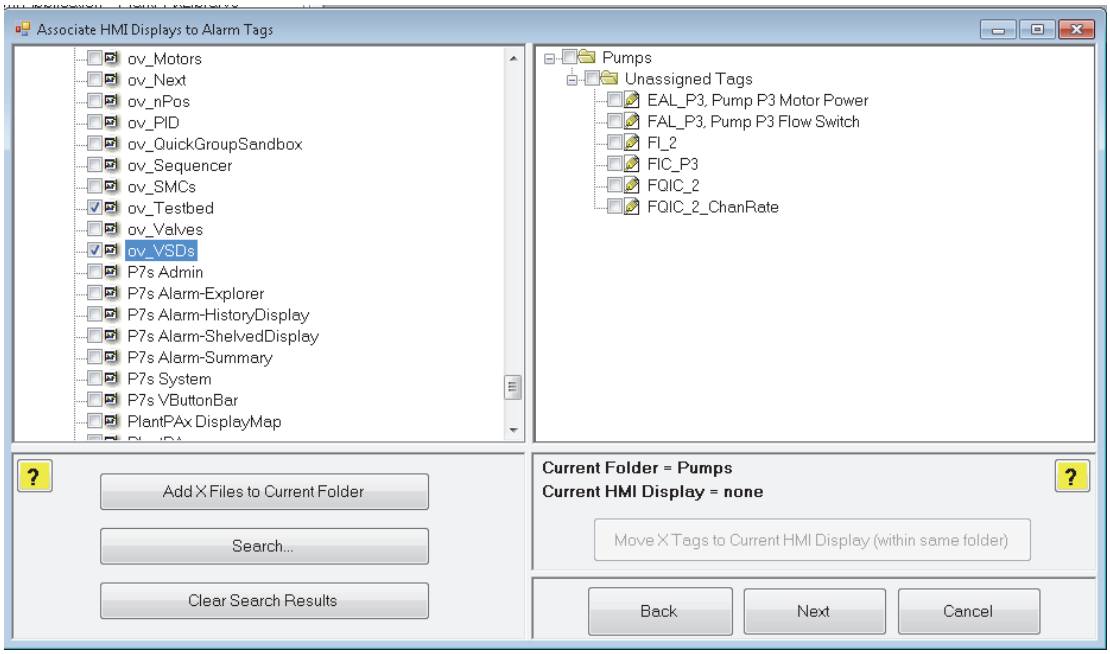
   The Add Project Folder dialog box appears.
4. Type a folder name (description is optional) and click OK.
5. With the folder selected in the right pane, click the Add X Tags to Folder button in the left pane to carry over the selected tags to the folder.
6. To remove tags from a folder, check each tag checkbox and click the Remove X Tags from Folders button in the right pane.

   TIP To filter tag conditions, see Create Alarm Import Files from the Help menu as explained on page 127.
7. Click Next.

B. Associate Alarm Tags to HMI Displays

This option lets you automatically add a user-defined macro, mcrAE_Display DisplayName', to the AE alarm tag's View command string. 'Display Name' is the name of the HMI display associated to the AE tag. You can configure the macro to open the HMI display directly or to access a faceplate that has a button to open a display.

The Associate HMI Displays to Alarm Tags dialog box has two panes. The functionally is the same as for the Select and Organize Alarm Tags dialog box explained on page 139.

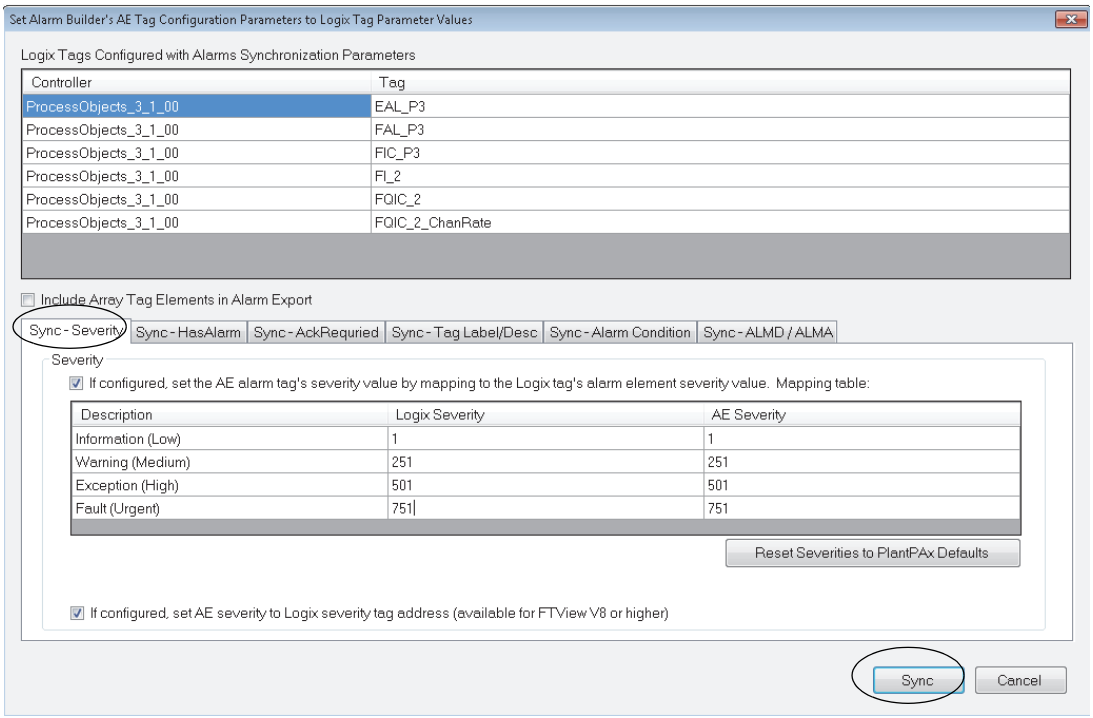


TIP For details, see Create Alarm Import Files from the Help menu as explained on page 127.

C. Synchronize AE and Logix Tags

AE configuration tags are initialized to the default values in the data type alarm definitions. Some of the configuration parameters (severity, acknowledge required) can be configured ('synched') with the Logix tag values.

The Set Alarm Builder's AE Tag Configuration Parameters to Logix Tag Parameter Values dialog box appears.



The top half of the dialog box lists the Logix tags that are configured with AE alarm definitions and synchronization data type elements. The bottom half of the dialog box lets you choose synchronize tab options.

8. Click one of the tabs in the bottom half of the dialog box based on the parameters that you want to synchronize. Sync-Severity is the tab selected in our example.

   The mapping values are set based on the Logix and AE tags' severity.

   See Table 38 on page 131 for mapping values.

IMPORTANT The AE tag is added to the XML import file if the Logix tag's 'HasAlarm' value is set to '1'. For example, if a P_AIn tag is configured with a Cfg_HasHiHiAlm equal to '1', and Cfg_HasHiAlm equal to '0', the Alarms Builder tool adds a HiHi alarm AE tag to the XML import file and does not add a Hi alarm AE tag.

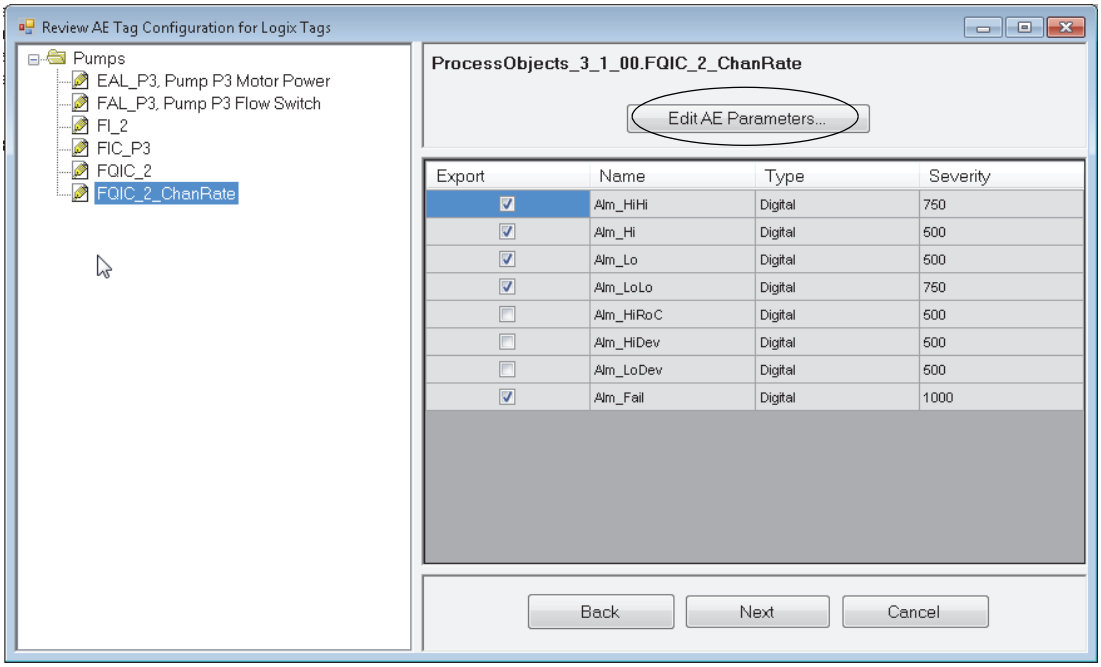
The Sync-AckRequired tab is not a configurable option. All of these parameters for AE tags are configured as **true** per the Rockwell Automation Process Library alarm configuration guidelines.

String local tags for storing Add-On Instructions' labels and descriptions can be used in the AE alarm messages created by the Alarms Builder tool.

9. Click Sync.
10. Click OK to complete synchronization.

D. Review Individual AE Alarm Tag Configuration

The Review AE Tag Configuration for Logix Tags dialog box lets you streamline and review the AE alarm tags that are generated for each Logix tag.



1. In the left pane, click a Logix tag node.

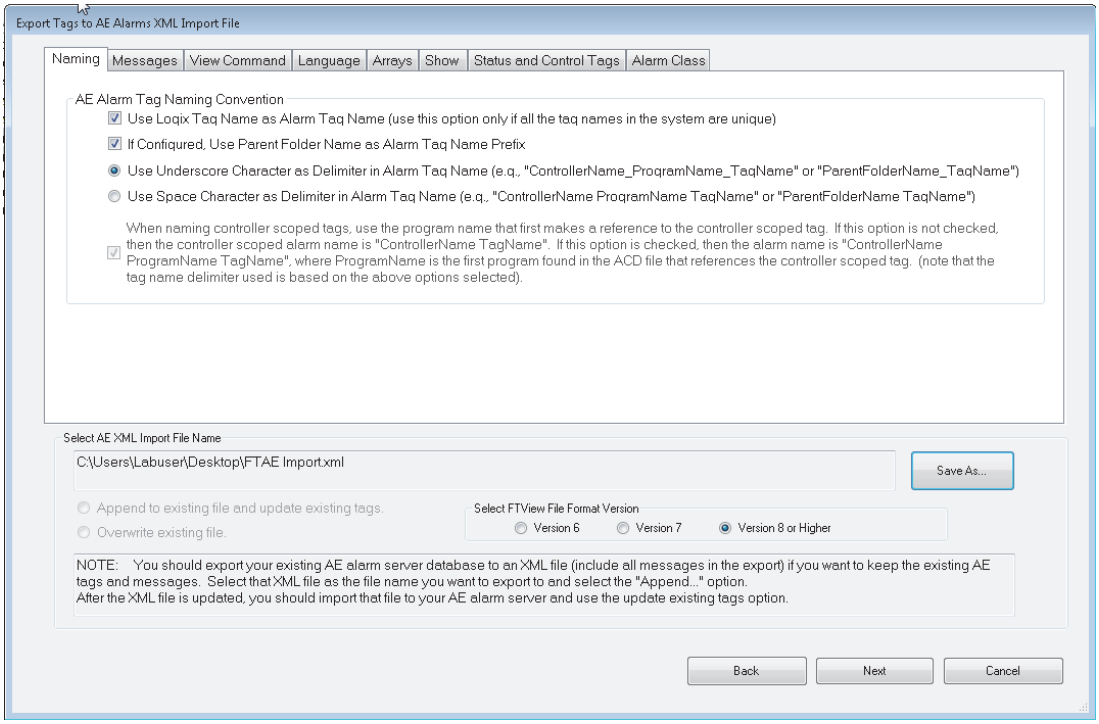
   The associated AE alarm tag appears in the right pane.

   The Export column checkboxes are checked for the alarm tags if the tag values were synchronized to the Logix values (see page 138).

2. Click the Edit AE Parameters button to change the severity values.

E. Review/Edit XML File Build Options

The Export Tags to AE Alarms XML Import File dialog box lets you review options for building the AE alarms XML import file.



1. In the top half of the screen, click each tab to view and select desired options.
2. In the lower half of the screen, click Save As to browse to the path of the import file.
3. Click Next.

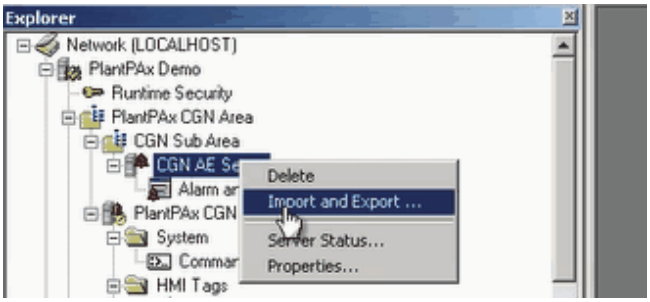
   An Export window appears.
4. Click OK to build the XML import file.

5. Save XML Import File

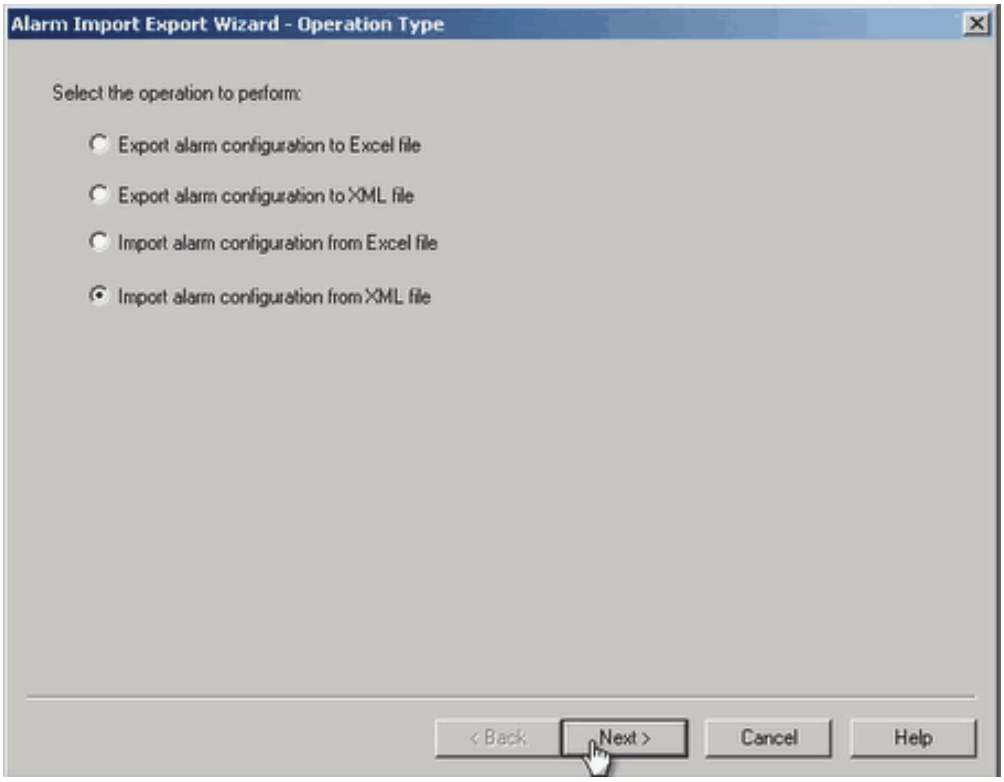
This section describes how to use FactoryTalk View Studio software to implement the XML file to import alarms into the PlantPAx system.

Follow these steps after opening the FactoryTalk View Studio software.

1. From the FactoryTalk View Studio Explorer tree configuration, right-click the AE alarm server and choose Import and Export.



The Alarm Import Export Wizard - Operation Type dialog box appears.



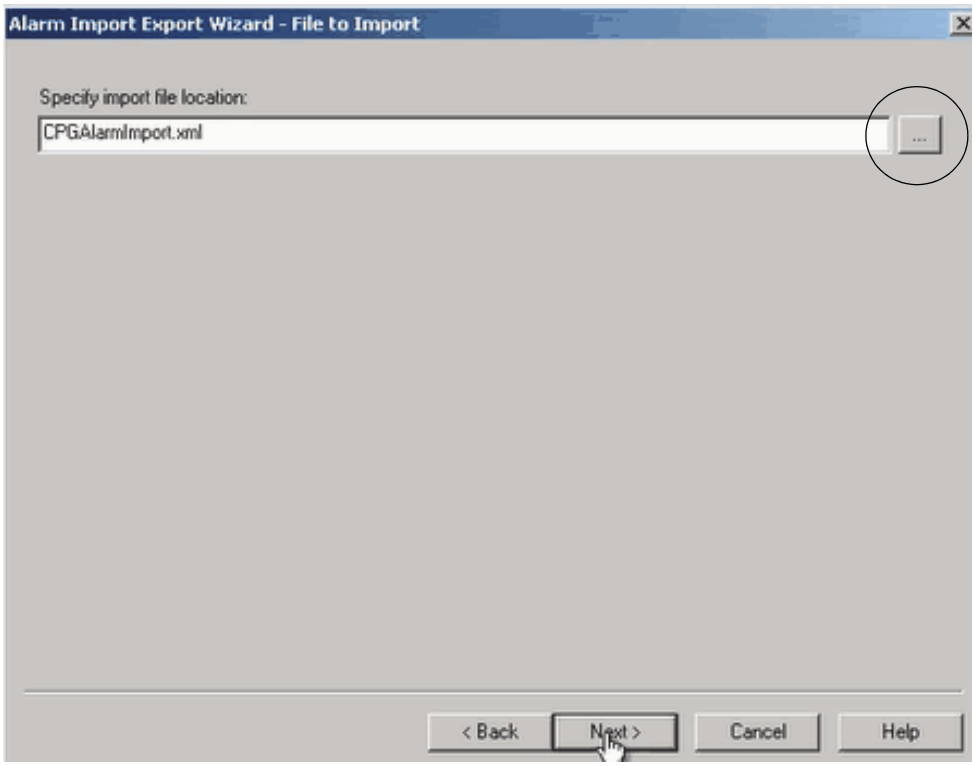
2. Click Import alarm configuration from XML file and click Next.

   You are importing the XML file that you created with QuickBuild on page 136.

The Alarm Import Export Wizard - File to Import dialog box appears.

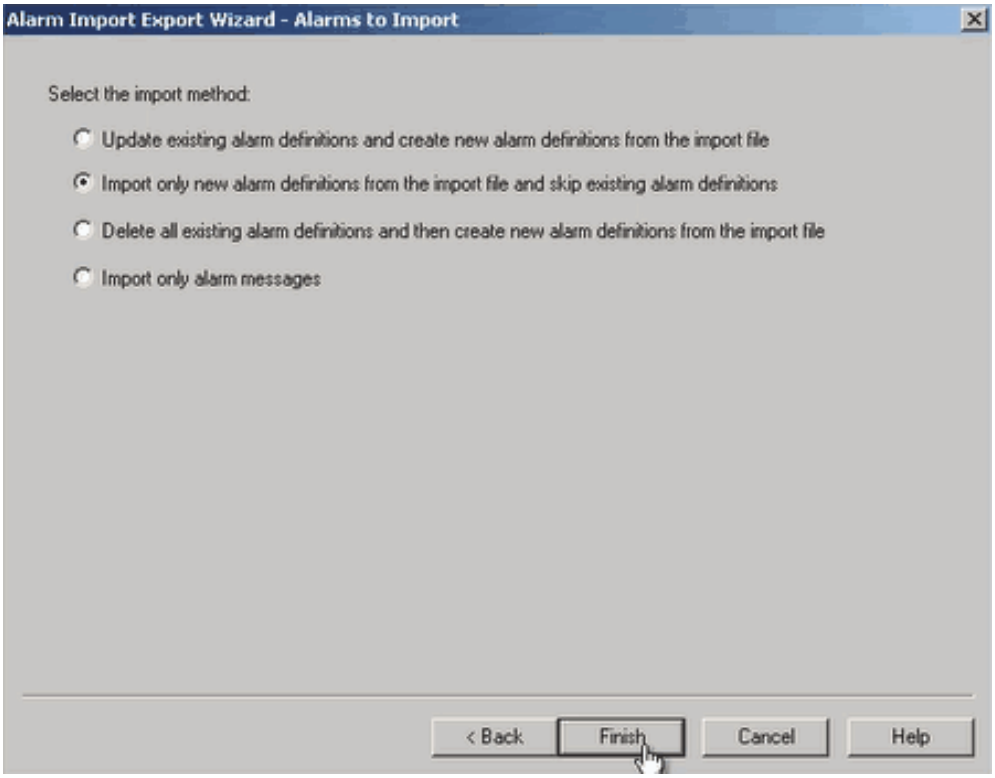
3. Click Browse (...) to find where you saved the file on page 137.
4. Select the import file and click Open.

   The name of your XML file displays in the import file location text box.



5. Click Next.

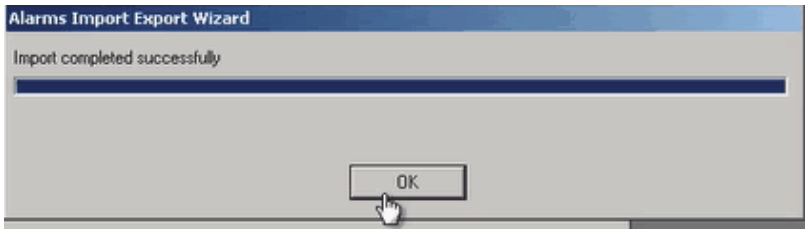
The Alarm Import Export Wizard - Alarms to Import dialog box appears with several choices.



6. Click Import only new alarm definitions from the import file and skip existing alarm definitions and click Finish.

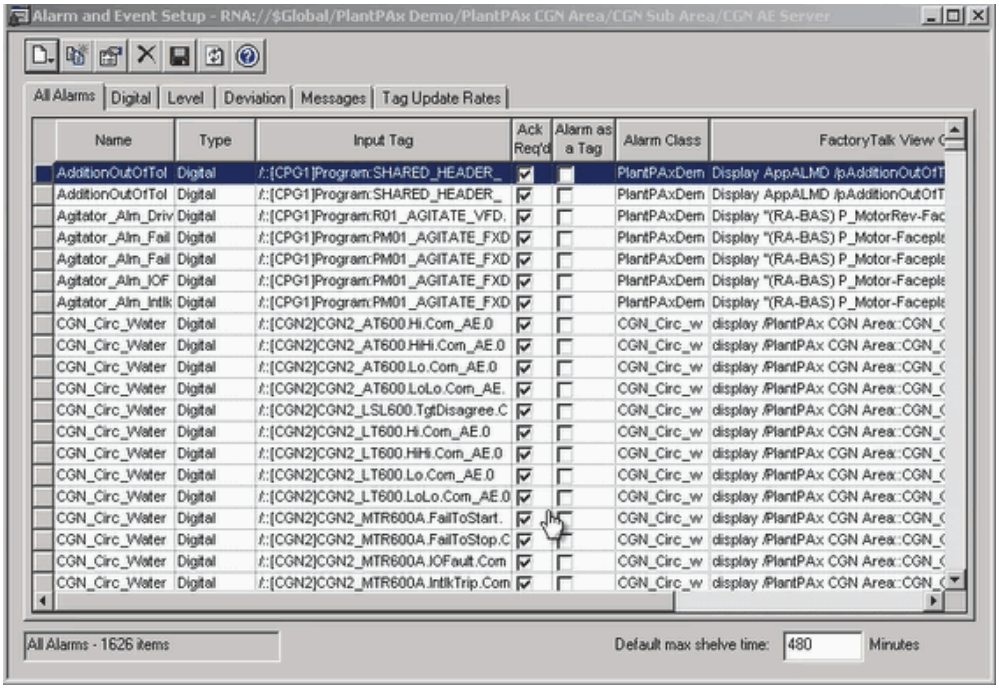
   The Alarm Builder loads the file. It can take several minutes for the import to complete.

   A message box appears telling you the import was successful.



7. Click OK.
8. In the Explorer tree configuration, double-click Alarm and Event Setup.

The Alarm and Event Setup tag database window appears to let you view the alarm definitions.

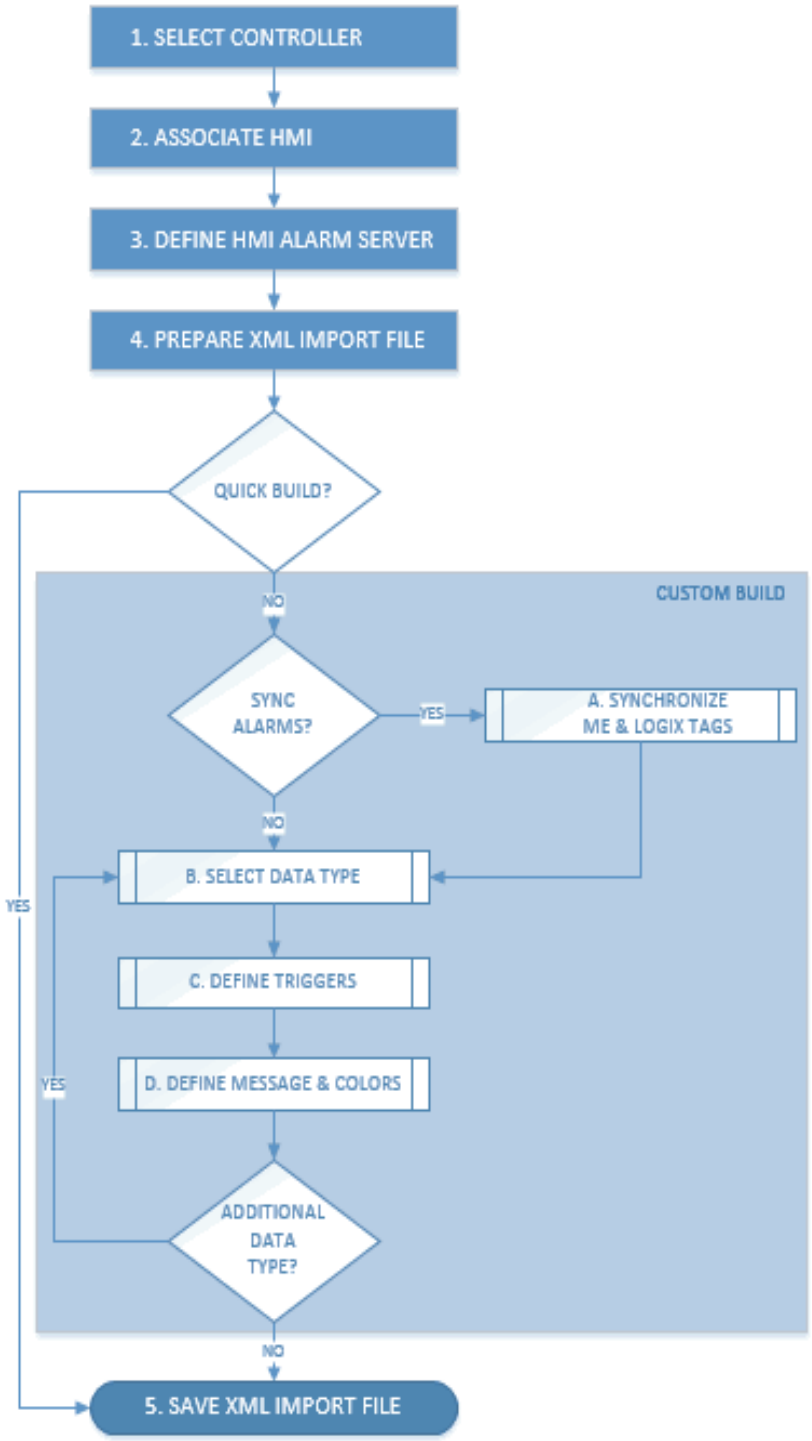


9. Click the 'X' in the upper, right corner to close the window.

Build ME Alarms

The diagram outlines the procedures for creating FactoryTalk View ME software alarms. The procedures in this section are in the same order as the headings in the diagram.

Figure 8 - Alarm Builder ME Workflow



1. Select Controller

Complete the steps, starting on page 129.

2. Associate HMI

Complete the steps, starting on page 133. However, select Machine Edition (ME) for the product type on the Select HMI Project dialog box.

IMPORTANT	If the controllers in the alarm server are associated with FactoryTalk View SE software projects, the ME XML import file building tools can be used as long as the data server device shortcut name for the controller is the same for SE and ME. If the device shortcut names are different, you must create separate Alarm Builder projects.

3. Define HMI Alarm Server

Complete the steps, starting on page 135.

4. Prepare XML Import File

You have two options to build the XML import file:

- Quick Build
- Custom Build

We recommend that you use the Quick Build default options to create the XML import files. For optional tag configurations (as shown in the flow chart), see Custom Build on page 151.

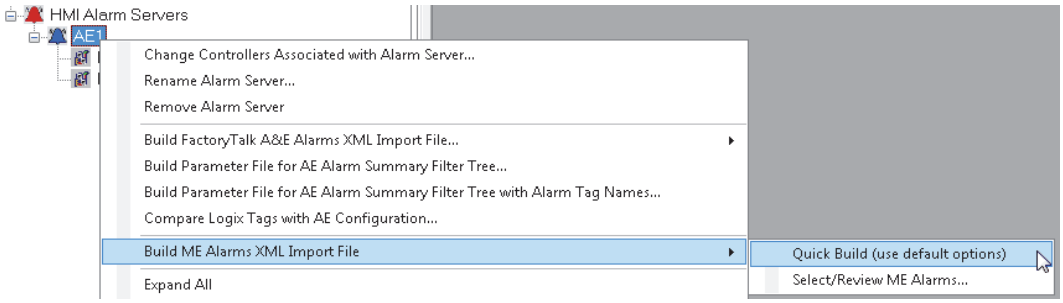
Quick Build

The ME alarms import file uses the following default build settings:

- Discrete alarms are only added
- All Logix tags in the alarm server controllers with alarm definitions are used
- ME tag is added to the XML file only if the Logix tag's HasAlarm value is '1'
- Logix tag's Cfg_Desc and Cfg_Label local string tag values are added to the ME alarm trigger message
- All other ME parameters are configured based on the Logix data type alarm definition

Follow these steps to create the XML import file by using the default settings.

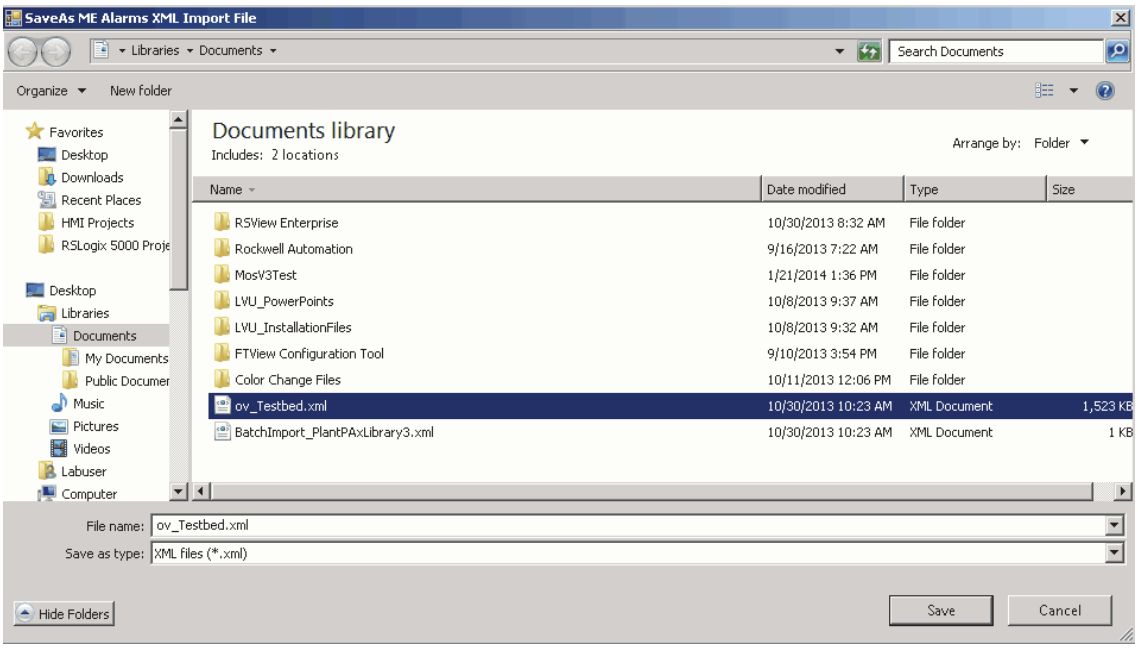
1. Right-click the alarm server name, and from the pull-down menu choose Build ME Alarms XML Import File>Quick Build (use default options).



A dialog box appears that explains the QuickBuild default build options.

2. Click Yes after reading about the settings when the XML file is imported to the ME alarm database.

The SaveAs ME Alarms Import XML File dialog box appears.



3. Choose where to save the file and type a file name.

IMPORTANT	Choose a file location that is easily accessible, such as your desktop, because you need this XML file to complete the alarms process.

4. Click Save.

A message box appears that asks if you want to overwrite or append to the file.

IMPORTANT	FactoryTalk View ME software deletes all current alarms in the ME alarm server tag database when importing an XML file.To keep existing ME alarms, we suggest that you export the alarms to an XML file and then select that file in the SaveAs dialog box. Click No to have the Alarms Builder tool to append to that file.

It can take a few minutes to generate the ME XML file.

5. Click OK.

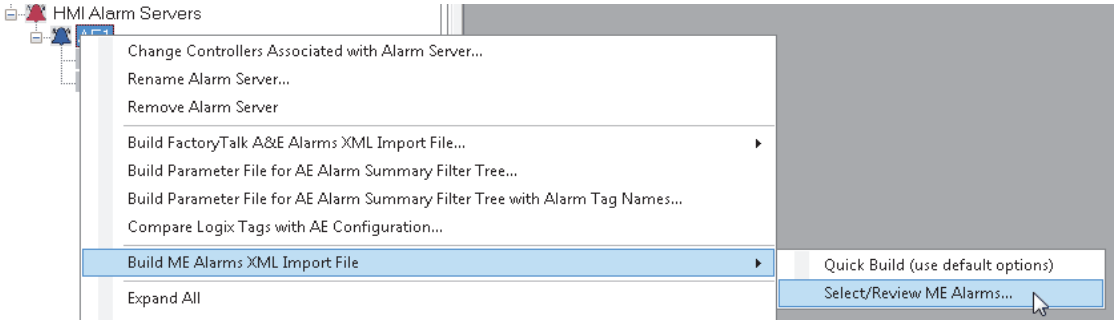
Custom Build

The Alarm Builder's ME tag configuration values are initialized to the default values defined in the Logix data types alarm definitions. Some of the ME configuration parameters (severity, acknowledge required) can be configured ('synched') with the Logix tag values.

A. Synchronize ME & Logix Tags

Follow these steps.

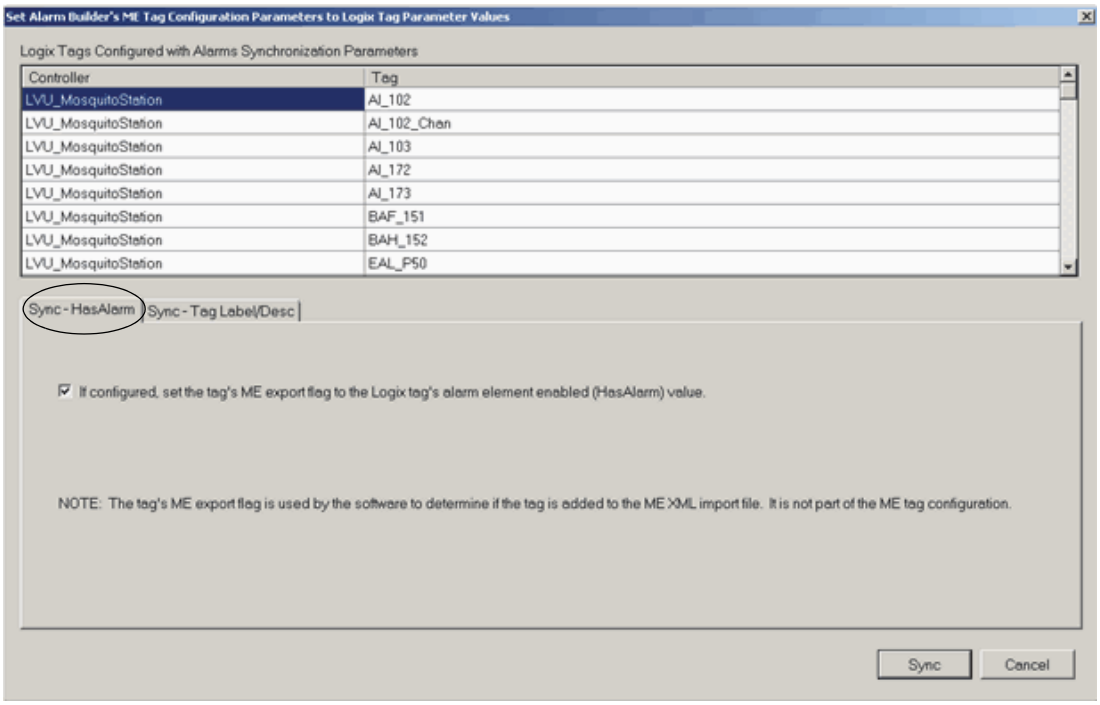
1. Right-click the alarm server name, and from the pull-down menu choose Build ME Alarms XML Import File>Select/Review ME Alarms.



A message box appears.

2. Click Yes.

The Set Alarm Builder's ME Tag Configuration Parameters to Logix Tag Parameter Values dialog box appears.



The top half of the dialog box lists the Logix tags that are configured with ME alarm definitions and synchronization data type elements. The bottom half of the dialog box lets you specify the synchronize options to use.

3. Click one of the tabs in the bottom half of the dialog box based on the parameters that you want to synchronize. Sync-HasAlarm is the tab selected in our example.

IMPORTANT The ME tag is added to the ME XML import file if the Logix tag's 'HasAlarm' value is set to '1'. For example, if a P_AIn tag is configured with a Cfg_HasHiHiAlm equal to '1', and Cfg_HasHiAlm equal to '0', the Alarms Builder tool adds a HiHi alarm ME tag to the XML import file and does not add a Hi alarm ME tag.

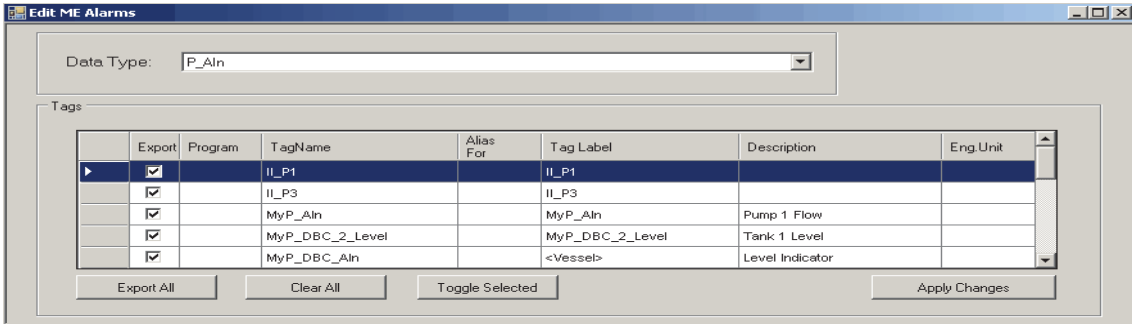
4. Click Sync.
5. Click OK to complete synchronization.

B. Select Data Type

You can select individual tags for the data types found for the alarm server controllers that have alarm definitions.

The Edit ME Alarms dialog box consists of three sections for tag configuration:

- Upper -- data type and tag selection (shown below)
- Middle -- trigger variables
- Lower -- trigger messages.



1. Click the Data Type drop-down to access the associated tags.
2. In the Export column, check each checkbox for the selected tags and use the buttons to export, toggle, and clear all.

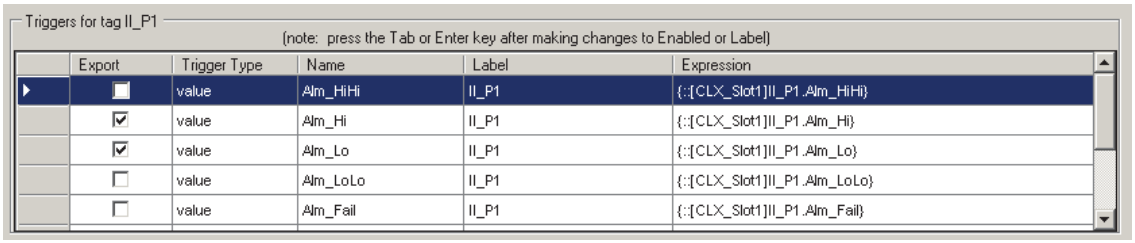
IMPORTANT	Make sure to click the Apply Changes button to save changes for the tags in the grid box.

3. Click the Tag Label, Description, or Eng. Unit text boxes to type new cell values.

   The Alarms Builder tool parses the Description in the controller's ACD file to set the initial values for the tag label, description, and engineering unit.

C. Define Triggers

The middle section of the Edit ME Alarms dialog box shows the trigger variables for the selected tags.

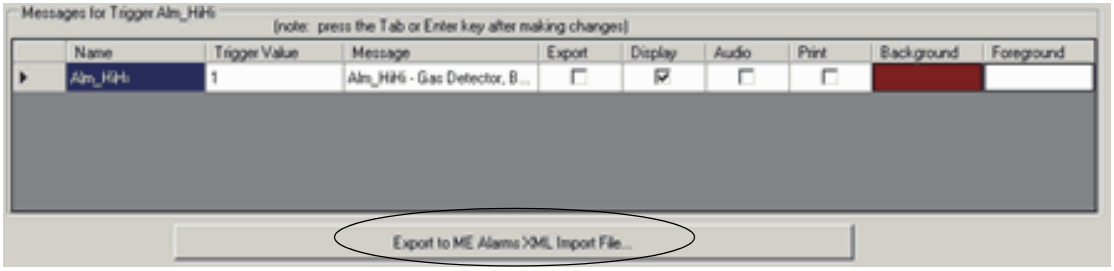


The trigger variables and trigger messages (lower portion of the dialog box) use the default alarm configuration values if the synchronization option is not used. Checkmarks default in the Export checkboxes for Logix tag elements with HasAlarm set to 1.

To edit the triggers, check or clear the checkboxes for the selected trigger values.

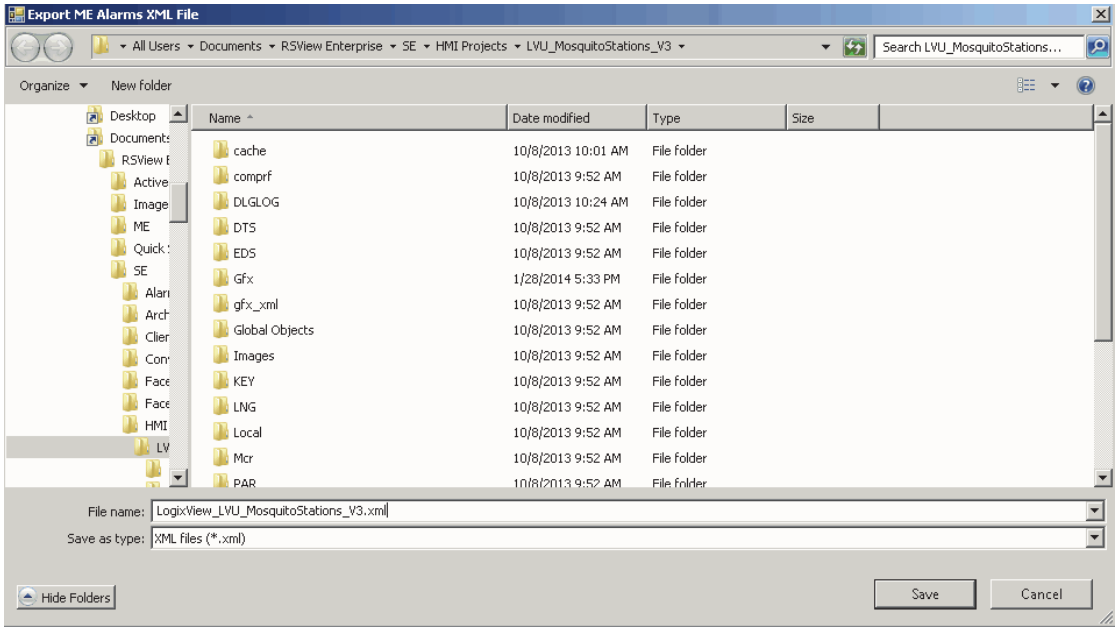
C. Define Message & Colors

The message name and color for the alarm trigger are maintained in the lower portion of the dialog box.



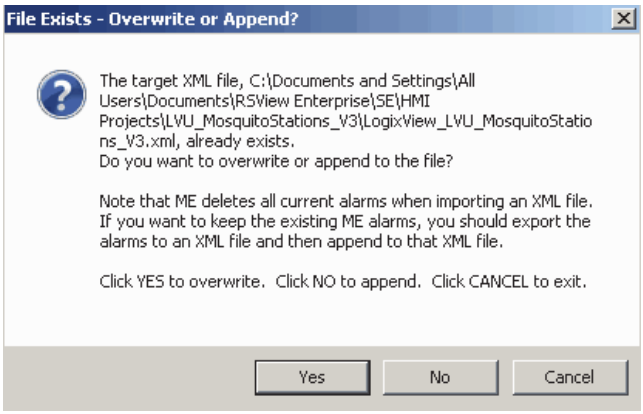
1. Click Name and Message cells to edit the text boxes.
2. Click Background and Foreground cells to change the colors.
3. Click Export to ME Alarms XML Import File to select a trigger variable.

The Export ME Alarms XML File dialog box appears. The file's directory defaults to the ME HMI project directory.



4. Type a file name and click Save.

   If the file name is an existing file, a warning message box appears.



IMPORTANT FactoryTalk ME software deletes all current alarms in the ME alarm server tag database when importing an XML file. To keep existing \ ME alarms, you must export the alarms to an XML file and then append to that XML file.

If append is clicked in the message box, the Alarms Builder tool overwrites any existing tag information in the XML file with the Alarms Builder tool-configured parameter settings.

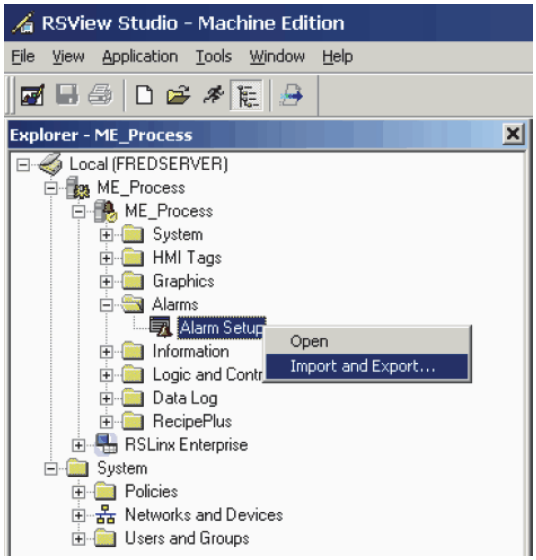
5. Click Yes to overwrite, No to append, or Cancel to abort the alarm export.
6. Click OK to export the tags to the ME alarms XML import file.

5. Save XML Import File

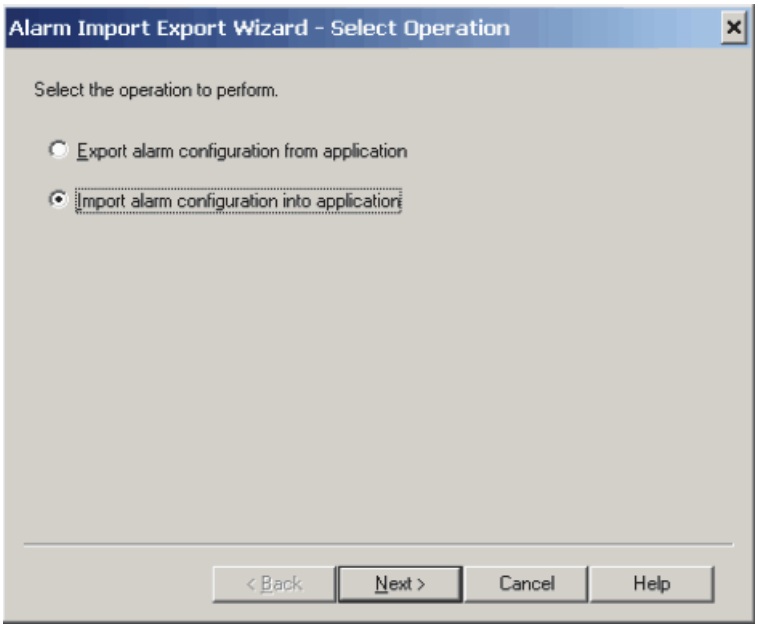
This section describes how to use FactoryTalk View Studio software to implement the XML file to import alarms into the PlantPAx system.

Follow these steps after opening the FactoryTalk View Studio software.

1. From the FactoryTalk View Studio Explorer tree configuration, right-click the AE alarm server and choose Import and Export.

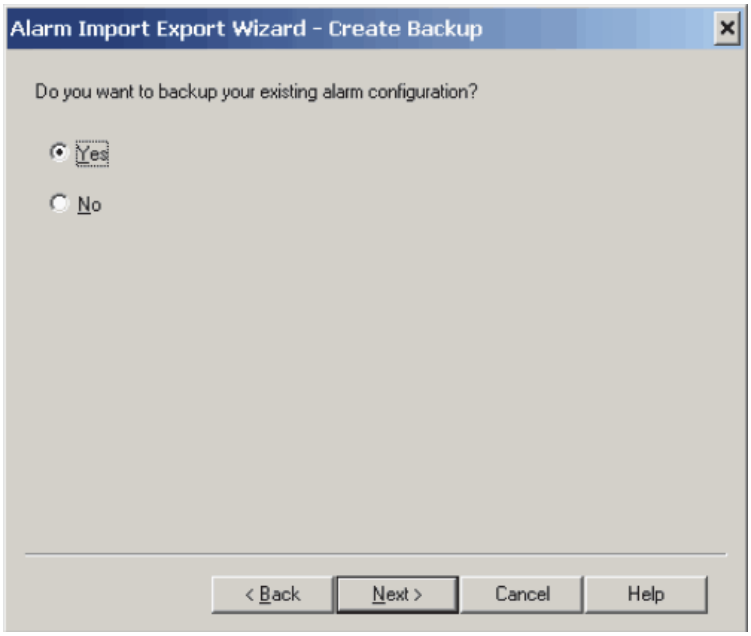


The Alarm Import Export Wizard - Select Operation dialog box appears.

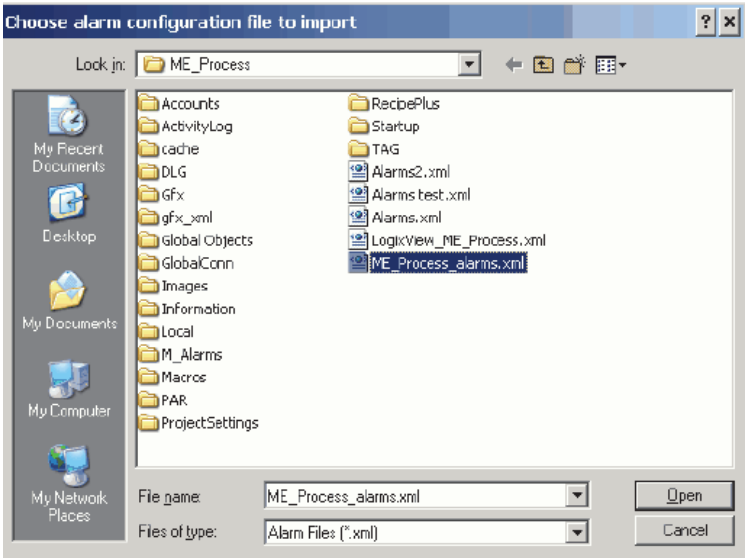


2. Click Import alarm configuration into application and click Next.

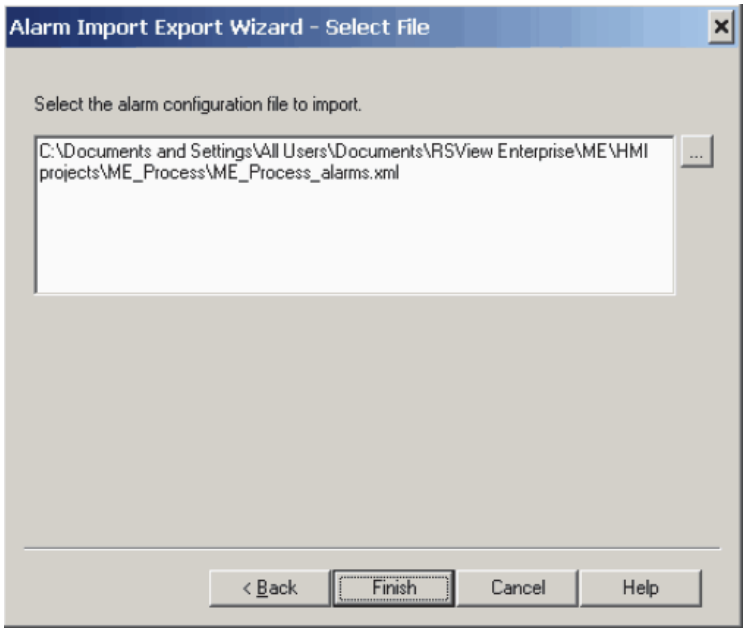
We recommend that you make a backup until you are familiar with the import process.



3. Type a name for the backup file and click Next.
4. Click Browse (...) on the alarm wizard dialog box to find where you saved the file.
5. Select the alarm configuration file and click Open.



The Alarm Import Export Wizard - Select File dialog box appears.



6. Click Finish.

Color Change Utility

This customizing tool lets you create a color palette to change the colors for FactoryTalk View software display elements (global objects) and faceplates.

The Color Change Utility uses three types of files:

- **FactoryTalk View Graphics .xml file:** This file is exported from the FactoryTalk View graphic (display or global object) in the View Studio software program. Once changes are made, it is imported into the View Studio software program to change the colors in the display or global object.
- **Color Association File:** This .xml file matches a color instance in the FactoryTalk View Graphics .xml file to the color palette entry. There is one Color Association File (CAXML) for each FactoryTalk View Graphics .xml file. It is created and maintained by the utility.
- **Color Palette:** This .xml file defines the colors for an application. It is created and maintained by the utility. There is one color palette file for all of the FactoryTalk View Graphics .xml files that are being customized. If you want to change the color, it is done in the color palette.

TIP We suggest that you make a copy of the color palette .xml file if you plan to use the color tool.

Install Tool File

The Color Change Utility can be downloaded with the Library of Process Objects from the Product Compatibility and Download Center at http://www.rockwellautomation.com/rockwellautomation/support/downloads.page. Choose Graphics>Color Change Utility>and double-click FTViewCustomizationSetup,msi.

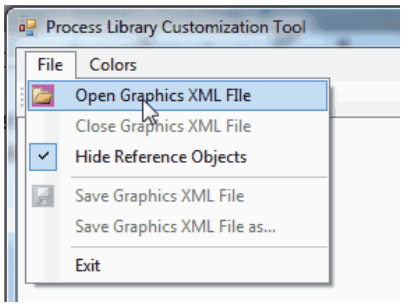
This file installs the program and adds a shortcut to your computer's Start menu under 'PlantPAx.'

Use the Utility with Library Objects

The download includes .xml exports for all of the global objects and display files in the library (for FactoryTalk View SE software). Make sure that you also download the CAXML and Process Library Standard Colors .xml files.

Follow these steps to change colors in the process library.

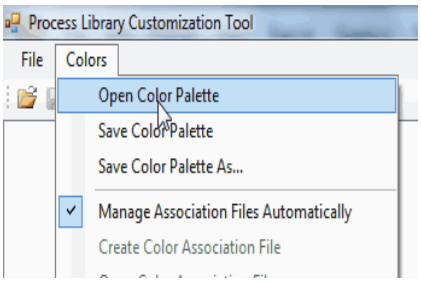
1. From the Process Library Customization Tool File menu, click Open Graphic XML File.



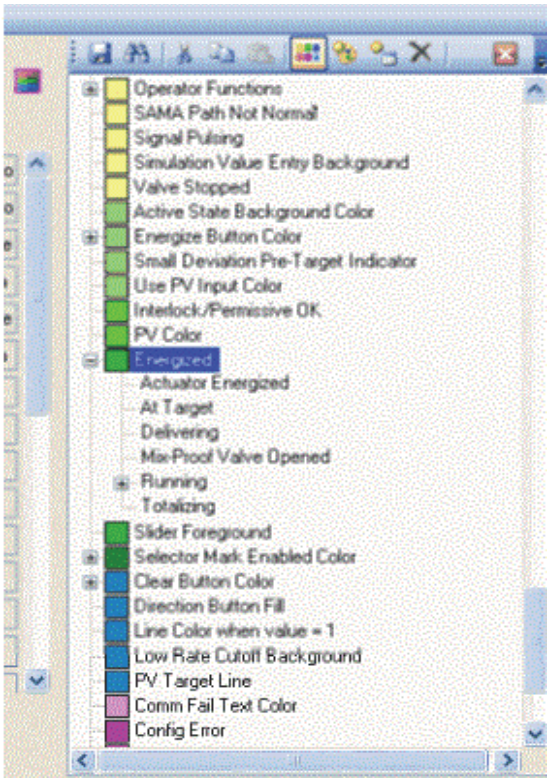
   The Open Graphics XML Files dialog box appears.

   Multiple global object and display files can be opened at the same time from the file open dialog box.

2. Click the Colors tab and choose Open Color Palette.

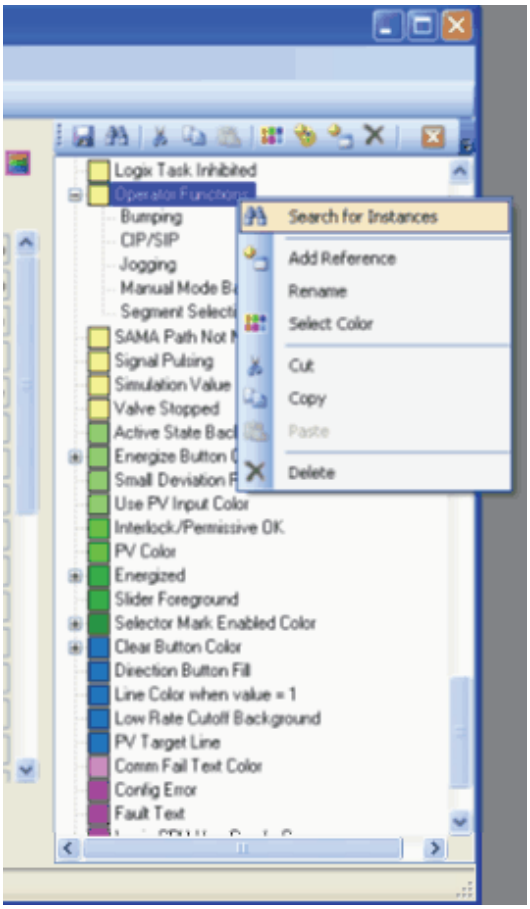


3. Select the colors that you want to change in the palette.

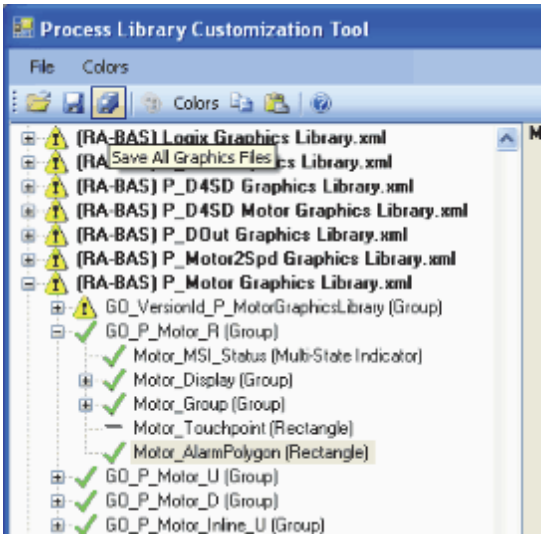


4. Click the Choose Color icon to select a new color.
5. Repeat step 4 for each color that needs to be changed.

6. To see where a color is used, right-click a color and choose Search for Instances.



7. Click Save All  to save all of the graphic files (along with their association files) and the color palette file.



8. Import the files into the FactoryTalk View software program.

There are bulk import files for the displays (BatchImport_Displays_PlantPAx)Library.xml) and global objects (BatchImport_Global_OlantPAx)Library.xml).

Modifying the Color Palette

The color palette appears in a tree format that shows a parent-child relationship between colors. 'Base Colors' are shown with a color box next to them. 'Reference Colors' reference either a Base Color or another Reference Color.

By changing a Base Color, all of the Reference Colors under it change. For example, you can create a generic Base Color, called 'Energized', and then reference it with the Reference Color, called 'Running'.

Color palette entries are not to be deleted unless they are known to be unused. To see if a color palette entry is being used, right-click the color and choose 'Find Color Instances'.

Any color palette entry (Reference or Base Color) can be moved to reference another color. This is done by simply dragging the color to be moved and dropping it on the new color to reference. When a color that has references is moved, all of its references move as well.

A Reference Color can also be made a Base Color by right-clicking the Reference Color and selecting 'Make Base Color' from the context menu.

Color palette entries are stored with an integer code and that integer code is used in the association file. Renaming a color palette entry does not break any existing associations. Multiple color palette entries can have the same name, but this is not recommended.

Follow these color palette considerations:

- Once a color palette entry is deleted and the palette is saved, the only way to restore associations is to manually recreate them.
- Object names in FactoryTalk View software usually have a number on the end. Names are considered to be similar if they are the same after the ending number is removed.
- Button icons are not associated with the color palette for the following reasons:
  - Future versions of the library can change these icons to images
  - Button icons are global objects and do not have multiple definitions, the colors need to be changed only in one file
  - Additional color palette entries for individual button icons can increase palette maintenance

Use the Utility with other FactoryTalk View Software Files

To apply the color palette to FactoryTalk View software files that are not part of the Rockwell Automation Library, the graphic elements in the file must be associated to the color palette. You must create associations and save them in a color association file. When opening an .xml graphics file, if the file already has an association file (CAXML) this automatically opened as well. If an association file does not exist, it is created.

Follow these steps to create associations.

1. From the Process Library Customization Tool File menu, click Open Graphic XML File.

   The Open Graphics XML Files dialog box appears.

2. Select an object from the tree on the left, and its colors appear in the center of the screen.

3. To associate a color from the palette, select the palette color and drag it to the text box next to the color display box.

   Once all colors for an object are associated with the color palette, a check appears next to the object in the tree.

   Colors that are actually used for the object only are displayed. For example, if an object is configured as 'Transparent', its background color does not show up in the utility. Also, instances of global objects from display files do not appear in the object tree. The tree can be configured to show instances of global objects, but the objects do not have any color instances because their colors are controlled by their parent global objects.

4. Copy and paste functions have been included to allow quick creation of color associations. To use these functions, right-click on the graphic object in the tree on the left and a menu appears.
   - **Copy Color Associations:** Use this function to copy the color associations for the object. If the object is a group, the color configurations for all group members is copied.
   - **Paste Color Associations (this Object only):** Use this function to paste the previously copied color associations to the selected object. This option is not available if the selected object is a group that has members with color associations.
   - **Paste Color Associations (to all group members):** Use this function to paste the previously copied color associations to the new object and all of its members. This option is available only if the source and destination objects are groups with members that have similar names and object types.
   - **Copy and Paste Color Associations to Similar Objects with Names like 'Xxxx#'**: This option copies the selected object and searches objects with a similar name and object type. Color associations are copied to all objects with similar names and types in any of the currently open graphics files. If the objects are groups, then the group members must have similar names and object types. Be careful when you use this feature to prevent unwanted changes.

Additional Add-On Instructions

Long Integer and Time Instructions

The Rockwell Automation Library of Process Objects provides additional sets of Add-On Instructions because the Logix firmware does not provide operations on Long Integers (LINT, 64-bit signed integers) used as timestamps. The instructions in Table 39 provide 64-bit integer match functionality for the library objects.

The long integer instructions are **calculation functions only**, and no HMI components are provided.

Table 39 - Long Integer Instructions

Name	Short Description	Long Description	File Name
L_ABS	Absolute Value (64-bit)	This instruction returns the absolute value (positive magnitude) of an input 64-bit integer (LINT) value.	L_ABS_1_0-00_AOI.L5X
L_ADD	Add (64-bit)	Adds two LINT (signed 64-bit) values and returns a LINT (signed 64-bit) sum. Also provides math status bits for Carry, Negative, Overflow and Zero result (equivalent to built-in S:C, S:N, S:V, S:Z).	L_ADD_1_0-00_AOI.L5X
L_AND	Bitwise AND (64-bit)	This instruction returns the bitwise Logical **and** (output bit true if both corresponding input bits true) of two input 64-bit integers (LINTs), into an output 64-bit integer.	L_AND_1_0-00_AOI.L5X
L_DEC	Decrement (64-bit)	This instruction decrements the input 64-bit integer, returning its value minus 1.	L_DEC_1_0-00_AOI.L5X
L_DIV	Divide (64-bit by 32-bit)	This instruction implements an elementary-school shift/subtract (looping) method of dividing a 64-bit integer (LINT) dividend by a 32-bit integer (DINT) divisor. The resulting quotient is a 64-bit integer (LINT), and the remainder (32-bit integer DINT) is also returned.	L_DIV_1_0-00_AOI.L5X
L_EQU	Equal (64-bit)	This instruction compares two LINT (64-bit signed integer) variables. If Inp_A is equal to Inp_B, EnableOut and Out are set to **true** (1). Otherwise EnableOut and Out are cleared to **false** (0).	L_EQU_1_0-01_AOI.L5X
L_FtoH	Float to Half-Precision	This instruction converts a 32-bit single-precision floating point number (REAL) to the best equivalent 16-bit 'half-precision' floating point number (stored in an INT, because Logix does not have a SREAL type). It accounts for positive and negative zero, subnormal (very small) numbers, Infinity (+/- 1.$), Indeterminate (-1.#IND) and Not a Number (+/- 1.#QNaN).	L_FtoH_1_0-00_AOI.L5X
L_GEQ	Greater Than or Equal (64-bit)	This instruction compares two LINT (64-bit signed integer) variables. If Inp_A is greater than or equal to Inp_B, EnableOut and Out are set to **true** (1). Otherwise EnableOut and Out are cleared to **false** (0).	L_GEQ_1_0-01_AOI.L5X
L_GRT	Greater Than (64-bit)	This instruction compares two LINT (64-bit signed integer) variables. If Inp_A is greater than Inp_B, EnableOut and Out are set to **true** (1). Otherwise EnableOut and Out are cleared to **false** (0).	L_GRT_1_0-01_AOI.L5X

Table 39 - Long Integer Instructions

Name	Short Description	Long Description	File Name
L_HtoF	Half-Precision to Float	This instruction converts a 16-bit ('half-precision') floating point number (contained in an INT, as Logix doesn't have an SREAL type) to the equivalent 32-bit single-precision floating point number (REAL). It accounts for positive and negative zero, subnormal (very small) numbers, Infinity (+/- 1.$), Indeterminate (-1.#IND) and Not a Number (+/- 1.#QNaN).	L_HtoF_1_0-00_AOI.L5X
L_INC	Increment (64-bit)	This instruction increments the input 64-bit integer, returning its value plus 1.	L_INC_1_0-00_AOI.L5X
L_LEQ	Less Than or Equal (64-bit)	This instruction compares two LINT (64-bit signed integer) variables. If Inp_A is less than or equal to Inp_B, EnableOut and Out are set to **true** (1). Otherwise EnableOut and Out are cleared to **false** (0).	L_LEQ_1_0-01_AOI.L5X
L_LES	Less Than (64-bit)	This instruction compares two LINT (64-bit signed integer) variables. If Inp_A is less than Inp_B, EnableOut and Out are set to **true** (1). Otherwise EnableOut and Out are cleared to **false** (0).	L_LES_1_0-01_AOI.L5X
L_LIM	Limit Test (Circular) (64-bit)	This instruction compares a 64-bit Input with a 64-bit High Limit and a 64-bit Low Limit. There are two cases: a 'normal case' (Low Limit <= High Limit) and a 'circular case' (Low Limit > High Limit) In the normal case, EnableOut and Out are set if: Low Limit <= Input <= High Limit In the circular case, EnableOut and Out are set if: Input >= Low Limit OR Input <= HighLimit (remember, High Limit < Low Limit) This instruction can be used in Ladder Diagram, Structured Text or Function Block Routines just like the LIM instruction is used for 32-bit integers and floating point numbers. However, because it has InOut Parameters (references to tags of LINT type), it is **not** left-justified on ladder rungs. On a **False** rung in LD, or in FBD with EnableIn cleared to 0, Out is cleared to 0.	L_LIM_1_0-01_AOI.L5X
L_MEQ	Masked Equal (64-bit)	Performs a 64-bit bitwise comparison of a Source Value against a Compare Value, and returns **true** if they are the same in all bit positions that have a '1' in the Mask Value. Therefore, output is **true** if (Source AND Mask) = (Compare **and** Mask). On a **False** rung in LD, or in FBD with EnableIn cleared to 0, Out is cleared to 0.	L_MEQ_1_0-01_AOI.L5X
L_MUL	Multiply (64-bit X 32-bit)	This instruction implements an elementary-school multiply-and-add-partial-products (place notation) method of multiplying a 64-bit integer (LINT) by a 32-bit integer (DINT). The resulting product is a 64-bit integer (LINT).	L_MUL_1_0-01_AOI.L5X
L_MVM	Move with Mask (64-bit)	Performs a 64-bit bitwise Move with Mask of a Source Value to an Output. If a bit in the Mask is **true**, the corresponding Source bit is copied to the Output. If a bit in the Mask is **false**, the corresponding Output bit is left unchanged. In other words, Output = (Output AND NOT Mask) OR (Source **and** Mask).	L_MVM_1_0-00_AOI.L5X
L_NEG	Negate (64-bit)	This instruction returns the negative (2's complement) of an input 64-bit integer (LINT) value.	L_NEG_1_0-00_AOI.L5X
L_NEQ	Not Equal (64-bit)	This instruction compares two LINT (64-bit signed integer) variables. If Inp_A is not equal to Inp_B, EnableOut and Out are set to **true** (1). Otherwise EnableOut and Out are cleared to **false** (0).	L_NEQ_1_0-01_AOI.L5X
L_NOT	Bitwise NOT (64-bit)	This instruction returns the bitwise Logical NOT (inverse or 1's complement) of an input 64-bit integer (LINT) value. (It flips all the bits.)	L_NOT_1_0-00_AOI.L5X
L_OR	Bitwise OR (64-bit)	This instruction returns the bitwise Logical OR (output bit true if either of the corresponding input bits are true) of two input 64-bit integers (LINTs), into an output 64-bit integer.	L_OR_1_0-00_AOI.L5X

Table 39 - Long Integer Instructions

Name	Short Description	Long Description	File Name
L_OTE	Output Energize (64-bit)	This instruction energizes the given bit of the referenced LINT (64-bit integer) tag, that is, it sets the given bit (true, 1) if the EnableIn condition is **true**, or clears the given bit (false, 0) if the EnableIn condition is **false**. If the given bit number is outside the range 0…63, the controller **major faults** on an invalid array index (bad bit number). There is no validity checking of the given bit number.	L_OTE_1_0-00_AOI.L5X
L_OTL	Output Latch (64-bit)	This instruction latches the given bit of the referenced LINT (64-bit integer) tag, that is, it sets the given bit (true, 1) if the EnableIn condition is **true**, or leaves the given bit (and the referenced LINT) unmodified if the EnableIn condition is **false**. If the given bit number is outside the range 0…63, the controller **major faults** on an invalid array index (bad bit number). There is no validity checking of the given bit number.	L_OTL_1_0-00_AOI.L5X
L_OTU	Output Unlatch (64-bit)	This instruction unlatches the given Bit of the referenced LINT (64-bit integer) tag, that is, it clears the given bit (false, 0) if the EnableIn condition is **true**, or leaves the given bit (and the referenced LINT) unmodified if the EnableIn condition is **false**. If the given Bit number is outside the range 0…63, the controller **major faults** on an invalid array index (bad bit number). There is no validity checking of the given bit number.	L_OTU_1_0-00_AOI.L5X
L_SEL	Select (64-bit)	This instruction returns Input A if the input selector bit is **false**, Input B if the selector bit is **true**. **IMPORTANT:** When EnableIn is **false**, the input selector bit sense is reversed. With the selector bit default value of 1, this lets the rung state in an LD instance to control the selector in a straightforward manner. If the rung is **true**, select Inp_B; if the rung is **false**, select Inp_A.	L_SEL_1_0-00_AOI.L5X
L_SUB	Subtract (64-bit)	Subtracts two LINT (signed 64-bit) values and returns a LINT (signed 64-bit) difference. Also provides math status bits for Carry (borrow), Negative, Overflow and Zero result (equivalent to built-in S:C, S:N, S:V, S:Z).	L_SUB_1_0-00_AOI.L5X
L_XIC	Examine On (64-bit)	This instruction examines the given Bit of the input LINT (64-bit integer) and outputs **true** (1) if the bit is 1, **false** (0) if the bit is 0. **IMPORTANT:** Use the output rung state or EnableOut to feed downstream logic. The output bit 'Out' reflects the state of the given bit only, for ladder animation, and **not** the rung state. If the given bit number is outside the range 0…63, the controller **major faults** on an invalid array index (bad bit number). There is no validity checking of the given bit number.	L_XIC_1_0-00_AOI.L5X
L_XIO	Examine Off (64-bit)	This instruction converts an LTIME (64-bit integer timestamp, for example, from an ALMD or ALMA instruction or the WALLCLOCKTIME object) to a DateTime (Year, Month, Day, Hour, Minute, Second, Microsecond as DINTs) in Coordinated Universal Time (UTC offset = 0). The input LTIME is the 64-bit (LINT) number of microseconds since DT#1970-01-01_00:00:00.000000 UTC.	L_XIO_1_0-00_AOI.L5X

Table 39 - Long Integer Instructions

Name	Short Description	Long Description	File Name
L_XOR	Bitwise XOR (64-bit)	This instruction returns the bitwise Logical XOR (exclusive OR, output bit **true** if either but NOT both of the corresponding input bits are **true**) of two input 64-bit integers (LINTs), into an output 64-bit integer.	L_XOR_1_0-00_AOI.L5X
T_LtoT	LTIME to DateTime	This instruction converts an LTIME (64-bit integer timestamp, for example, from an ALMD or ALMA instruction or the WALLCLOCKTIME object) to a DateTime (Year, Month, Day, Hour, Minute, Second, Microsecond as DINTs) in Coordinated Universal Time (UTC offset = 0). The input LTIME is the 64-bit (LINT) number of microseconds since DT#1970-01-01_00:00:00.000000 UTC.	T_LtoT_1_0-00_AOI.L5X
T_TtoL	DateTime to LTIME	This instruction converts a DateTime (Year, Month, Day, Hour, Minute, Second, Microsecond as DINTs) in Coordinated Universal Time (UTC offset = 0) to an LTIME (64-bit integer timestamp, for example, from an ALMD or ALMA instruction or the WALLCLOCKTIME object). The output LTIME is the 64-bit (LINT) number of microseconds since DT#1970-01-01_00:00:00.0000 UTC.	T_TtoL_1_0-00_AOI.L5X

Time and Date Instructions

The Rockwell Automation Library of Process Objects also includes instructions for performing date and time functions.

The time and date instructions are **calculation functions only**, and no HMI components are provided.

Table 40 - Time and Date Instructions

Name	Short Description	Long Description	File Name
T_ADD	DateTime:= DateTime + Time	T_Add; Add Date/Time plus time to get new Date/Time This instruction adds a given amount of Time to a Date/Time to arrive at a new Date/Time. The new Date/Time is 'normalized', that is, given as a valid (if possible) Gregorian Date and Time: • 0 <= Microseconds < 1,000,000 • 0 <= Seconds < 60 (This instruction cannot add leap seconds) • 0 <= Minutes < 60 • 0 <= Hours < 24 • 1 <= Day <= 31 **and** Date is a valid Gregorian date • 1 <= Month <= 12	T_ADD_1_0-01_AOI.L5X
T_Clock	Date/Time Clock	This object manages the controller 'Wall Clock', providing date and time services, including: • Accepts downloaded date and time for HMI or other sync source and sets the clock • Reads the clock and provides the local date and time to other logic – **IMPORTANT:** Current date/time is provided as individual DINTs and as a Date Time type for use with Date/Time math instructions (T_ADD, T_SUB, T_GRT, and so forth) • Calculates and provides the day of the week for the current date – **IMPORTANT:** Use T_DoW to calculate the day of the week for **any** given date • Optionally sets a flag once a day to request a clock sync update • Based on configured shift start times, determines the current production shift (for up to three shifts) The controller clock can be synchronized by writing a valid year, month, day, hour, minute and second into the appropriate settings. When the clock has been set, the settings are returned to '-1' and the time is reflected in the corresponding values and status.	T_Clock_1_0-01_AOI.L5X
T_DIFF	Time:= DateTime - DateTime	T_DIFF: Date/Time minus Date/Time gives time difference This instruction is given two Date/Time points and determines the amount of time between them. The result is given in days, hours, minutes, seconds and microseconds. (Years and months are returned as zero, as the number of months is generally not used.) The Date/Time parameters must be valid Gregorian Dates and valid clock times: • 0 <= Microseconds < 1,000,000 • 0 <= Seconds < 60 (This instruction cannot add leap seconds) • 0 <= Minutes < 60 • 0 <= Hours < 24 • 1 <= Day <= 31 **and** Date is a Valid Gregorian Date • 1 <= Month <= 12	T_DIFF_1_0-00_AOI.L5X
T_DoW	Day of the Week	T_DoW: Day of the Week This instruction takes a given Date/Time, and, for the date part, returns the day of the week: (0 = Sun, 1 = Mon, 2 = Tue, 3 = Wed, 4 =T hu, 5 = Fri, 6 = Sat) If the given date is **invalid**, a flag is set (but the calculated day of the week is returned anyway.) **IMPORTANT:** The time part of input parameter DT (hours, minutes, seconds, microseconds) is ignored.	T_DoW_1_0-00_AOI.L5X

Table 40 - Time and Date Instructions

Name	Short Description	Long Description	File Name
T_DST	Daylight Savings Time	This instruction manages Daylight Saving Time. It uses a number of configuration values to allow handling a wide variety of national and regional rules for when to start and end Daylight Saving Time (or 'summer time'). For use with HMI, it also provides values for display of the Month/Day and Hour/Minute of the points in time when DST starts and ends. Plus, for logging logic, it provides bits to indicate when timestamps have an overlap (1:30 a.m. happens twice) or there is a gap (one-shot). Follow these steps for best results. 1. Clear Cfg_EnableDST to 0. 2. Open the Controller Properties, clear the DST checkbox and set the clock to local STANDARD time 3. Configure the T_DST instruction per the instructions below. 4. Set the Cfg_EnableDST bit to 1. The clock is switched to DST based on the rules entered if DST is currently in effect for your location. Configuration: • Cfg_FwdMo -- Month specified in rule for date to spring forward (1…12) • Cfg_FwdOccur -- Occurrence of day of week to spring forward 1 = 1st, 2 = 2nd…5 = last • Cfg_FwdDoW -- Day of the week to spring forward (0 = Sun…6 = Sat) • Cfg_FwdDoM -- Day of month for spring forward if on a fixed date (1…31) • Cfg_FwdDoWBefore -- Day of the week BEFORE the first…last day of week or date (0 = Sun…6 = Sat) • Cfg_FwdHr --Hour (LOCAL) to spring forward (0…23) • Cfg_FwdMin -- Minute (LOCAL) to spring forward (0…59) • Cfg_FwdFixedDate --1 = Spring forward on fixed date, 0 = on occurrence of day of week • Cfg_FwdUseBefore --1 = Spring forward on day of week **before** date or day of week • Cfg_BackMo -- Month specified in rule for date to fall back (1…12) • Cfg_BackOccur -- Occurrence of day of week to fall back 1 = 1st, 2 = 2nd…5 = last • Cfg_BackDoW -- Day of the week to fall back (0 = Sun…6 = Sat) • Cfg_BackDoM -- Day of month for fall back if on a fixed date (1…31) • Cfg_BackDoWBefore -- Day of the week BEFORE the first…last day of week or date (0 =S un…6 = Sat) • Cfg_BackHr --Hour (LOCAL) to fall back (0…23) • Cfg_BackMin -- Minute (LOCAL) to fall back (0…59) • Cfg_BackFixedDate -- 1 = Fall back on fixed date, 0=on occurrence of day of week • Cfg_BackUseBefore: --1 = Fall back on day of week **before** date or day of week • Cfg_Offset --Number of minutes to spring forward or fall back (0…1439, def = 60) • Cfg_EnableDST --1 = Automatically adjust clock for DST, 0 = Always Standard Time, no DST	T_DST_1_0-01_AOI.L5X

Configuration Values for T_DST for Sample Rulesets
(T_DST was tested in each of these configurations)

Parameter	U.S./Can (default)	European Union	Russia	Morocco	Israel	New Zealand	Newfoundland(1988)
"Spring Forward" Rule	Second Sunday in March at 02:00 Local	Last Sunday in March at 01:00 UTC	Saturday before Last Sunday in March at 23:00 UTC	For 2014: March 30 at 02:00 Local	Last Friday before April 2 at 02:00 Local	Last Sunday in September at 02:00 Local	Second Sunday in March at 00:01 Local (advance 2 hours)
Cfg_FwdMo	3	3	3	3	4	9	3
Cfg_FwdOccur	2	5	5	---	---	5	2
Cfg_FwdDoW	0	0	0	---	---	0	0
Cfg_FwdDoM	-	-	-	**30**	**2**	-	-
Cfg_FwdDoWBefore	-	-	**6**	-	**5**	-	-
Cfg_FwdHr	2	varies by zone	varies by zone	2	2	2	0
Cfg_FwdMin	0	0	0	0	0	0	**1**
Cfg_FwdFixedDate	0 (false)	0 (false)	0 (false)	**1 (true)**	**1 (true)**	0 (false)	0 (false)
Cfg_FwdUseBefore	0 (false)	0 (false)	**1 (true)**	0 (false)	**1 (true)**	0 (false)	0 (false)
"Fall Back" Rule	First Sunday in November at 02:00 Local	Last Sunday in October at 01:00 UTC	Saturday before Last Sunday in October at 23:00 UTC	For 2014: June 28 at 02:00 Local	Sunday between Rosh Hashanah and Yom Kippur (varies)	First Sunday in April at 03:00 Local	First Sunday in November at 00:01 Local (fall back 2 hours)
Cfg_BackMo	11	10	10	6	varies	4	11
Cfg_BackOccur	1	5	5	---	varies	1	1
Cfg_BackDoW	0	0	0	---	0	0	0
Cfg_BackDoM	-	-	-	**28**	-	-	-
Cfg_BackDoWBefore	-	-	**6**	-	-	-	-
Cfg_BackHr	2	varies by zone	varies by zone	2	2	2	0
Cfg_BackMin	0	0	0	0	0	0	**1**
Cfg_BackFixedDate	0 (false)	0 (false)	0 (false)	**1 (true)**	0 (false)	0 (false)	0 (false)
Cfg_BackUseBefore	0 (false)	0 (false)	**1 (true)**	0 (false)	0 (false)	0 (false)	0 (false)
Cfg_Offset	60	60	60	60	60	60	**120**
Cfg_EnableDST	1	1	1	1	1	1	1

Name	Short Description	Long Description	File Name
T_EQU	DateTime = DateTime?	This instruction compares two Date-and-Time-of-Day (DateTime) variables. If DT1 is equal to DT2, EnableOut and Out are set to **true** (1). Otherwise EnableOut and Out are cleared to **false** (0). This instruction can be used in Ladder Diagram, Structured Text, or Function Block Routines just like the EQU instruction is used for integers and floating point numbers. However, because it has InOut Parameters (tag references to user-defined types), it is **not** left-justified on ladder rungs. On a **False** rung in LD, or in FBD with EnableIn cleared to 0, Out is cleared to 0. **IMPORTANT:** This instruction is dependent on the user-defined type 'DateTime' (external to this Add-On Instruction definition). 'DateTime' is defined as the following: • Yr -- DINT (year) • Mo -- DINT (month) • Da -- DINT day) • Hr -- DINT (hour) • Min -- DINT (minute) • Sec -- DINT (second) • uSec -- DINT (microsecond)	T_EQU_1_0-01_AOI.L5X

Name	Short Description	Long Description	File Name
T_GEQ	DateTime >= DateTime?	This instruction compares two Date-and-Time-of-Day (DateTime) variables. If DT1 is greater than (after) or equal to DT2, EnableOut and Out are set to **true** (1). Otherwise EnableOut and Out are cleared to **false** (0). This instruction can be used in Ladder Diagram, Structured Text, or Function Block Routines just like the GEQ instruction is used for integers and floating point numbers. However, because it has InOut Parameters (tag references to user-defined types), it is **not** left-justified on ladder rungs. On a **False** rung in LD, or in FBD with EnableIn cleared to 0, Out is cleared to 0. **IMPORTANT:** This instruction is dependent on the user-defined type 'DateTime' (external to this Add-On Instruction definition). 'DateTime' is defined as the following: • Yr -- DINT (year) • Mo -- DINT (month) • Da -- DINT day) • Hr -- DINT (hour) • Min -- DINT (minute) • Sec -- DINT (second) • uSec -- DINT (microsecond)	T_GEQ_1_0-01_AOI.L5X
T_GRT	DateTime > DateTime?	This instruction compares two Date-and-Time-of-Day (DateTime) variables. If DT1 is greater than (after) DT2, EnableOut and Out are set to **true** (1). Otherwise EnableOut and Out are cleared to **false** (0). This instruction can be used in Ladder Diagram, Structured Text, or Function Block Routines just like the GRT instruction is used for integers and floating point numbers. However, because it has InOut Parameters (tag references to user-defined types), it is **not** left-justified on ladder rungs. On a **False** rung in LD, or in FBD with EnableIn cleared to 0, Out is cleared to 0. **IMPORTANT:** This instruction is dependent on the user-defined type 'DateTime' (external to this Add-On Instruction definition). 'DateTime' is defined as the following: • Yr -- DINT (year) • Mo -- DINT (month) • Da -- DINT day) • Hr -- DINT (hour) • Min -- DINT (minute) • Sec -- DINT (second) • uSec -- DINT (microsecond)	T_GRT_1_0-01_AOI.L5X

Name	Short Description	Long Description	File Name
T_LEQ	DateTime <= DateTime?	This instruction compares two Date-and-Time-of-Day (DateTime) variables. If DT1 is less than (before) or equal to DT2, EnableOut and Out are set to **true** (1). Otherwise EnableOut and Out are cleared to **false** (0). This instruction can be used in Ladder Diagram, Structured Text, or Function Block Routines just like the LEQ instruction is used for integers and floating point numbers. However, because it has InOut Parameters (tag references to user-defined types), it is **not** left-justified on ladder rungs. On a **False** rung in LD, or in FBD with EnableIn cleared to 0, Out is cleared to 0. **IMPORTANT:** This instruction is dependent on the user-defined type 'DateTime' (external to this Add-On Instruction definition). 'DateTime' is defined as the following: • Yr -- DINT (year) • Mo -- DINT (month) • Da -- DINT day) • Hr -- DINT (hour) • Min -- DINT (minute) • Sec -- DINT (second) • uSec -- DINT (microsecond)	T_LEQ_1_0-01_AOI.L5X
T_LES	DateTime < DateTime?	This instruction compares two Date-and-Time-of-Day (DateTime) variables. If DT1 is less than (before) DT2, EnableOut and Out are set to **true** (1). Otherwise EnableOut and Out are cleared to **false** (0). This instruction can be used in Ladder Diagram, Structured Text, or Function Block Routines just like the LES instruction is used for integers and floating point numbers. However, because it has InOut Parameters (tag references to user-defined types), it is **not** left-justified on ladder rungs. On a **False** rung in LD, or in FBD with EnableIn cleared to 0, Out is cleared to 0. **IMPORTANT:** This instruction is dependent on the user-defined type 'DateTime' (external to this Add-On Instruction definition). 'DateTime' is defined as the following: • Yr -- DINT (year) • Mo -- DINT (month) • Da -- DINT day) • Hr -- DINT (hour) • Min -- DINT (minute) • Sec -- DINT (second) • uSec -- DINT (microsecond)	T_LES_1_0-01_AOI.L5X

Name	Short Description	Long Description	File Name
T_LIM	DateTime Limit Test	This instruction compares a Date-and-Time-of-Day or amount of time with a High Limit (Date/Time or amount of time) and a Low Limit (Date/Time or amount of time). There are two cases: • 'normal case' (Low Limit <= High Limit) • 'circular case' (Low Limit > High Limit) In the normal case, EnableOut and Out are set if: Low Limit <= DateTime <= High Limit In the circular case, EnableOut and Out are set if: DateTime >= Low Limit OR DateTime <= High Limit (remember, High Limit < Low Limit) This instruction can be used in Ladder Diagram, Structured Text, or Function Block Routines just like the LIM instruction is used for integers and floating point numbers. However, because it has InOut Parameters (tag references to user-defined types), it is **not** left-justified on ladder rungs. On a **False** rung in LD, or in FBD with EnableIn cleared to 0, Out is cleared to 0. **IMPORTANT:** This instruction is dependent on the user-defined type 'DateTime' (external to this Add-On Instruction definition). 'DateTime' is defined as the following: • Yr -- DINT (year) • Mo -- DINT (month) • Da -- DINT day) • Hr -- DINT (hour) • Min -- DINT (minute) • Sec -- DINT (second) • uSec -- DINT (microsecond)	T_LIM_1_0-00_AOI.L5X
T_LtoT	LTIME to DateTime	This instruction converts an LTIME (64-bit integer timestamp, for example, from an ALMD or ALMA instruction or the WALLCLOCKTIME object) to a DateTime (year, month, day, hour, minute, second, microsecond as DINTs) in Coordinated Universal Time (UTC offset = 0). The input LTIME is the 64-bit (LINT) number of microseconds since DT#1970-01-01_00:00:00.000000 UTC.	T_LtoT_1_0-01_AOI.L5X

Name	Short Description	Long Description	File Name
T_NEQ	DateTime <> DateTime?	This instruction compares two Date-and-Time-of-Day (DateTime) variables. If DT1 is not equal to DT2, EnableOut and Out are set to true (1). Otherwise EnableOut and Out are cleared to false (0). This instruction can be used in Ladder Diagram, Structured Text, or Function Block Routines just like the NEQ instruction is used for integers and floating point numbers. However, because it has InOut Parameters (tag references to user-defined types), it is **not** left-justified on ladder rungs. On a **False** rung in LD, or in FBD with EnableIn cleared to 0, Out is cleared to 0. **IMPORTANT:** This instruction is dependent on the user-defined type 'DateTime' (external to this Add-On Instruction definition). 'DateTime' is defined as the following: • Yr -- DINT (year) • Mo -- DINT (month) • Da -- DINT day) • Hr -- DINT (hour) • Min -- DINT (minute) • Sec -- DINT (second) • uSec -- DINT (microsecond)	T_NEQ_1_0-01_AOI.L5X
T_Now	Current DateTime	Returns the current local date and time from the controller clock as a DateTime In/Out Parameter. This instruction depends on the (external) DateTime data type: • Yr -- DINT (year) • Mo -- DINT (month) • Da -- DINT day) • Hr -- DINT (hour) • Min -- DINT (minute) • Sec -- DINT (second) • uSec -- DINT (microsecond)	T_Now_1_0-00_AOI.L5X
T_Scan	Time Since Previous Scan	Returns the time between the previous scan of the instance and the current scan of the same instance as a REAL number of Seconds.	T_Scan_1_0-01_AOI.L5X
T_SEL	DateTime Select	This instruction uses an Input bit signal to select one of two Date/Time values. **IMPORTANT:** The selected Date/Time is only 'moved through' as-is; it is **not** 'normalized' to a valid Gregorian Date and Time: Inp_Sel is defaulted to 1 so this instruction can be used on a Ladder Diagram Routine rung with the rung condition as the selector: Rung True selects DT1, Rung False selects DT0. The Inp_Sel is **inverted** when EnableIn is **False** (false rung). This inversion can be useful beyond this ladder diagram function.	T_SEL_1_0-00_AOI.L5X
T_SUB	DateTime:= DateTime - Time	T_Sub: Subtract Date/Time minus time to get new Date/Time. This instruction subtracts a given amount of time from a Date/Time to arrive at a new Date/Time. The new Date/Time is 'normalized', that is, given as a valid (if possible) Gregorian Date and Time: • 0 <= Microseconds < 1,000,000 • 0 <= Seconds < 60 (This instruction cannot add leap seconds) • 0 <= Minutes < 60 • 0 <= Hours < 24 • 1 <= Day <= 31 AND Date is a Valid Gregorian Date • 1 <= Month <= 12	T_SUB_1_0-01_AOI.L5X

Name	Short Description	Long Description	File Name
T_Sun	Sun Rise / Set / Az/El	This instruction takes a given Date/Time, and for the date part, plus the configured latitude, longitude and UTC offset, returns the Date/Time of local sunrise and local sunset (to the nearest minute, to an accuracy of within about two minutes). Solar Azimuth (heading, clockwise in degrees from true north) and Elevation (degrees above horizon) are calculated and accurate to within about half a degree when the sun is above the horizon. Azimuth bearing is not necessarily accurate when elevation is more than a degree or two negative. The given date is assumed **valid**. If necessary, check by using T_Valid first. **IMPORTANT:** This instruction only uses the month and day to estimate the sunrise and sunset times to within a couple minutes. It does **not** deal with detailed astronomical calculations based on planetary models. It is based on the current Gregorian calendar and does **not** deal with Julian dates for dates before 1582. For algorithms, see: http://www.srrb.noaa.gov/highlights/sunrise/solareqns.PDF To get your latitude and longitude, see: http://www.batchgeocode.com/lookup/	T_Sun_1_0-00_AOI.L5X

Name	Short Description	Long Description	File Name
T_Sync	Synchronize Controller Clock	This object synchronizes the controller real-time clock with a 'reliable' NTP Time Server (computer responsible for network time, or a standard time server, like time.windows.com) **IMPORTANT:** This is **not** a full NTP precision exchange. It is simply a quick 'get' of the NTP server time and applying it to the controller clock. It supports the following features: • Ability to sync controller clock to server on Maintenance command • Ability to sync on a periodic (default = daily) basis • Ability to retry on a periodic (default = hourly) basis on a failure to retrieve date/time from server • Ability to sync on controller powerup or PROGRAM to RUN transition. • Ability via configuration to allow or disallow each of the above methods • Reads time from NTP server and displays time received as Values. • Updates the controller clock to time received (if allowed by configuration) • Calculates clock drift (difference between previous and new controller time) and displays as Values. Cfg_PollT: The Poll Time (def = 1440 min = 1 day) is the number of minutes between polls of the NTP server for reliable time after a **successful** get of the time. Cfg_RetryT: The Retry Time (def = 60 min = 1 hour) is the number of minutes between polls of the NTP server for reliable time after a **failure** to get the time. Cfg_ENSlotNumber: Enter the chassis slot number of the EtherNet/IP module (for example, 1756-EN2T) that can communicate with the time server. This module must support 'socket services'. For more information, see the EtherNet/IP Socket Interface Application Technique, publication ENET-AT002. Cfg_IPOctetX: Enter the four octets of the time server's IP Address, for example, 192.168.1.1 (in decimal) Cfg_AllowClockUpdate: 1 = Allow AOI to update controller clock. 0 = Just get the time (UTC) and display it. Cfg_AllowMCmdSync: 1 = Permit manual sync request via MCmd_Sync. 0 = No manual sync request allowed. Cfg_AllowPeriodicSync: 1 = Permit enabling periodic clock sync via MCmd_Enable. 0 = Periodic sync is kept disabled. Cfg_SyncOnPwrup: 1 = Request time sync on controller first scan. 0 = Do not initiate sync on controller first scan.	T_Sync_1_0-04_AOI.L5X T_Sync_1_0-04_RUNG.L5X

Name	Short Description	Long Description	File Name
T_TtoISO	Date to ISO Week Date	This object converts a Date in common form (Year, Month, Day) to an ISO-8601 Week Date (like 2014-W04-2, meaning Tuesday in the fourth week of Week-Numbered Year 2014) for 2014-01-21. This object calculates the ISO Year, ISO Week and ISO Day (day of the week). Note the ISO Day is specified as 1=Monday . . .7=Sunday. This object also determines the number of weeks (52 or 53) in the calculated ISO Year. The Date to convert is given in the Year, Month and Day of the Ref_DT reference tag, of type DateTime (Year, Month, Day, Hour, Minute, Second, Microsecond). The Week-Numbered Year does not necessarily start on January 1. It can start from December 29 through January 4. The first week of an ISO Week-Numbered Year is the week, beginning with Monday and ending on Sunday, that contains the first THURSDAY of the calendar year. See Wikipedia, 'ISO Week Date', for more information.	T_TtoISO_1_0-00_AOI.L5X
T_TtoL	DateTime to LTIME	This instruction converts a DateTime (year, month, day, hour, minute, second, microsecond as DINTs) in Coordinated Universal Time (UTC offset = 0) to an LTIME (64-bit integer timestamp, for example, from an ALMD or ALMA instruction or the WALLCLOCKTIME object). The output LTIME is the 64-bit (LINT) number of microseconds since DT#1970-01-01_00:00:00.0000 UTC.	T_TtoL_1_0-00_AOI.L5X
T_TtoS	Date/Time to String	This instruction takes the given date and formats it as a human-readable STRING. For example, for the Date/Time: 2008 12 31 23 59 59 999999 the return STRING is (based on configuration): Wednesday, December 31, 2008 11:59:59.999999 p.m. Options are provided for: • 24- or 12-hour time format (with a.m. or p.m. indicator on the 12-hour format) • Displaying or not displaying microseconds • Displaying or not displaying seconds • Displaying or not displaying the Day of the Week • Displaying day first (31 July) or month first (July 31) • Displaying date in an ISO-format (YYYY-MM-DD) This instruction checks for a valid (Gregorian) date and time and returns 'Invalid Date' and/or 'Invalid Time' as appropriate in the output STRING. The following are valid dates/times: • 0 <= Microseconds < 1,000,000 • 0 <= Seconds < 60 • 0 <= Minutes < 60 • 0 <= Hours < 24 • 0 <= Days <= 31 and a valid Gregorian Date (Feb = 28 or 29 days) • 1 <= Month <= 12 • Year in the range +/- 5879600 The names of the days of the week and the months of the year, and the AM and PM indicator text can be changed by using the Local Tags Monitor for the instance.	T_TtoS_1_0-00_AOI.L5X

Name	Short Description	Long Description	File Name
T_Valid	Is DateTime Valid?	This instruction tests the given DateTime variable and verifies that it is a valid calendar date and clock time, as follows: • 0 <= Microseconds < 1,000,000 • 0 <= Seconds < 60 (This instruction cannot check leap seconds) • 0 <= Minutes < 60 • 0 <= Hours < 24 • 1 <= Day <= 31 **and** Day is Valid for Gregorian Date (28, 29, 30, or 31 days in month • 1 <= Month <= 12 • Year is within the range of dates that this instruction can calculate a Gregorian day number (about +/- 5.8 million years) **IMPORTANT:** This instruction does **not** switch to Julian dates for dates before 1582 (or 1753). This instruction assumes the Gregorian Calendar extends 'indefinitely' (at least 5.8 million years) either side of 'zero'. It **does** handle the Gregorian 4-, 100-, 400-year rules, so Feb. 29, 2000 is a valid date, but Feb. 29, 2100 is not.	T_Valid_1_0-00_AOI.L5X

Notes:

HMI Security Codes Configuration

FactoryTalk View software security codes protect information that is contained within HMI faceplates. Operators, maintenance personnel, and engineers must have security permission to modify their respective faceplate tabs.

Display elements (global objects) have an associated faceplate that appears when the display element is clicked. Variables, setpoints, alarms, and other device configuration data is entered and viewed on the HMI faceplates.

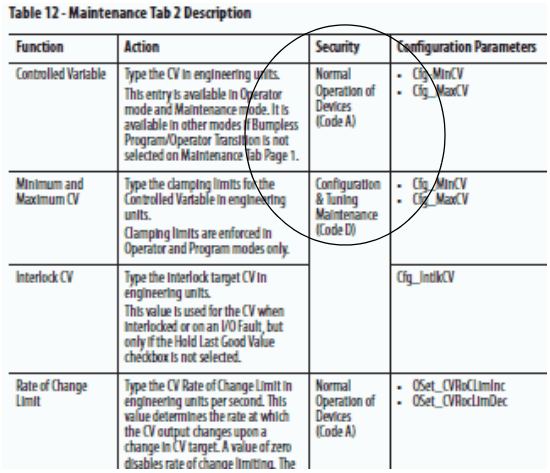


Table 12 - Maintenance Tab 2 Description

Function	Action	Security	Configuration Parameters
Controlled Variable	Type the CV in engineering units. This entry is available in Operator mode and Maintenance mode. It is available in other modes if Bumpless Program/Operator Transition is not selected on Maintenance Tab Page 1.	Normal Operation of Devices (Code A)	• Cfg_MinCV • Cfg_MaxCV
Minimum and Maximum CV	Type the clamping limits for the Controlled Variable in engineering units. Clamping limits are enforced in Operator and Program modes only.	Configuration & Tuning Maintenance (Code D)	• Cfg_MinCV • Cfg_MaxCV
Interlock CV	Type the interlock target CV in engineering units. This value is used for the CV when interlocked or on an I/O Fault, but only if the Hold Last Good Value checkbox is not selected.		Cfg_IntlkCV
Rate of Change Limit	Type the CV Rate of Change Limit in engineering units per second. This value determines the rate at which the CV output changes upon a change in CV target. A value of zero disables rate of change limiting. The	Normal Operation of Devices (Code A)	• OSet_CVRocLimInc • OSet_CVRocLimDec

Data cannot be entered or changed without an administrator granting permission with the corresponding security code on each faceplate tab. The Add-On Instruction documentation (see page 12) lists the security codes for the functions on the faceplates.

See Table 41 on page 182 for a description of the FactoryTalk View HMI security codes.

Table 41 - FactoryTalk View Software Code Descriptions

FTView A-P Security Code Configuration (as of 5/15/2013)	Normal Operation of Devices	Manual Device Operation (non coordinated)	Equipment Maintenance	Configuration & Tuning Maintenance	Engineering Configuration	Acknowledge and Shelve Alarms	Supervisory Operations	Disable Alarms Bypass Permissives and Interlocks	spare	Normal Production (Batches & Lots)	Setpoint and Parameter Override	Override/Force Sequences	Process Exception Handling, Advanced Production	Navigate Across Units/Applications	Shutdown Application, Operating System Access	Admin: Security, Users, Passwords
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**
Command Equipment in Operator Mode	X															
Enter Setpoints, Control Variables	X															
Reset Latched Interlocks, Restart Equipment	X															
Add Batch to Batch List, Run Batches										X						
Hold, Restart Batches, clear failures, bind, bind ack										X						
Acquire/Lock and Release Equipment Operator Mode		X														
Change Loop Mode (Manual, Auto, Cascade)	X															
Acquire/Release Equipment Maintenance Mode			X													
Reset Run Time Accumulators			X													
Override Inputs			X													
Bypass Feedback			X													
Enable/Disable Device			X													
Configuration (Limits, Constants, Timers)				X												
Modify Alarm Delay Times				X												
Tuning				X												
Change Machine Configuration					X											
Setup Configuration (Advanced)					X											
Alarm Configuration					X											
Put Device in Simulation					X											
Edit HMI Application					X											
Acknowledge Alarms						X										
Reset Alarms						X										
Shelve Alarms						X										
Disable Alarms								X								
Modify Alarm Limits and Deadbands								X								
Bypass Permissives and Interlocks								X								
Respond to Prompt (level 1)										X						
Respond to Prompt (level 2)													X			
Exception Processing (Resume, Manual, Auto, Semi-Auto, Pause, Disconnect, Release)													X			
Exception Processing (Step Change, Parameter Change, Step, Acquire, Reorder, Reactivate Step)												X				
Override Downloaded Setpoints											X					
Override Downloaded Phase Parameters											X					
Manual Batch Processing (Stop, Abort, Reset Phases)													X			
Manual Supervisory EP/EM Control							X									
Force Steps/States												X				
Change Inflights and Preacts				X												
Force Queue Indexing													X			
Navigate to Other Units														X		
Shutdown HMI Application															X	
Access Windows Start Menu, Windows Apps															X	
Change Accounts/Passwords																X
Change Security Settings																X
Process																
HMI Operators	X					X				X			X			
HMI Operating Supervisor	X	X				X	X	X		X	X	X	X	X		
HMI Maintenance	X	X	X			X	X	X		X	X	X	X		X	
HMI Maintenance Supervisor	X	X	X	X		X	X	X		X	X	X	X	X	X	
HMI Manager	X	X	X	X		X	X	X		X	X	X	X	X	X	
HMI Engineering	X	X	X	X	X	X	X	X		X	X	X	X	X	X	
HMI Admin														X	X	X

Faceplates for Built-in Logix5000 Instructions

The faceplates that appear in this section are designed to let the function blocks and built-in firmware instructions for the Logix5000 controllers interface with the Process Library Add-On Instructions.

For details on built-in instructions, see the Logix5000 Controllers Advanced Process Control and Drives Instructions Reference Manual, publication 1756-RM006.

Built-in Autotune

The RSLogix 5000 PIDE autotuner provides an open-loop autotuner that is built into the PIDE instruction. This function filters a signal to assist with the calculation of control variables. You can autotune from PanelView Plus terminals or any other operator interface devices as well as RSLogix 5000 software.

The PIDE block has an Autotune Tag (type PIDE_AUTOTUNE) that you specify for those PIDE blocks that you want to autotune.

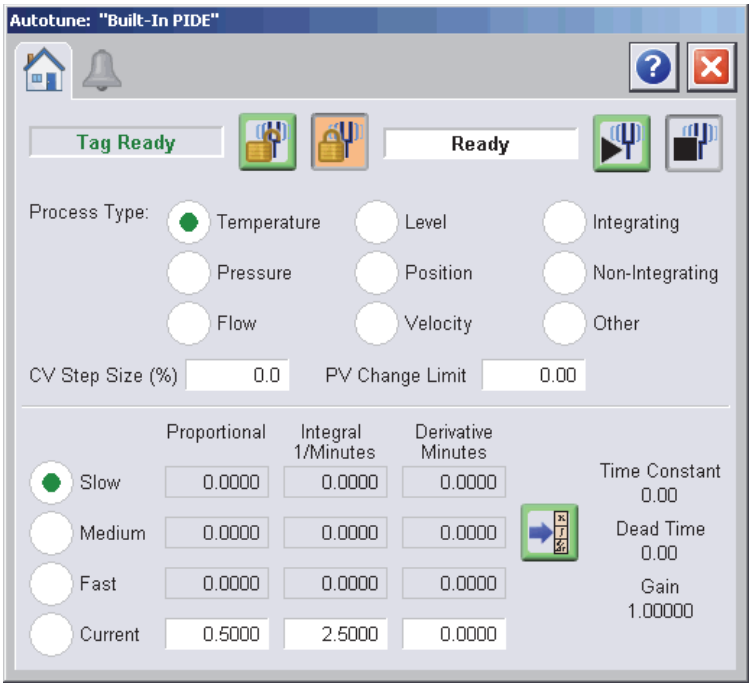
IMPORTANT	The PIDE autotuner is installed with RSLogix 5000 software, but you need an activation key to enable the autotuner. The autotuner is supported only in function block programming; it is not available in relay ladder or structured text programming.

The Autotune function is accessed through page 2 of the Maintenance tab of the Built-in PIDE faceplate.

Operator Tab

The Operator tab of the PIDE faceplate shows the following information:

- Proportional gain tuned slow, medium, or fast
- Integral gain tuned slow, medium, or fast
- Derivative gain tuned slow, medium, or fast
- Time constant
- Dead time
- Gain



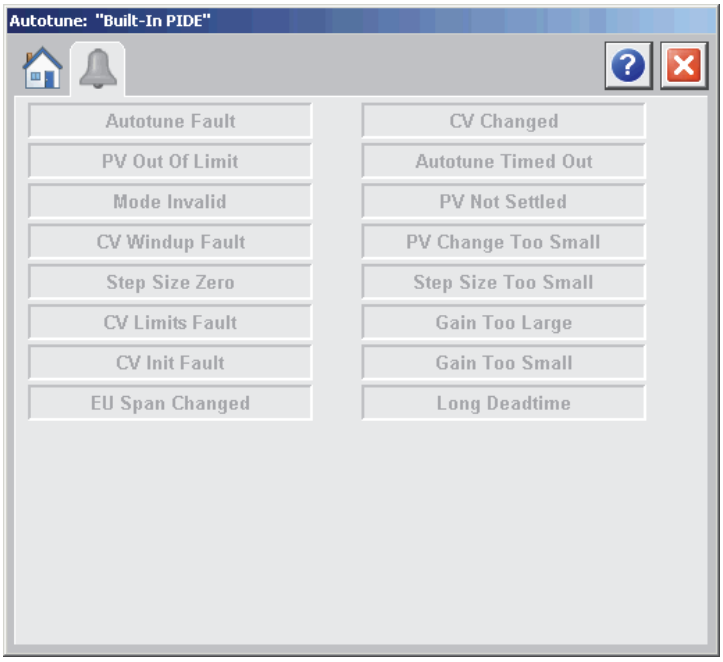
The following table lists the functions on the Autotune Operator tab.

Table 42 - Autotune Operator Tab Description

Function	Action	Security
	Click to release the Autotune tag.	Configuration and Tuning Maintenance (Code D)
	Click to acquire the Autotune tag.	
	Click to start the autotune process for CV1, CV2, or CV3.	
	Click to stop the autotune process.	Configuration and Tuning Maintenance (Code D)
	Click to replace the current model parameters with the calculated Autotune model parameters.	
Process Type	Click the item that best describes the process.	
CV Step Size (%)	Type a value for CV step size in percent for the tuning step test.	
PC Change Limit	Type a value for the PV Change Limit. The autotune is aborted if the PV changes by more than this amount.	
Gains: Proportional Integral Derivative	Type in a value for: Proportional gain Integral gain Derivative gain	

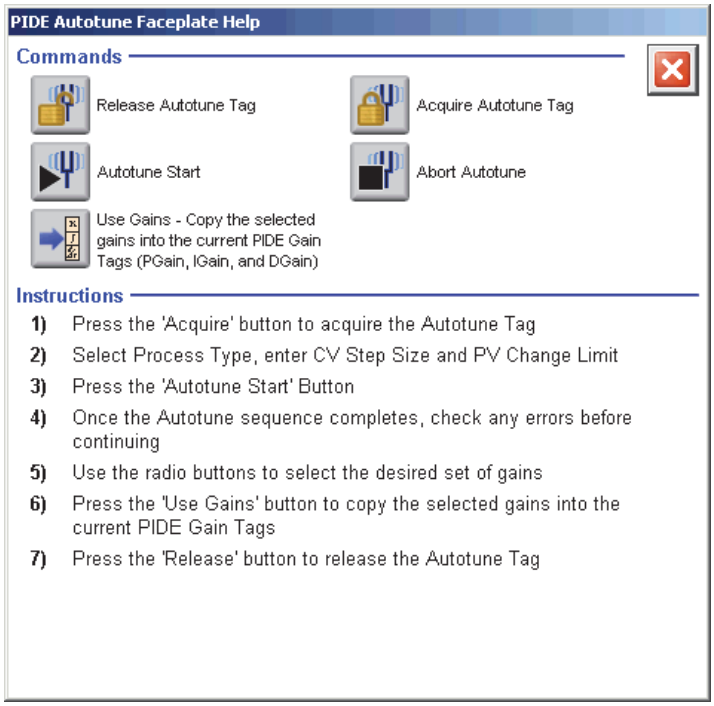
Alarms Tab

The Autotune Alarms tab shows all the available alarms for the device and the current status of each alarm.



Faceplate Help

The Faceplate Help page shows the indicators and command buttons that are used by the Built-in Autotune.



Coordinated Control (CC)

The Coordinated Control (CC) function block controls a single process variable by manipulating as many as three different control variables. As an option, any of the three outputs can be used as an input to create feed forward action in the control variable. The CC function block calculates the control variables (CV1, CV2, and CV3) in the Auto mode based on the PV - SP deviation, internal models, and tuning.

Visualization Files

The following files are required to use the CC object and can be downloaded from the Product Compatibility and Download Center at http://www.rockwellautomation.com/rockwellautomation/support/pcdc.page.

IMPORTANT Files must be imported in the following order: image files, then global object files, and then graphic files. This order is required to properly configure the visualization files.

Table 43 - CC Visualization File Types

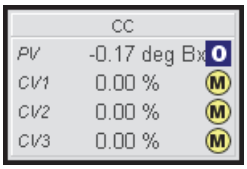
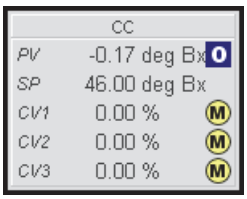
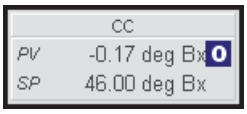
Application Type	File Type	FactoryTalk View SE Software	FactoryTalk View ME Software	Description
Graphics - Displays	GFX	(RA-BAS) Common Analog-Edit	N/A	Faceplate used for analog input data entry. The FactoryTalk View ME faceplates use the native analog input data entry so no file is required.
		(RA-BAS) Built-In CC Autotune-Faceplate	(RA-BAS-ME) Built-In CC Autotune-Faceplate	The faceplate display used for the Autotune object.
		(RA-BAS) Built-In CC Autotune-Help	(RA-BAS-ME) Built-In CC Autotune-Help	Help information that is accessed from the CC Autotune Help faceplate.
		(RA-BAS) Built-In CC Faceplate	(RA-BAS-ME) Built-In CC Faceplate	The faceplate display used for the CC object.
		(RA-BAS) Built-In CC Help	(RA-BAS-ME) Built-In CC Help	Help information that is accessed from the CC Help faceplate.
		(RA-BAS) Built-In CC Quick	(RA-BAS-ME) Built-In CC Quick	The Quick display used for the CC object.
Graphics - Global Objects	GGFX	(RA-BAS) Common Faceplate Objects	(RA-BAS-ME) Common Faceplate Objects	Common global objects used on all Process Object faceplates.
		(RA-BAS) BuiltIn Faceplate Objects	(RA-BAS-ME) BuiltIn Faceplate Objects	Global objects used on CC faceplates.
		(RA-BAS) BuiltIn Graphics Librarys	(RA-BAS-ME) BuiltIn Graphics Librarys	CC display elements used to build process graphics.
		(RA-BAS) BuiltIn Help Objects	(RA-BAS-ME) BuiltIn Help Objects	BuiltIn global objects used for all BuiltIn help displays.
Graphics - Images	BMP	All .png files in the images folder	All .png files in the images folder	These are the common icons used in the global objects and faceplates for all Process Objects.
HMI Tags	CSV	N/A	FTVME_PlantPAxLib_Tags_3_1_**00**.csv(1)	These tags must be imported into the FactoryTalk View ME project to support switching tabs on any Process Object faceplate.

(1) The service release number (boldfaced) can change as service revisions are created.

Display Elements

A display element (global object) is created once and can be referenced multiple times on multiple displays in an application. When changes are made to the original (base) object, the instantiated copies (reference objects) are automatically updated. Use of global objects, in conjunction with tag structures in the ControlLogix system, aid consistency and save engineering time.

Elements Descriptions

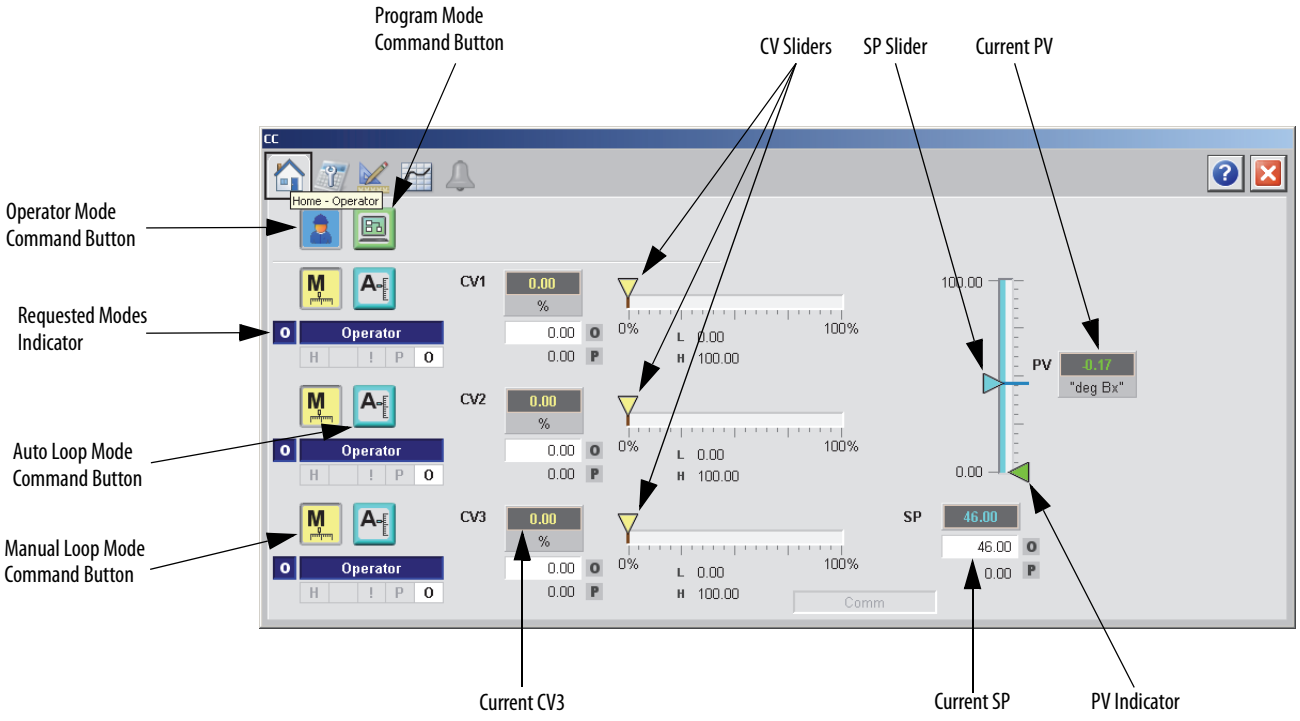
	Display Element	Description
		Coordinated Control object with a Process Variable and three Control Variables.
		Coordinated Control object with a Process Variable, Setpoint, and three Control Variables.
GO_BuiltIn_CC2		Coordinated Control object with a Process Variable and a Setpoint.

Operator Tab

The faceplate initially opens to the Operator ('Home') tab. From here, an operator can monitor the device status and manually operate the device when it is in Operator mode.

The Operator tab shows the following information:

- Requested modes indicator
- Current process variable and bar graph
- Current control variables and bar graph for each
- Current setpoint
- High (H) and Low (L) clamping limits for the CVs
- Input status (Communications OK, Communications Fail, Bad PV Quality, or Uncertain PV Quality)



The following table lists the functions on of the CC Operator tab.

Table 45 - CC Operator Tab Description

Function	Action	Security
	Click to request Operator mode.	Manual Device Operation (Code B)
	Click to request Program mode.	
	Click to request Manual Loop mode.	Normal Operation of Devices (Code A)
	Click to request Auto Loop mode.	
Operator CV value (CV1, CV2, and CV3)	Type a value for a CV output.	
CV Slider (CV1, CV2, and CV3)	Move this slider to adjust the loop CV output.	Equipment Maintenance (Code C)
SP Slider	Move this slider to adjust the loop setpoint.	
Operator Setpoint Value	Type a value for the loop Setpoint.	Normal Operation of Devices (Code A)

Maintenance Tab

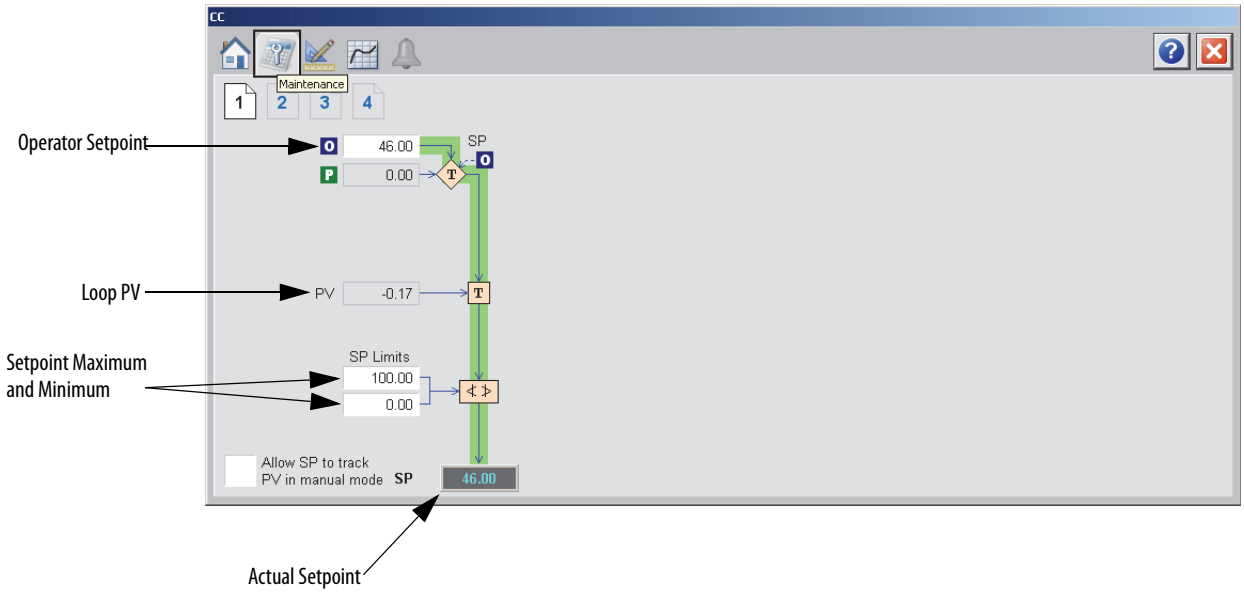
Maintenance personnel use the information and controls on the Maintenance tab to make adjustments to device parameters, troubleshoot and temporarily work around device problems, and disable the device for routine maintenance.

The Maintenance tab is divided into four tabs.

Maintenance Tab Page 1

Page 1 of the Maintenance tab shows the following information:

- Source of the setpoint, by animation of the data path and the transfer points
- Actual loop setpoint after selection and clamping
- Loop process variable (PV)



The following table shows the functions of page 1 of the Maintenance tab.

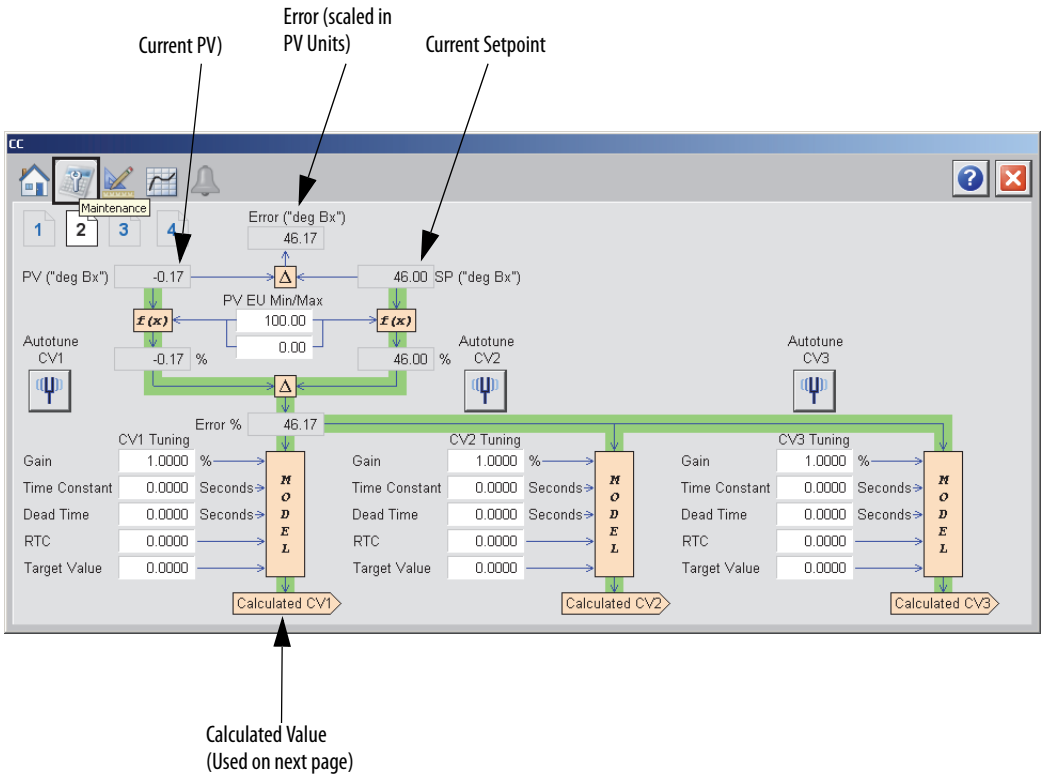
Table 46 - CC Maintenance Tab Page 1 Description

Function	Action	Security	Configuration Parameters
Operator Setpoint	Type the Operator setpoint.	Normal Operation of Devices (Code A)	.SPOper
Setpoint Limits	Type the maximum limit for the setpoint.	Configuration and Tuning Maintenance (Code D)	.SPHLimit
	Type the minimum limit for the setpoint.		.SPLLimit
Allow SP to track PV in manual mode	Click to Set **true** to enable CV Tracking when autotune is OFF. This parameter is ignored in Hand and Override mode.	Equipment Maintenance (Code C)	.PVTracking

Maintenance Tab Page 2

Page 2 of the CC Maintenance tab shows the following information:

- Error (scaled in PV units)
- Current process variable (PV)
- Current setpoint
- PV (percent of span)
- SP (percent of span)



The following table shows the functions of page 2 of the CC Maintenance tab.

Table 47 - CC Maintenance Tab Page 2 Description

Function	Action	Security	Configuration Parameters
PV EU Min/Max	Type the maximum limit for the PV in engineering units.	Engineering Configuration (Code E)	.PVEUMax
	Type the minimum limit for the PV in engineering units.		.PVEUMin
Gain (CV1, CV2, and CV3)	Enter the CV1, CV2, or CV3 gain for the appropriate model.	Configuration and Tuning Maintenance (Code D)	• .CV1ModelGain • .CV2ModelGain • .CV3ModelGail
Time Constant (CV1, CV2, and CV3)	Enter the CV1, CV2, or CV3 time constant for the appropriate model.		• .CV1ModelTC • .CV2ModelTC • .CV3ModelTC
Dead Time (CV1, CV2, and CV3)	Enter the internal model deadtime for CV1, CV2, or CV3 for the appropriate model.		• .CV1ModelDT • .CV2ModelDT • .CV3ModelDT

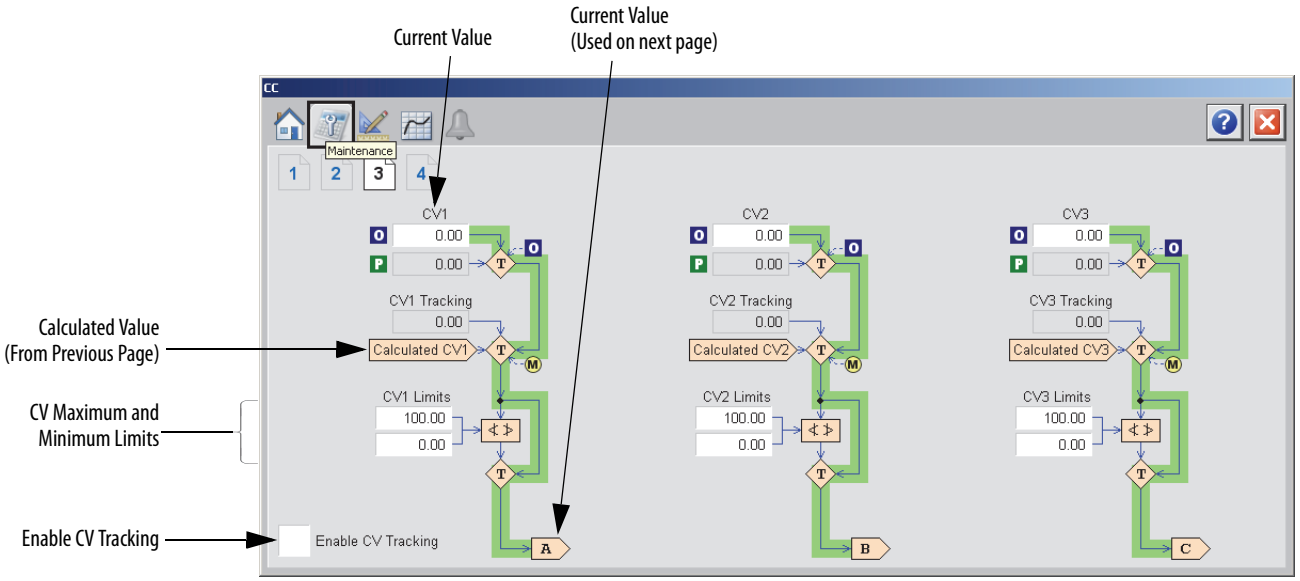
Table 47 - CC Maintenance Tab Page 2 Description

Function	Action	Security	Configuration Parameters
RTC (CV1, CV2, and CV3)	Enter the CV1, CV2, or CV3 response time constant for the appropriate model. This value determines the speed of the CV action in seconds.	Configuration and Tuning Maintenance (Code D)	• .CV1RespTC • .CV2RespTC • .CV3RespTC
Target Value (CV1, CV2, and CV3)	Enter the target value for CV1, CV2, or CV3 for the appropriate model.		• .CV1Target • .CV2Target • .CV3Target

Maintenance Tab Page 3

Page 3 of the CC Maintenance tab shows the following information:

- Program value in percent
- CV Track value
- Calculated CV (from Maintenance page 2)



The following table shows the functions of page 3 of the CC Maintenance tab.

Table 48 - CC Maintenance Tab Page 3 Description

Function	Action	Security	Configuration Parameters
CV1 (CV1, CV2, or CV3)	Type the value (%) for CV.	Normal Operation of Devices (Code A)	• .CV1Oper • .CV2Oper • .CV3Oper
CV Maximum Limit (CV1, CV2, or CV3)	Type the maximum limit for CV1, CV2, or CV3.	Configuration and Tuning Maintenance (Code D)	• .CV1HLimit • .CV2HLimit • .CV3HLimit
CV Minimum Limit (CV1, CV2, or CV3)	Type the minimum limit for CV1, CV2, or CV3.		• .CV1LLimit • .CV2LLimit • .CV3LLimit

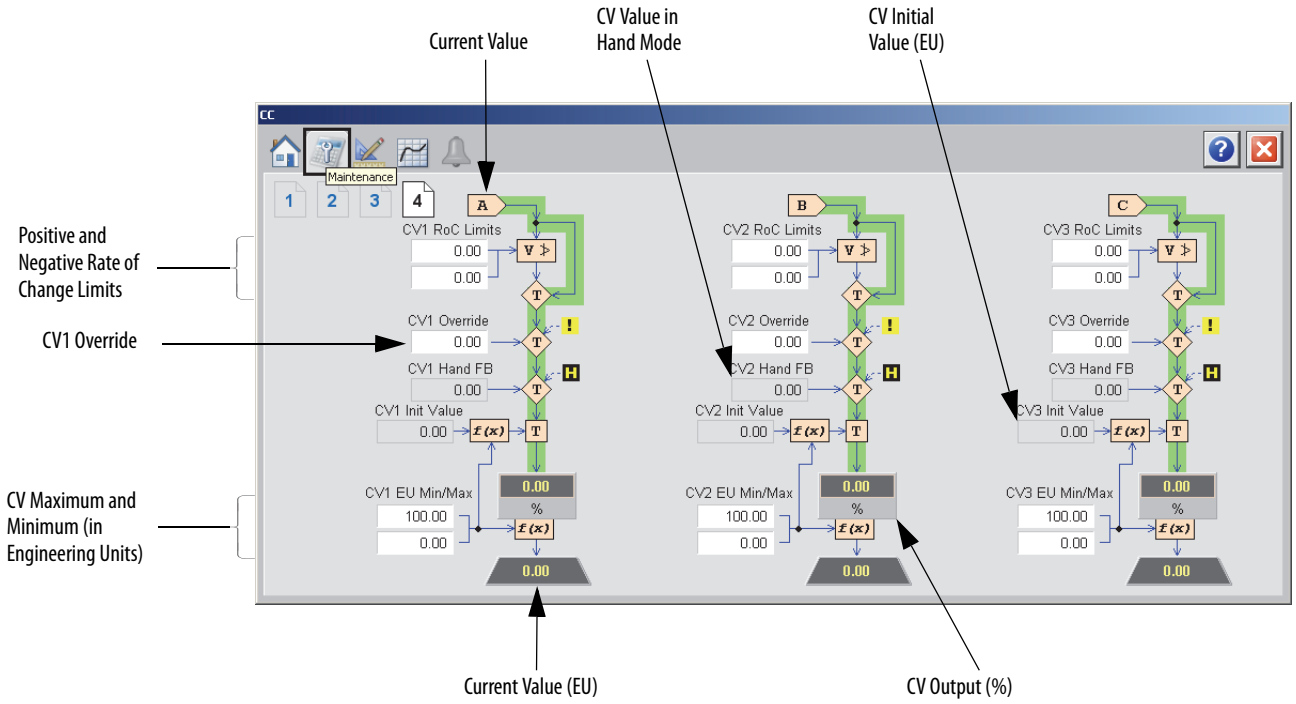
Table 48 - CC Maintenance Tab Page 3 Description

Function	Action	Security	Configuration Parameters
Enable CV Tracking	Check to enable CV tracking when autotune is OFF. This parameter is ignored in Hand and Override mode.	Equipment Maintenance (Code C)	.CVTrackReq

Maintenance Tab Page 4

Page 4 of the CC Maintenance tab shows the following information:

- CV value in Hand mode
- Initial CV value (EU)
- CV output value (EU)
- Current value (from Maintenance page 3)



The following table shows the functions of page 3 of the CC Maintenance tab.

Table 49 - CC Maintenance Tab Page 4 Description

Function	Action	Security	Configuration Parameters
CV RoC Positive Limit	Type the CV1, CV2, or CV3 positive or negative rate of change limit, in percent per second. Rate of change limiting is used only when in Auto mode or in Manual mode if CVManLimiting is **true**. A value of zero disables CV1 ROC limiting.	Configuration and Tuning Maintenance (Code D)	• .CV1ROCPosLimit • .CV2ROCPosLimit • .CV3ROCPosLimit • .CV1ROCNegLimit • .CV2ROCNegLimit • .CV3ROCNegLimit
CV RoC Negative Limit			
CV Override	Type the CV1, CV2, or CV3 Override value. CV1, CV2, or CV3 is set to this value when in Override mode. This value is recommended to correspond to a safe state output of the loop.		• .CV1OverrideValue • .CV2OverrideValue • .CV3OverrideValue

Table 49 - CC Maintenance Tab Page 4 Description

Function	Action	Security	Configuration Parameters
CV EU Maximum	Type the maximum value for CV1EU, CV2EU, or CV3EU. The value of CV1EU, CV2EU, or CV3EU that corresponds to 100% CV1, CV2, or CV3.	Engineering Configuration (Code E)	• .CV1EUMax • .CV2EUMax • .CV3EUMax
CV EU Minimum	Type the minimum value for CV1EU, CV2EU, or CV3EU. The value of CV1EU, CV2EU, or CV3EU that corresponds to 0% CV1, CV2, or CV3.		• .CV1EUMin • .CV2EUMin • .CV3EUMin

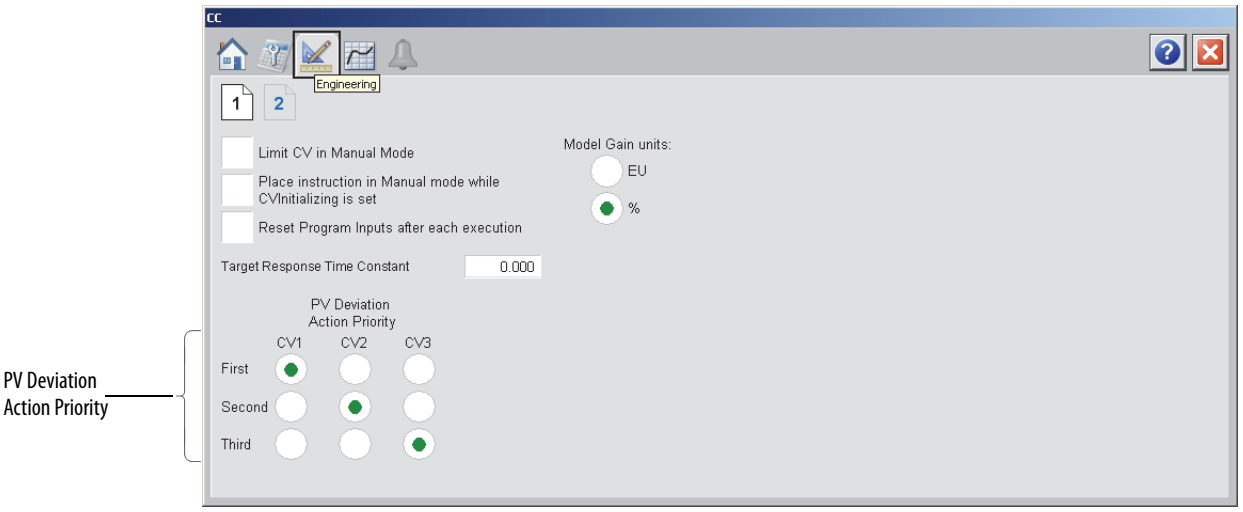
Engineering Tab

The Engineering tab provides access to device configuration parameters and ranges, options for device and I/O setup, displayed text, and faceplate-to-faceplate navigation settings, for initial system commissioning or later system changes.

The Engineering tab is divided into two pages.

Engineering Tab Page 1

Page 1 of the CC Engineering tab has various Operator inputs and options for the CV, model gain, and PV deviation.



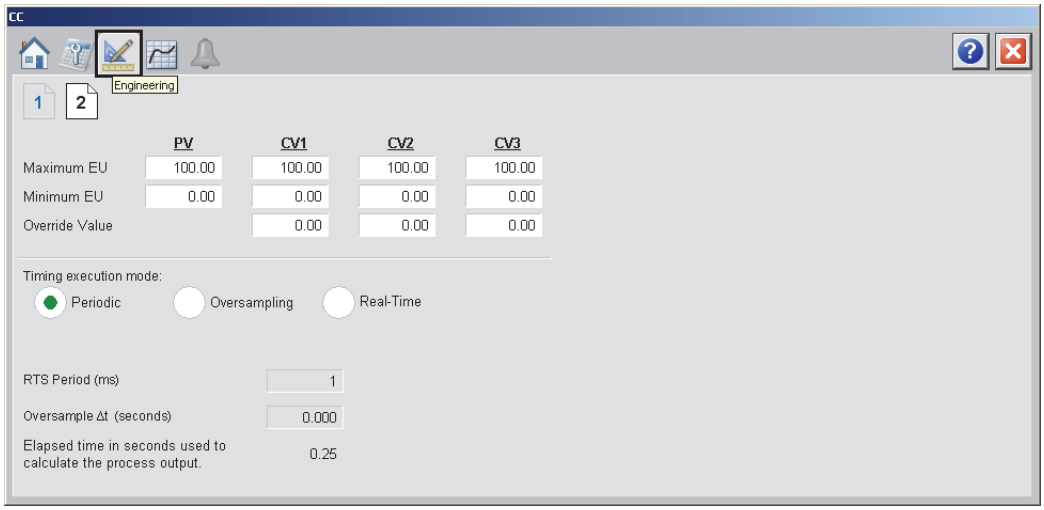
The following table shows the functions of page 1 of the CC Engineering tab.

Table 50 - CC Engineering Tab Page 1 Description

Function	Action	Security	Configuration Parameters
Limit CV in manual Mode	Check to limit the current value when in Manual mode.	Engineering Configuration (Code E)	.CVManLimiting
Place instruction in Manual mode while CVInitializing is set	Check to set the Loop mode to manual when CV initialization is requested. Clear the checkbox to leave the Loop mode unchanged when initialization is requested. When the initialization request clears, the loop resumes control in its previous Loop mode.		.ManualAfterInit
Reset Program Inputs after each execution	Check to reset Program inputs after each execution.		.ProgValueReset
Model Gain units	Select either 'EU' or '%' for the Model Gain units.		.GainEUSpan
Target Response Time Constant	Type a value for the Target Response Time Constant.		.TargetRespTC
PV Deviation Action Priority: First Second Third	Click to select CV1, CV2, or CV3 to be the first, second, or third to act to compensate for PV-SP deviation.		• .Act1stCV • .Act2ndCV • .Act3rdCV

Engineering Tab Page 2

Page 2 of the CC Engineering tab has various Operator inputs/options for the PV, CV, timing execution mode, RTS period, oversample, and time used to calculate output.



The following table shows the functions of page 2 of the CC Engineering tab.

Table 51 - CC Engineering Tab Page 2 Description

Function	Action	Security	Configuration Parameters
Maximum/ Minimum EU: PV CV1 CV2 CV3	Type the maximum or minimum scaled value for PV. Type the maximum and minimum value of CV1EU, CV2EU, or CV3EU. This is the value of CV1EU, CV2EU, or CV3EU that corresponds to 100% or 0% of CV1, CV2, or CV3.	Engineering Configuration (Code E)	• .PVEUMax • .CV2EUMax • .CV2EUMax • .CV2EUMax • .PVEUMin • .CV2EUMin • .CV2EUMin • .CV2EUMin
Override Value CV1 CV2 CV3	Type the CV1, CV2, or CV3 Override value. CV1, CV2, or CV3 is set to this value when in Override mode.		• .CV1OverrideValue • .CV2OverrideValue • .CV3OverrideValue
Timing execution mode	Click to select the time base execution mode.		.TimingMode

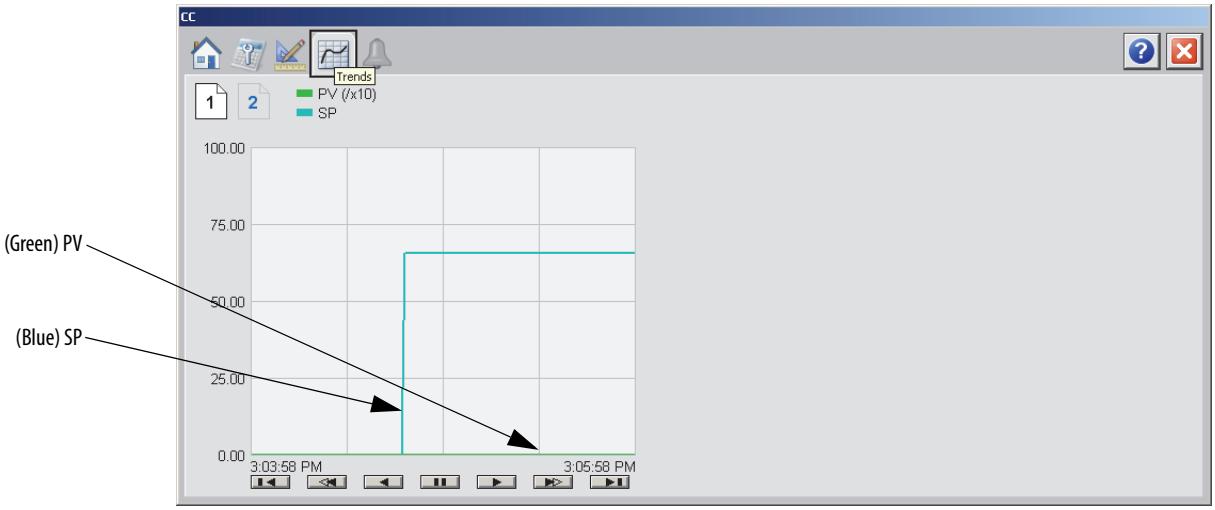
Trends Tab

The Trends tab shows trend charts of key device data over time. These faceplate trends provide a quick view of current device performance to supplement, but not replace, dedicated historical or live trend displays.

The Trends tab is divided into two pages.

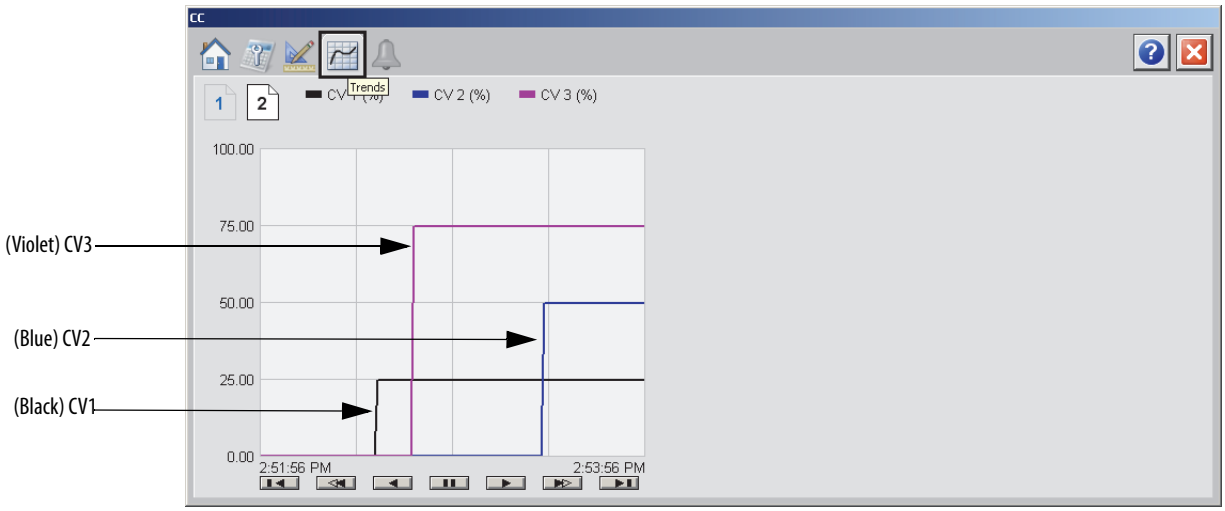
Trends Tab Page 1

Page 1 of the CC Trends tab trends values of PV and SP over time.



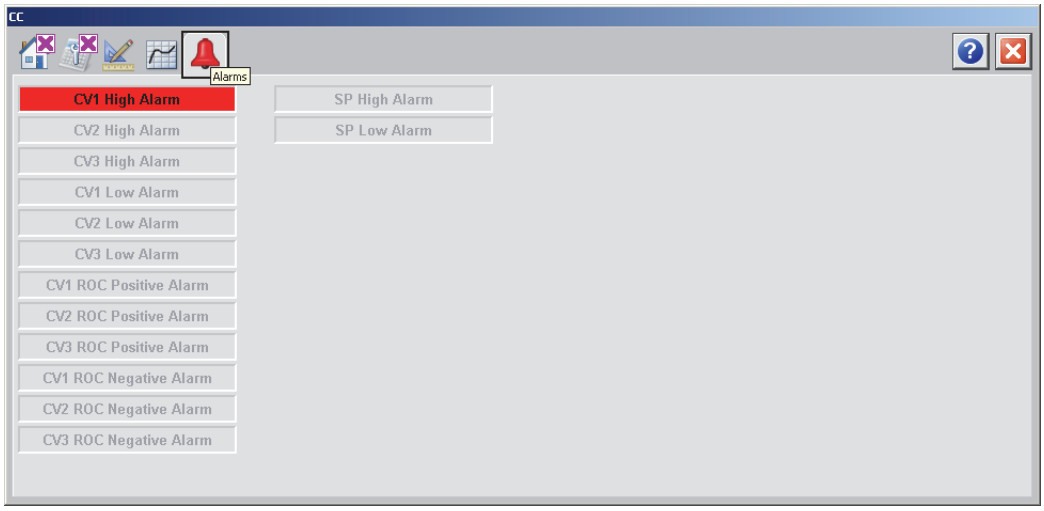
Trends Tab Page 2

Page 2 of the CC Trends tab trends values of CV1, CV2, and CV3 over time.



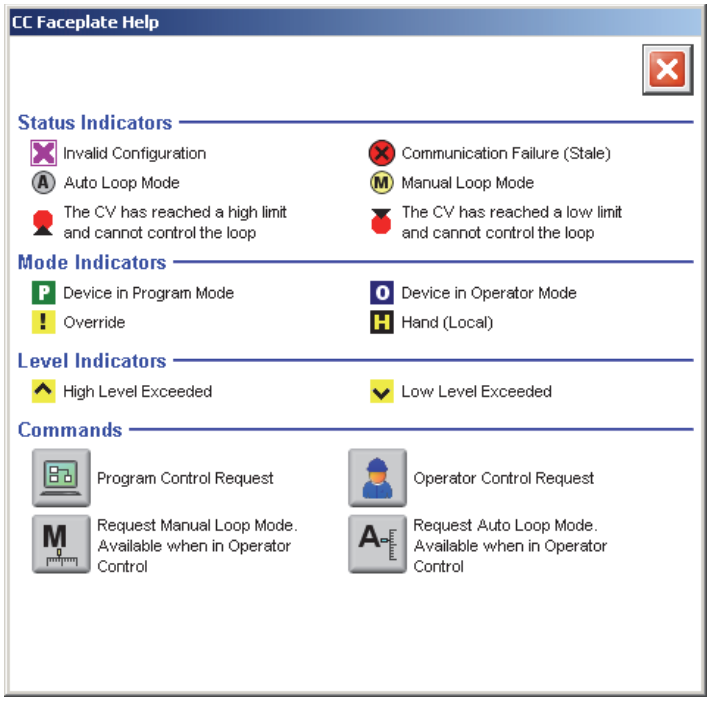
Alarms Tab

The CC Alarms tab shows all the available alarms for the device and their current status.



Faceplate Help

The Faceplate Help page shows the indicators and command buttons that are used by the Coordinated Control (CC) faceplate.



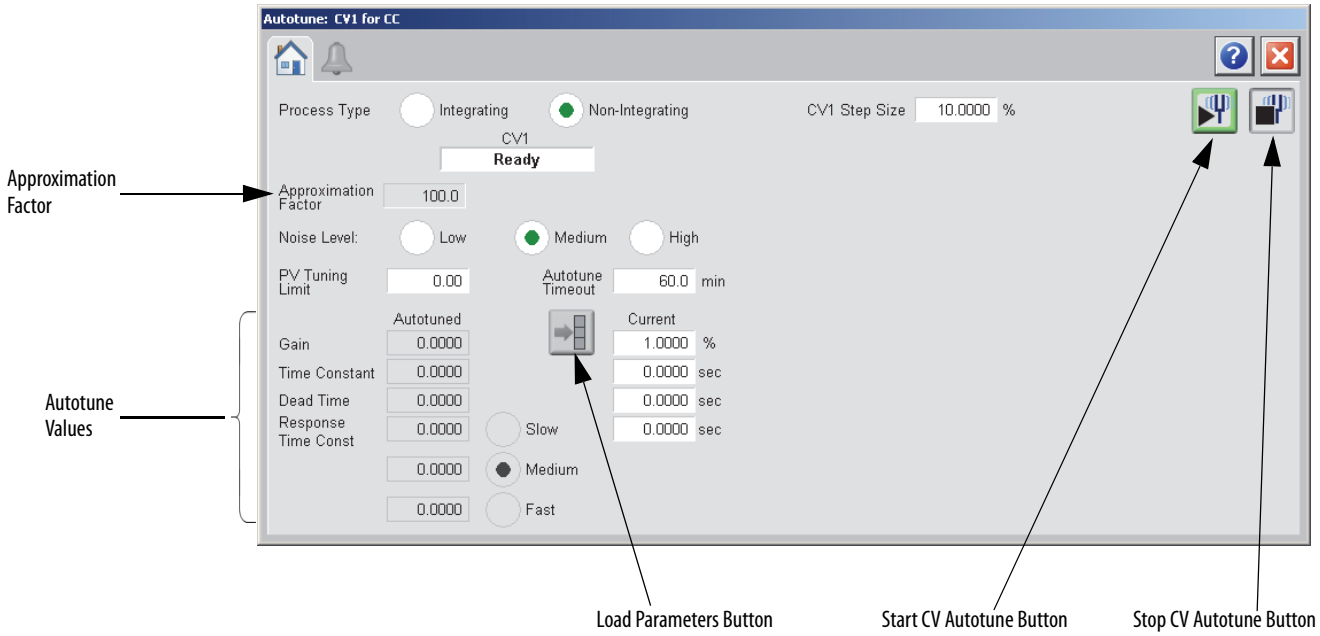
Coordinated Control (CC) Autotune

The faceplates in this section let you access all of the necessary parameters to autotune the CC function block as well as hand-tune the instruction.

CC Autotune Operator Tab

The CC Autotune Operator tab shows the following information:

- Approximation factor
- Autotuned gain, time constant, and dead time.
- Calculated value of CV time constant for slow response, medium response, and fast response speeds
- Load selected parameters into CC configuration parameters button
- Start CV autotune button
- Stop CV autotune button



The following table lists the functions on of the CC Autotune Operator tab.

Table 52 - CC Autotune Operator Tab Description

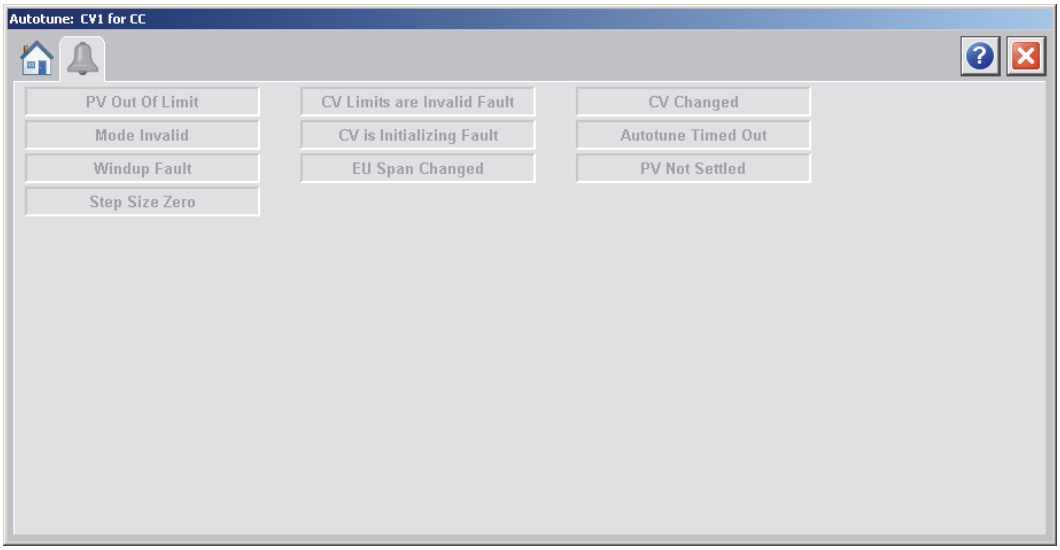
Function	Action	Security
Process Type	Click on either Integrating or Non-inegrating.	Configuration and Tuning Maintenance (Code D)
Approximation Factor	Type a value for the integrating model approximation factor. **IMPORTANT:** You can enter this value only when Integrating is selected as the Process Type.	
CV Step Size (CV1, CV2, or CV3)	Type a value for CV1, CV2, or CV3 step size in percent for the tuning step test.	
Noise Level	Click on Low, Medium, or High to set the estimate of the noise level expected on the PV to compensate for it during tuning.	
PV Tuning Limit	Type a value for the PV tuning limit scaled in PV units. When Autotune is running and predicted PV exceeds this limit, the tuning is aborted.	
Autotune Timeout	Type a value for the maximum time for autotune to complete following the CV step change. When autotune exceeds this time, tuning is aborted.	
Current Gain	Type a value for the internal model gain for CV1, CV2, or CV3. Enter a positive or negative gain depending on process direction.	
Current Time Constant	Type a value for CV1, CV2, or CV3 internal model time constant in seconds.	
Current Dead Time	Type a value for CV1, CV2, or CV3 internal model deadtime in seconds.	
Current Response Time Constant	Type a value for the tuning parameter that determines the speed of the control variable action for CV1, CV2, or CV3 in seconds	

Table 52 - CC Autotune Operator Tab Description

Function	Action	Security
	Click to replace the current model parameters with the calculated Autotune model parameters.	Configuration and Tuning Maintenance (Code D)
	Click to start the autotune process for CV1, CV2, or CV3.	
	Click to abort the autotune process. This button also becomes active if the process is aborted due to an error.	

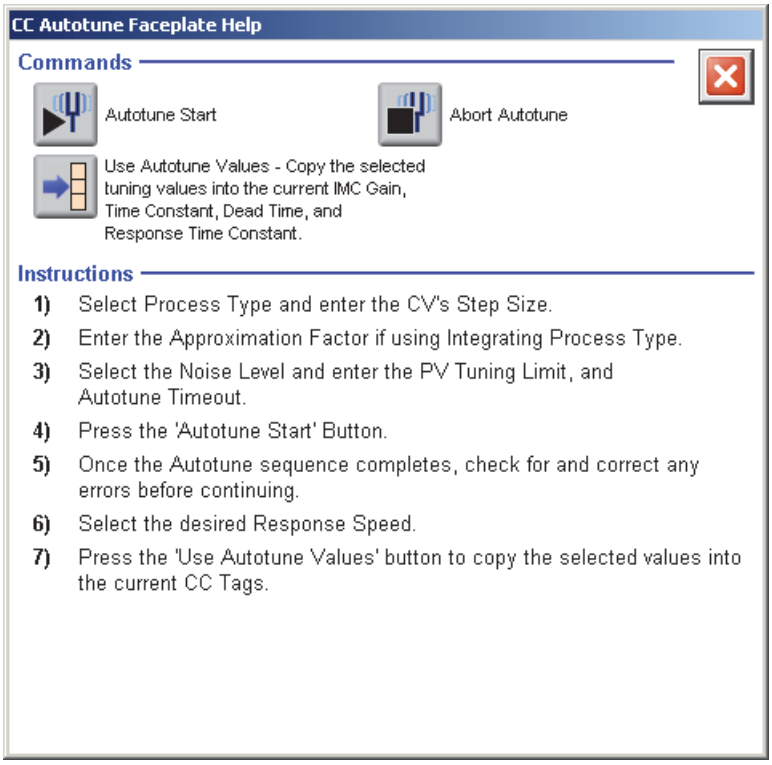
CC Autotune Alarms Tab

The Autotune Alarms tab shows all the available alarms for the device and their current status.



CC Autotune Faceplate Help

The Faceplate Help page shows the indicators and command buttons that are used by CC Autotune.



Internal Model Control (IMC)

The Internal Model Control (IMC) function block controls a single process variable by manipulating a single control-variable output. This function block performs an algorithm where the actual error signal is compared against that of an internal first-order lag plus deadtime model of the process. The IMC function block calculates the control variable output (CV) in the Auto mode based on the PV - SP deviation, internal model, and tuning.

Visualization Files

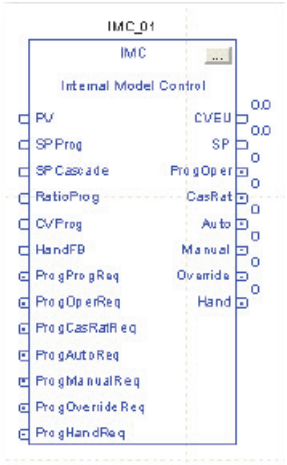
The following files are required to use the IMC Object and can be downloaded from the Product Compatibility and Download Center at http://www.rockwellautomation.com/rockwellautomation/support/pcdc.page.

IMPORTANT Files must be imported in the following order: image files, then global object files, and then graphic files. This order is required to properly configure the visualization files.

Table 53 - Internal Model Control (IMC) Visualization File Types

Application Type	File Type	FactoryTalk View SE Software	FactoryTalk View ME Software	Description
Graphics - Displays	GFX	(RA-BAS) Common Analog-Edit	N/A	Faceplate used for analog input data entry. The FactoryTalk View ME faceplates use the native analog input data entry so no file is required.
		(RA-BAS) Built-In IMC Autotune-Faceplate	(RA-BAS-ME) Built-In IMC Autotune-Faceplate	The faceplate display used for the Autotune object.
		(RA-BAS) Buit-In IMC Autotune-Help	(RA-BAS-ME) Buit-In IMC Autotune-Help	Help information that is accessed from the IMC Autotune Help faceplate.
		(RA-BAS) Built-In IMC Faceplate	(RA-BAS-ME) Built-In IMC Faceplate	The faceplate display used for the IMC object.
		(RA-BAS) Built-In IMC Help	(RA-BAS-ME) Built-In IMC Help	Help information that is accessed from the IMC Help faceplate.
		(RA-BAS) Built-In IMC Quick	(RA-BAS-ME) Built-In IMC Quick	The Quick display used for the IMC object.
Graphics - Global Objects	GGFX	(RA-BAS) Common Faceplate Objects	(RA-BAS-ME) Common Faceplate Objects	Common global objects used on all Process Object faceplates.
		(RA-BAS) BuiltIn Faceplate Objects	(RA-BAS-ME) BuiltIn Faceplate Objects	Global objects used on IMC faceplates.
		(RA-BAS) BuiltIn Graphics Librarys	(RA-BAS-ME) BuiltIn Graphics Librarys	Builtin display elements used to build process graphics.
		(RA-BAS) BuiltIn Help Objects	(RA-BAS-ME) BuiltIn Help Objects	BuiltIn global objects used for all BuiltIn help displays.
Graphics - Images	BMP	All .png files in the images folder	All .png files in the images folder	These are the common icons used in the global objects and faceplates for all Process Objects.
HMI Tags	CSV	N/A	FTVME_PlantPAxLib_Tags_3_1_**00**.csv(1)	These tags must be imported into the FactoryTalk View ME project to support switching tabs on any Process Object faceplate.

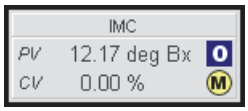
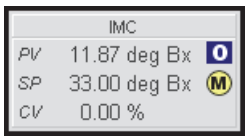
(1) The service release number (boldfaced) can change as service revisions are created.



Display Elements

A display element (global object) is created once and can be referenced multiple times on multiple displays in an application. When changes are made to the original (base) object, the instantiated copies (reference objects) are automatically updated. Use of global objects, in conjunction with tag structures in the ControlLogix system, aid consistency and save engineering time.

play Elements Descriptions

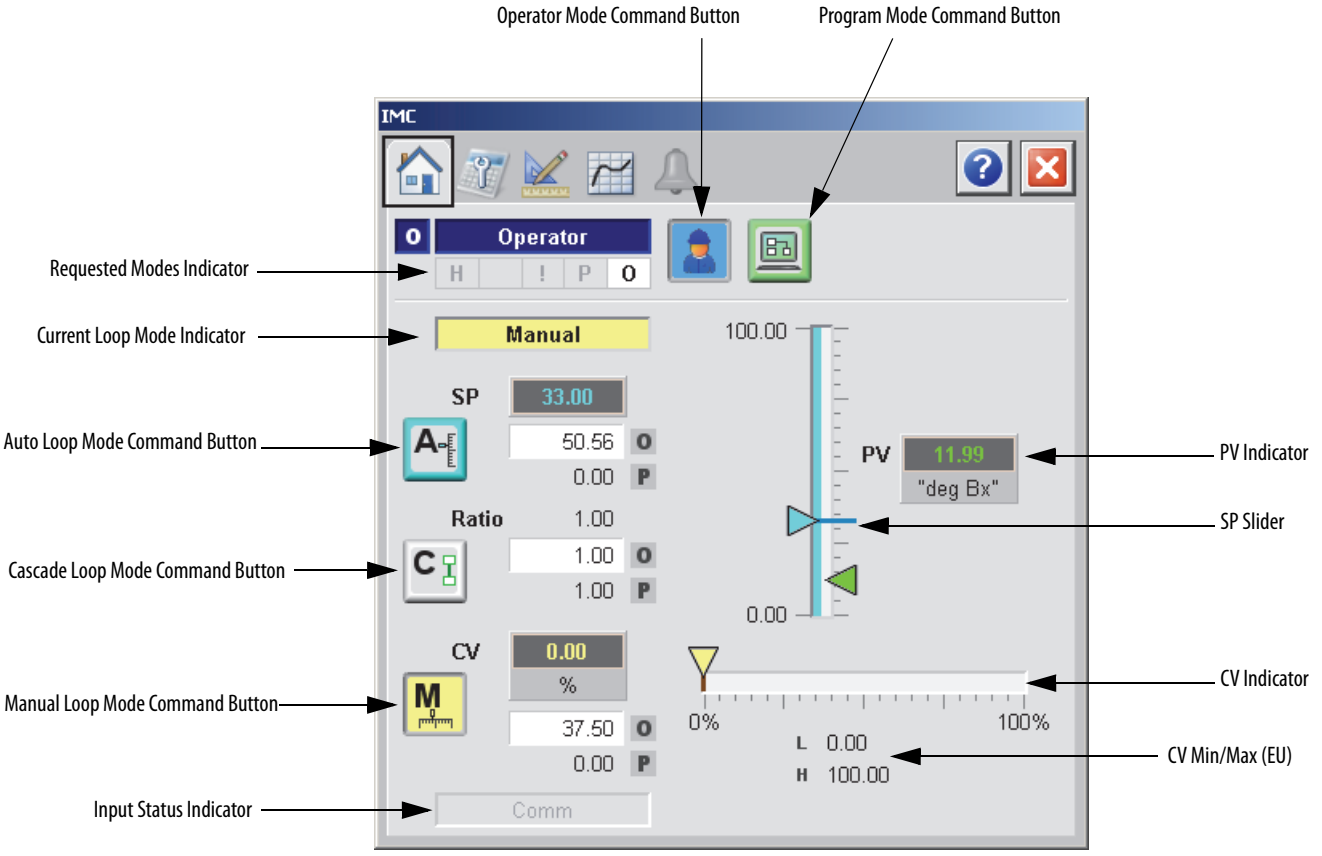
	Display Element	Description
		Internal Model Control object with a Process Variable and a Control Variable.
		Internal Model Control object with a Process Variable, Setpoint, and a Control Variable.

Operator Tab

The IMC faceplate initially opens to the Operator ('Home') tab. From here, an operator can monitor the device status and manually operate the device when it is in Operator mode.

The IMC Operator tab shows the following information:

- Requested modes indicator
- Current ratio multiplier
- Current Program multiplier
- Current Process Variable and bar graph
- Current Control Variable and bar graph
- Current Set Point
- High (H) and Low (L) clamping limits for the PV
- Input Status (Communications OK, Communications Fail, Bad PV Quality, or Uncertain PV Quality)



The following table lists the functions on of the Operator tab.

Table 55 - IMC Operator Tab Description

Function	Action	Security
	Click to request Operator mode.	Manual Device Operation (Code B)
	Click to request Program mode.	
	Click to request Auto Loop mode.	Normal Operation of Devices (Code A)
Operator Setpoint Value	Type the SP Operator value, scaled in PV units. SP is set to this value when in Operator control.	
	Click to request Cascade Loop mode. **IMPORTANT:** This button is available only if 'Allow Cascade/ Ratio Mode' on page 12 of the Engineering tab is checked. (See Engineering Tab Page 1 on page 193.)	
Operator CV value (CV1, CV2, and CV3)	Type a value for CV Operator Manual value. CV is set to this value when in Operator control and Manual mode.	

Table 55 - IMC Operator Tab Description

Function	Action	Security
	Click to request Manual Loop mode.	Normal Operation of Devices (Code A)
Operator Ratio Value	Type a value for the Ratio Operator multiplier. Ratio is set to this value when in Operator control.	
CV Slider (CV1, CV2, and CV3)	Move this slider to adjust the loop CV output.	Equipment Maintenance (Code C)
SP Slider	Move this slider to adjust the loop setpoint.	

Maintenance Tab

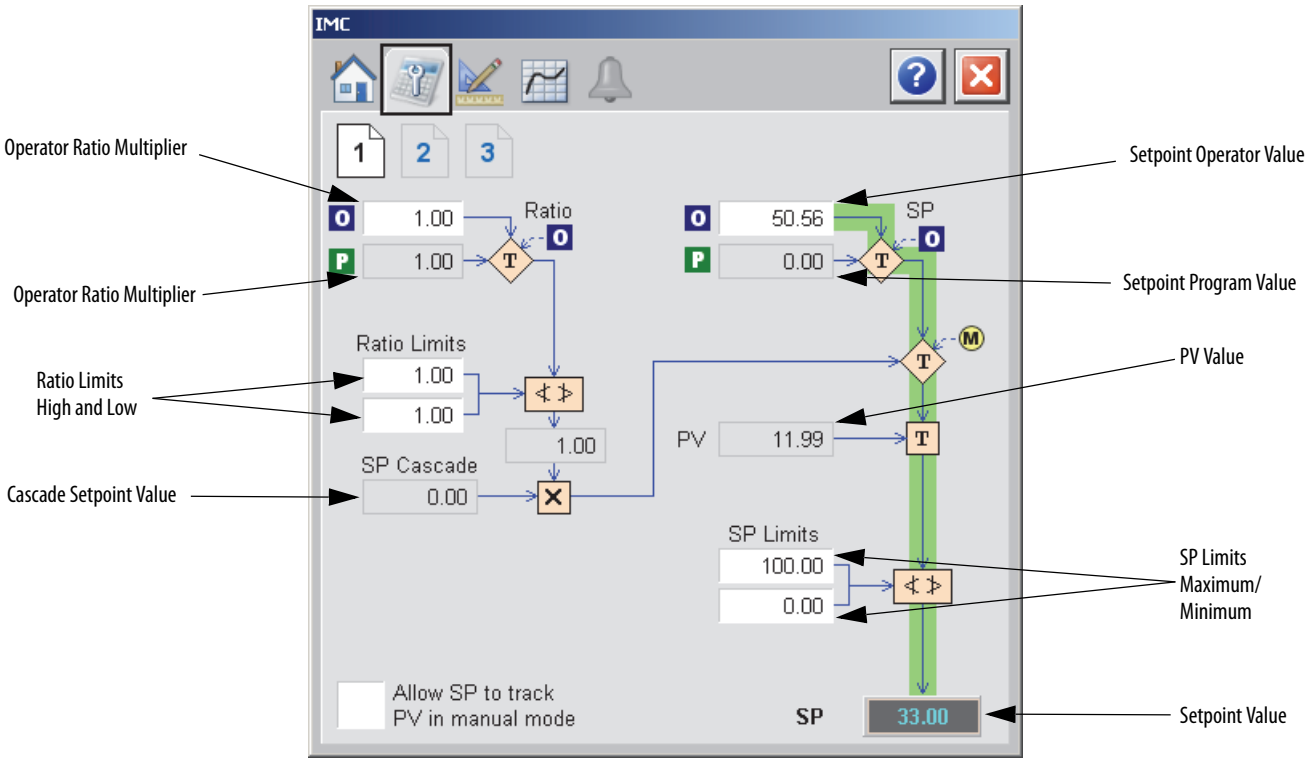
Maintenance personnel use the information and controls on the IMC Maintenance tab to make adjustments to device parameters, troubleshoot and temporarily work around device problems, and disable the device for routine maintenance.

The IMC Maintenance tab is divided into three tabs.

Maintenance Tab Page 1

Page 1 of the IMC Maintenance tab shows the following information:

- The ratio Program multiplier
- The Cascade setpoint value in PV units
- The setpoint Program value in PV units
- The PV value
- The setpoint value in PV units



The following table shows the functions of page 1 of the IMC Maintenance tab.

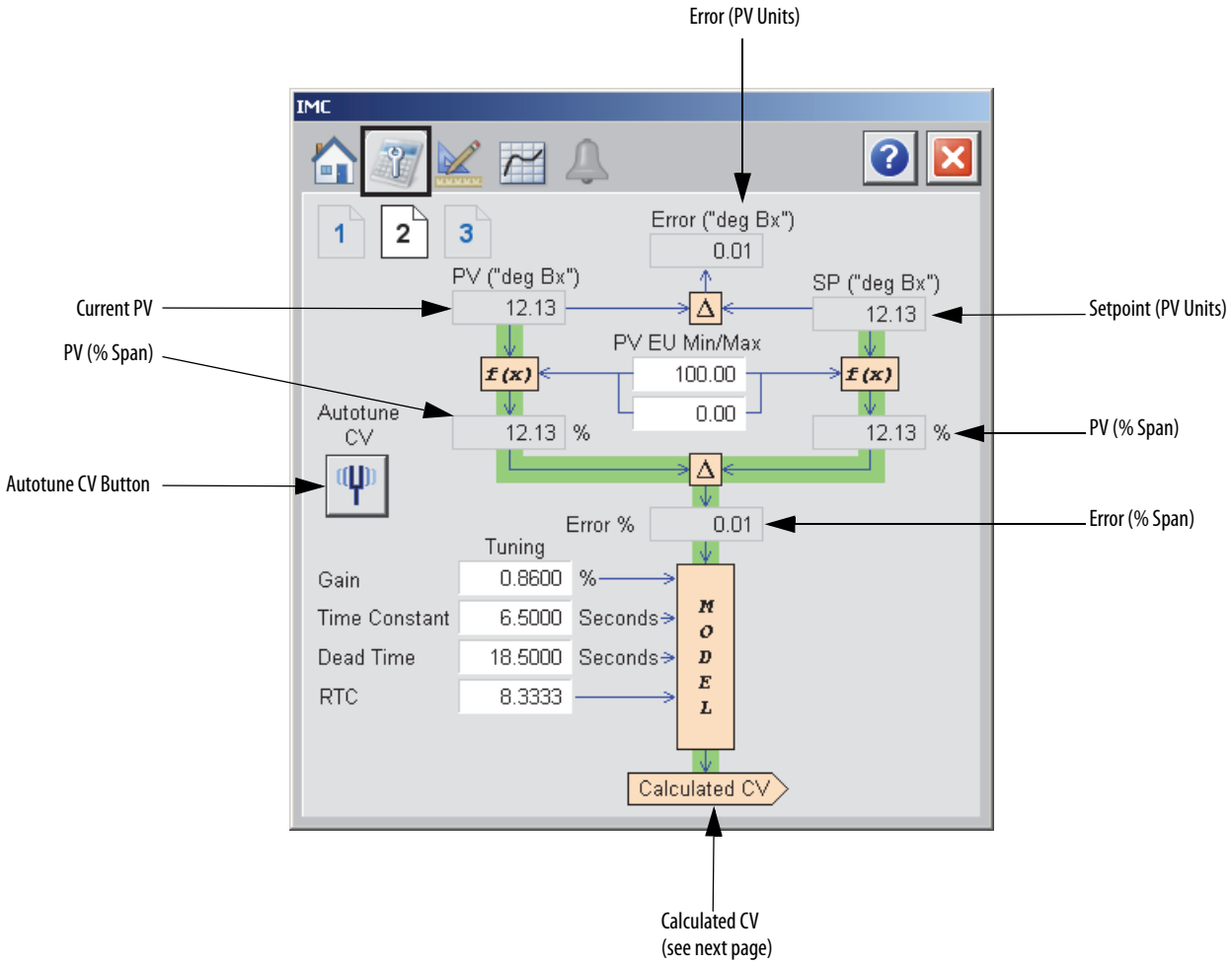
Table 56 - IMC Maintenance Tab Page 1 Description

Function	Action	Security	Configuration Parameters
Operator Ratio Multiplier	Type a value for the Ratio Operator multiplier. Ratio is set to this value when in Operator control.	Normal Operation of Devices (Code A)	.RatioOper
Ratio Limits High and Low	Type a value for the Ratio high and low limits. These parameters limit the value of Ratio obtained from RatioProg or RatioOper.	Configuration and Tuning Maintenance (Code D)	.RatioHLimit .RatioLLimit
Operator Setpoint	Type a value for the SP Operator value, scaled in PV units. SP is set to this value when in Operator control.	Normal Operation of Devices (Code A)	.SPOper
SP Limits	Type the SP maximum and minimum limits.	Configuration and Tuning Maintenance (Code D)	.SPHLimit .SPLLimit
Allow SP to track PV in manual mode	Click to enable CV Tracking when autotune is OFF. This parameter is ignored in Hand and Override mode.	Equipment Maintenance (Code C)	.PVTracking

Maintenance Tab Page 2

Page 2 of the Maintenance tab shows the following information:

- Error value in PV units
- Current process variable (PV)
- Setpoint value in PV units
- PV (percent of span)
- PV(percent of span)
- Error (percent of span)



The following table shows the functions of page 2 of the IMC Maintenance tab.

Table 57 - IMC Maintenance Tab Page 2 Description

Function	Action	Security	Configuration Parameters
PV EU Min/Max	Type the minimum or maximum limits for the PV in engineering units.	Engineering Configuration (Code E)	.PVEUMax .PVEUMin

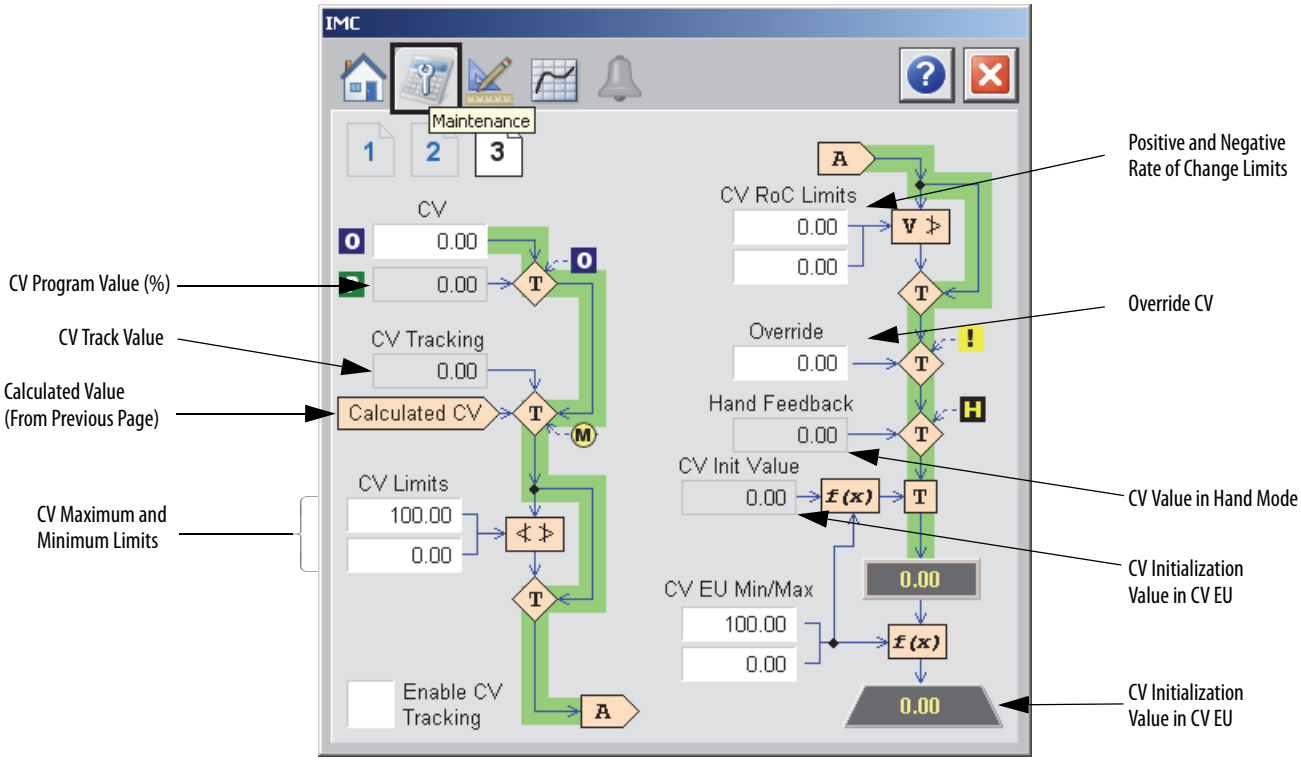
Table 57 - IMC Maintenance Tab Page 2 Description

Function	Action	Security	Configuration Parameters
Gain	Type a value for the internal model gain. Enter a positive or negative gain depending on process direction.	Configuration and Tuning Maintenance (Code D)	.ModelGain
Time Constant	Type a value for the internal model Time Constant in seconds.		.ModelTC
Dead Time	Type a value for the internal model Deadtime in seconds.		.ModelDT
RTC	Type a value for the tuning parameter that determines the speed of the control variable action in seconds.		.RespTC
	Click to open the CV Autotune faceplate.	None	None

Maintenance Tab Page 3

Page 3 of the IMC Maintenance tab shows the following information:

- Calculated CV (from Maintenance page 2)
- CV Program value in percent
- CV track value
- CV value in Hand mode
- CV initialization value in CV engineering units
- CV in engineering units



The following table shows the functions of page 3 of the IMC Maintenance tab.

Table 58 - IMC Maintenance Tab Page 3 Description

Function	Action	Security	Configuration Parameters
Operator CV Value (%)	CV Operator-Manual value in percent. CV is set to this value when in Operator control and Manual mode.	Normal Operation of Devices (Code A)	.CVOper
CV High Limit CV Low Limit	Type values for the CV high and low limits. It is also used for limiting CV when in Auto or CascadeRatio modes or Manual mode if CVManLimiting is **true**.	Configuration and Tuning Maintenance (Code D)	• .CVHLimit • .CVLLimit
CV RoC Positive Limit CV RoC Negative Limit	CV increasing or decreasing rate of change limit, in percent per second. The rate of change limiting is used only when in Auto or CascadeRatio modes or Manual mode if CVManLimiting is **true**. A value of zero disables CV ROC limiting.		• .CVROCPosLimit • .CVROCNegLimit
Override	Type the value for the CV Override value. CV is set to this value when in Override mode.		.CVOverrideValue
CV EU Maximum CV EU Minimum	Type the maximum or minimum value for CVEU scaling. This the value of CVEU that corresponds to 100% or 0% CV respectively.	Engineering Configuration (Code E)	• .CVEUMax • .CVEUMin
Enable CV Tracking	Check to enable CV Tracking when autotune is OFF. This parameter is ignored in Hand and Override mode.	Equipment Maintenance (Code C)	.CVTrackReq

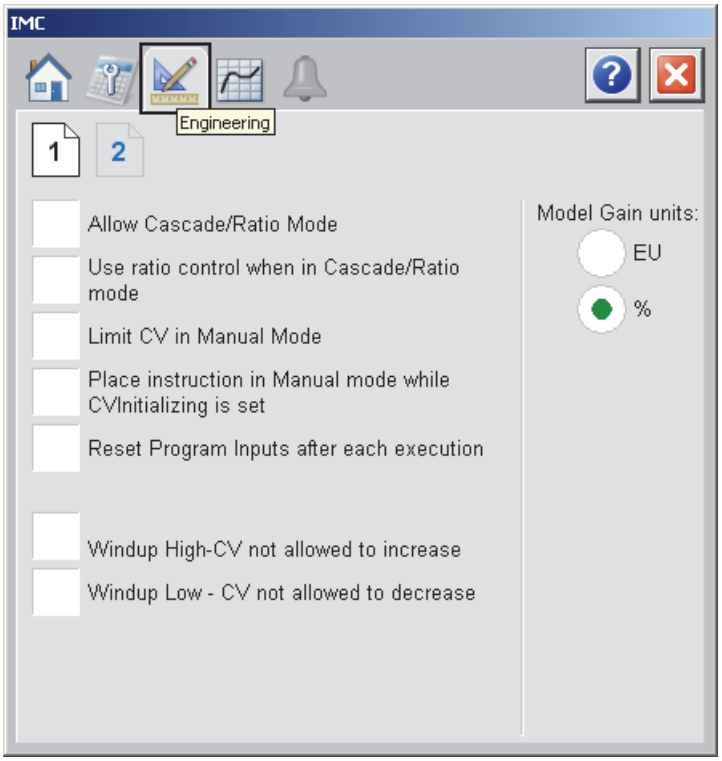
Engineering Tab

The Engineering tab provides access to device configuration parameters and ranges, options for device and I/O setup, displayed text, and faceplate-to-faceplate navigation settings, for initial system commissioning or later system changes.

The Engineering tab is divided into two pages.

Engineering Tab Page 1

Page 1 of the Engineering tab has various Operator inputs/options for the CV, Cascade/Ratio mode, and Windup CV.



The following table shows the functions of page 1 of the IMC Engineering tab.

Table 59 - IMC Engineering Tab Page 1 Description

Function	Action	Security	Configuration Parameters
Allow Cascade/ Ratio Mode	Check if Cascade/Ratio mode is to be used. **IMPORTANT:** Checking this option displays the Cascade button on the Operator tab. (See Operator Tab on page 187.)	Engineering Configuration (Code E)	.AllowCasRat

Table 59 - IMC Engineering Tab Page 1 Description

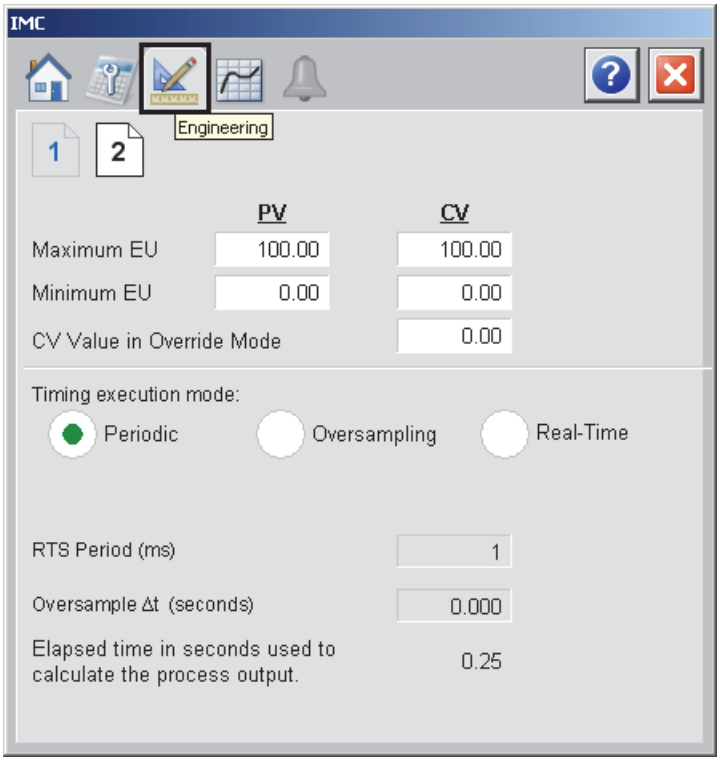
Function	Action	Security	Configuration Parameters
Use ratio control when in Cascade/ Ratio mode	Check to enable ratio control when in CascadeRatio mode. **IMPORTANT:** Checking this option displays the current ratio multiplier, Operator ratio multiplier, and Program ratio multiplier on the Operator tab. (See Operator Tab on page 187.) Checking this option also displays the Cascade/Ratio portion of the SAMA diagram on page 1of the Maintenance tab. (See Maintenance Tab Page 1 on page 189.)	Engineering Configuration (Code E)	.UseRatio
Limit CV in manual Mode	Check to limit the CV while in Manual mode.		.CVManLimiting
Place instruction in Manual mode while CVInitializing is set	Check to set the Loop mode to manual when CV initialization is requested. Clear the checkbox to leave the Loop mode unchanged when initialization is requested. When the initialization request clears, the loop resumes control in its previous Loop mode.		.ManualAfterInit
Reset Program Inputs after each execution	Check to reset Program inputs after each execution.		.ProgValueReset
Model Gain units	Select either 'EU' or '%' for the Model Gain units.		.GainEUSpan
Windup High - CV not allowed to increase	Check so that CV cannot increase in value.		.WindupHIn
Windup Low - CV not allowed to decrease	Check so that CV cannot decrease in value.		.WindupLIn

Engineering Tab Page 2

Page 2 of the Engineering tab has various Operator inputs/options for the PV, CV, and Timing Execution mode.

The following values are displayed:

- RTS period
- Oversample delta-t
- Time used to calculate output.



The following table shows the functions of page 2 of the IMC Engineering tab.

Table 60 - IMC Engineering Tab Page 2 Description

Function	Action	Security	Configuration Parameters
Maximum/ Minimum EU: PV CV	Type the maximum or minimum scaled value for PV. Type the maximum and minimum value of CV. This is the value of CVEU that corresponds to 100% or 0% of CV.	Engineering Configuration (Code E)	• .PVEUMax • .CVEUMax • .PVEUMin • .CVEUMin
CV Value in Override Mode	Type the CV Override value. CV is set to this value when in Override mode.		.CVOverrideValue
Timing execution mode	Click to select the time base execution mode.		.TimingMode

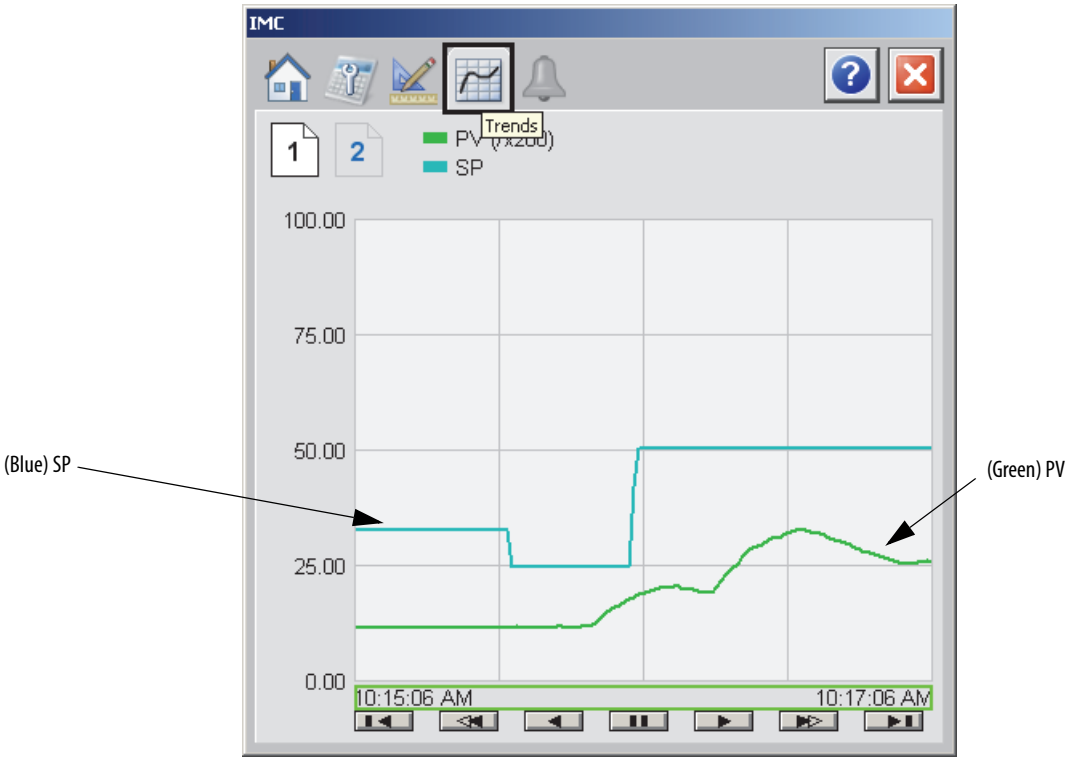
Trends Tab

The Trends tab shows trend charts of key device data over time. These faceplate trends provide a quick view of current device performance to supplement, but not replace, dedicated historical or live trend displays.

The Trends tab is divided into two pages.

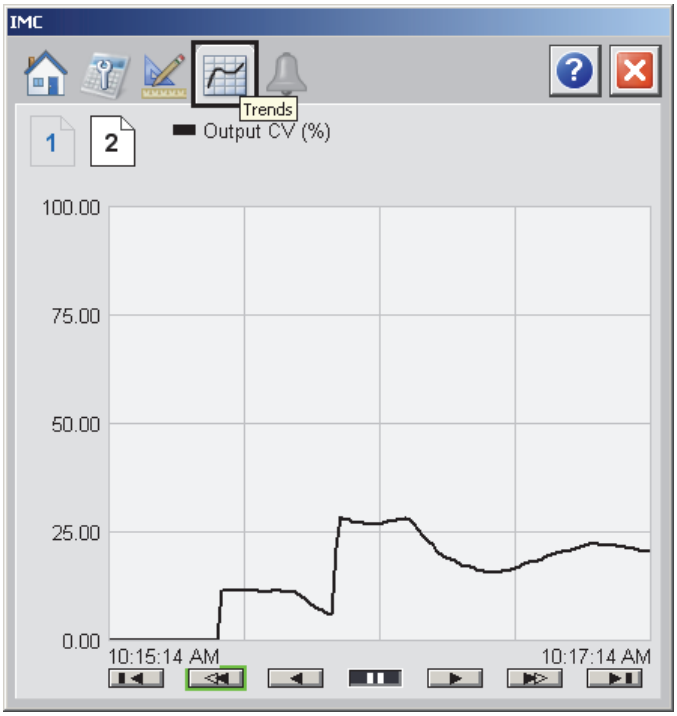
Trends Tab Page 1

Page 1 of the IMC Trends tab shows the relationship between PV and SP for the same time frame of a process.



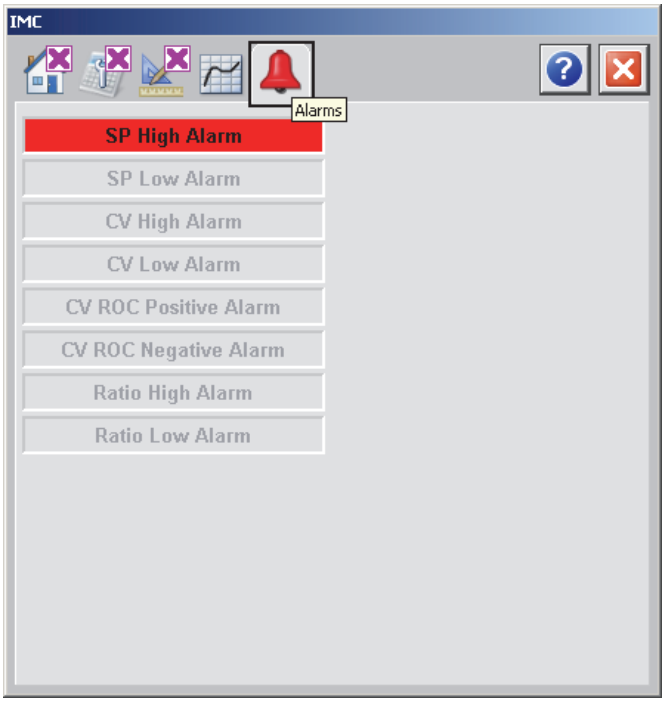
Trends Tab Page 2

Page 2 of the IMC Trends tab shows the output CV waveform.



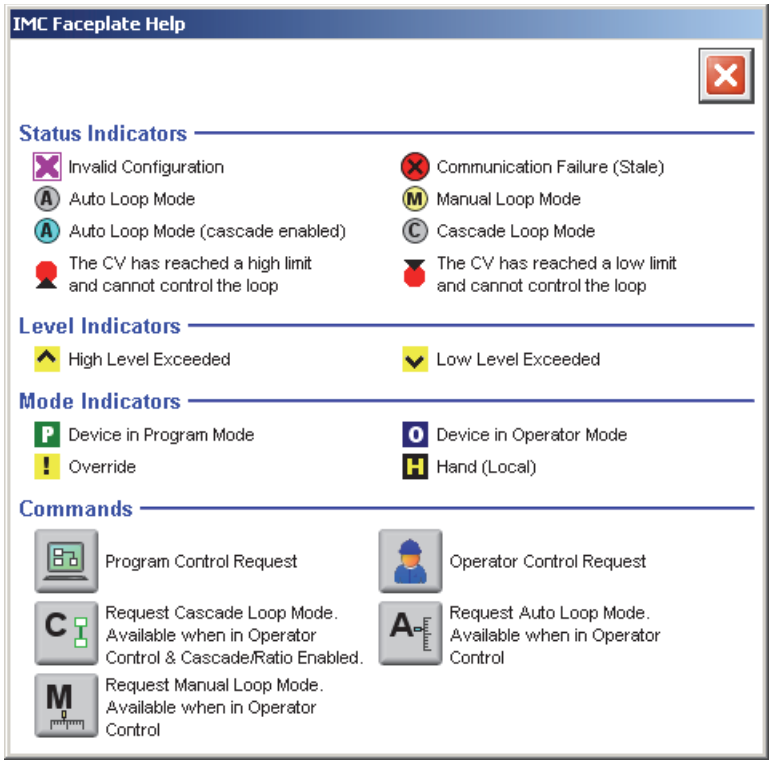
Alarms Tab

The IMC Alarms tab shows all the available alarms for the device.



Faceplate Help

The Faceplate Help page shows the indicators and command buttons that are used by the Internal Model Control (IMC) faceplate.



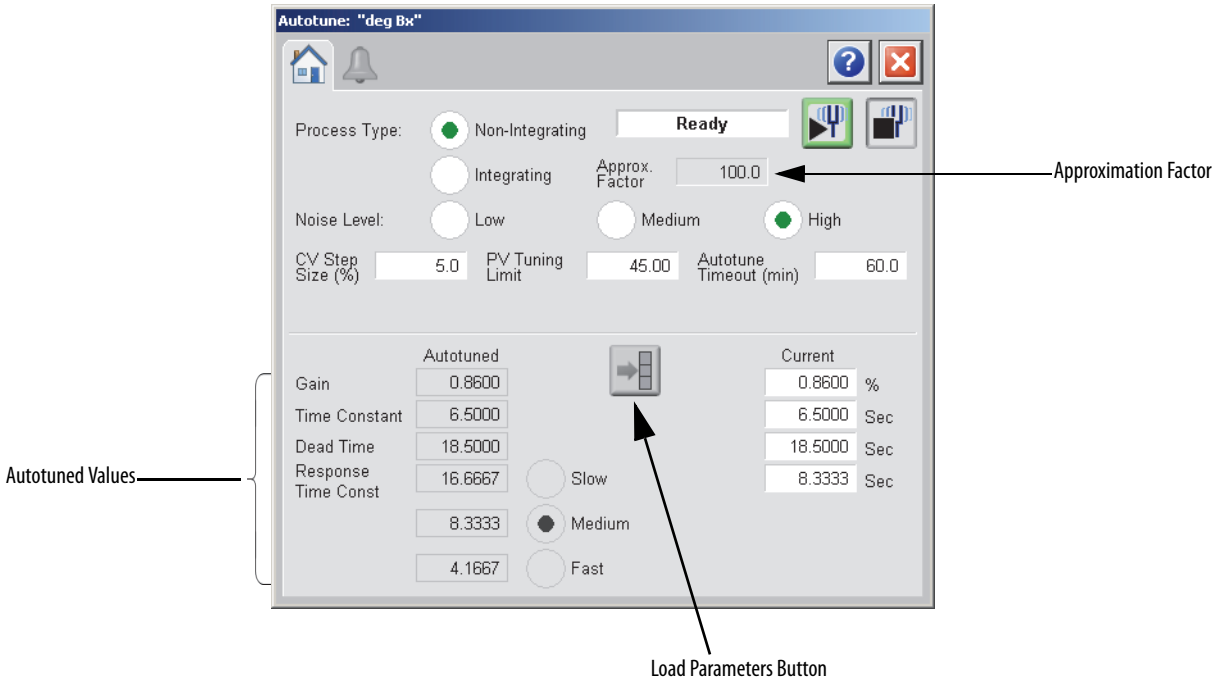
Internal Model Control (IMC) Autotune

The faceplates in this section let you access all of the necessary parameters to autotune the IMC function block as well as hand-tune the instruction.

IMC Autotune Operator Tab

The IMC Autotune Operator tab shows the following information:

- Approximation factor
- Autotuned gain, Time constant, and dead time.
- Calculated value of CV time constant for slow response, medium response, and fast response speeds
- Load selected parameters into CC configuration parameters button
- Start CV autotune button
- Stop CV autotune button



The following table lists the functions on of the IMC Autotune Operator tab.

Table 61 - IMC Autotune Operator Tab Description

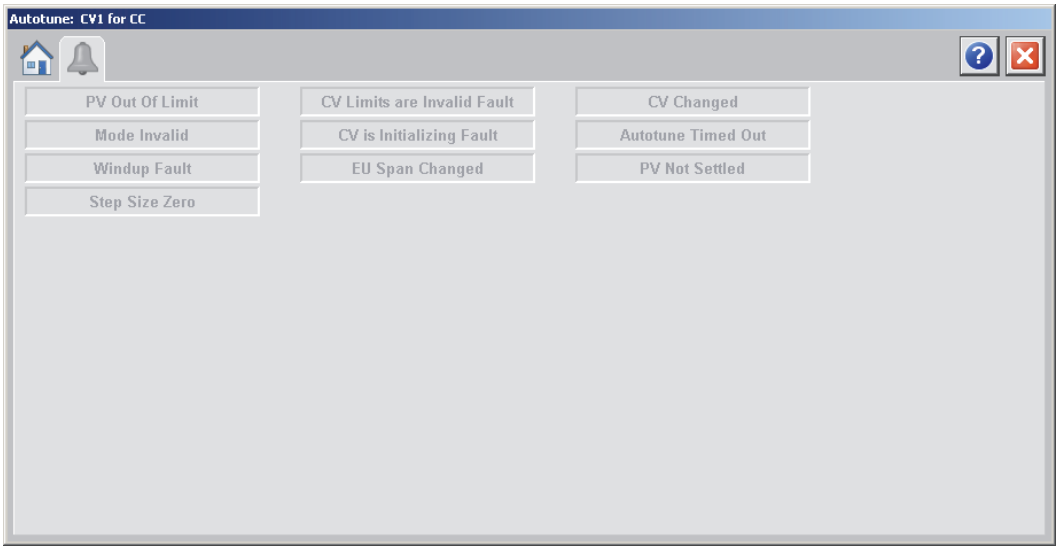
Function	Action	Security
Process Type	Click on either Integrating or Non-inegrating.	Configuration and Tuning Maintenance (Code D)
Approximation Factor	Type a value for the non-integrating model approximation factor. **IMPORTANT:** You can enter this value only when Integrating is selected as the Process Type.	
CV Step Size (%)	Type a value for CV1, CV2, or CV3 step size in percent for the tuning step test.	
Noise Level	Click on Low, Medium, or High to set the estimate of the noise level expected on the PV to compensate for it during tuning.	
PV Tuning Limit	Type a value for the PV tuning limit scaled in PV units. When Autotune is running and predicted PV exceeds this limit, the tuning is aborted.	
Autotune Timeout	Type a value for the maximum time for autotune to complete following the CV step change. When autotune exceeds this time, tuning is aborted.	
Currrent Gain	Type a value for the internal model gain for CV1, CV2, or CV3. Enter a positive or negative gain depending on process direction.	
Current Time Constant	Type a value for CV1, CV2, or CV3 internal model time constant in seconds.	
Current Dead Time	Type a value for CV1, CV2, or CV3 internal model deadtime in seconds.	
Current Response Time Constant	Type a value for the tuning parameter that determines the speed of the control variable action for CV1, CV2, or CV3 in seconds	

Table 61 - IMC Autotune Operator Tab Description

Function	Action	Security
	Click to replace the current model parameters with the calculated Autotune model parameters.	Configuration and Tuning Maintenance (Code D)
	Click to start the autotune process for CV1, CV2, or CV3.	
	Click to abort the autotune process. This button also becomes active if the process is aborted due to an error.	

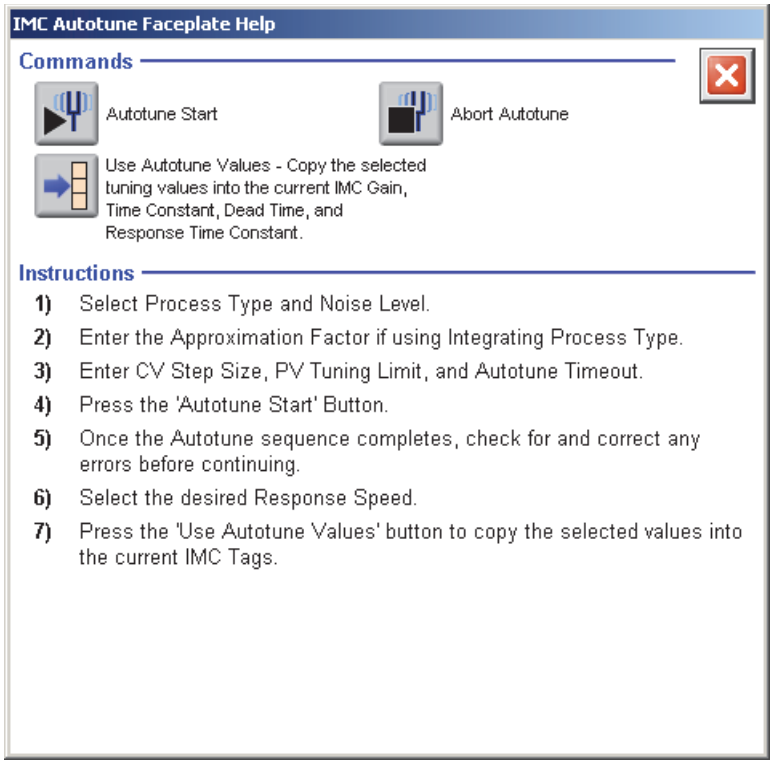
IMC Autotune Alarms Tab

The IMC Autotune Alarms tab shows all the available alarms for the device and their current status.



IMC Autotune Faceplate Help

The Faceplate Help page shows the command buttons that are used by IMC Autotune. The faceplate also provides basic instruction on how to use Autotune.



Modular Multivariable Control (MMC)

The Modular Multivariable Control (MMC) function block controls two process variables to their setpoints manipulating up to three control variables. The MMC function block calculates the control variables (CV1, CV2, and CV3) in the Auto mode based on the PV1 - SP1, PV2 - SP2 deviation, internal model, and tuning. The MMC function block is a flexible model-based algorithm that can be used in two basic configuration modes:

- Three control variables used to control two interacting process variables
- Two control variables used to control two interacting process variables

Visualization Files

The following files are required to use the IMC Object and can be downloaded from the Product Compatibility and Download Center at http://www.rockwellautomation.com/rockwellautomation/support/pcdc.page.

IMPORTANT Files must be imported in the following order: image files, then global object files, and then graphic files. This order is required to properly configure the visualization files.

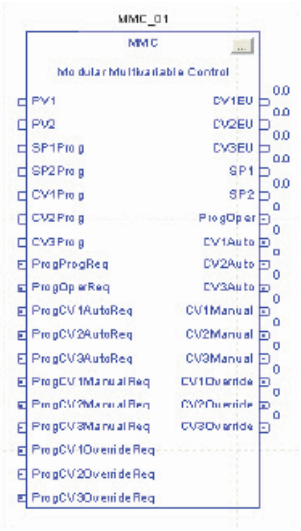
Table 62 - Modular Multivariable Control (MMC) Visualization File Types

Application Type	File Type	FactoryTalk View SE Software	FactoryTalk View ME Software	Description
Graphics - Displays	GFX	(RA-BAS) Common Analog-Edit	N/A	Faceplate used for analog input data entry. The FactoryTalk View ME faceplates use the native analog input data entry so no file is required.
		(RA-BAS) Built-In MMC Autotune-Faceplate	(RA-BAS-ME) Built-In MMC Autotune-Faceplate	The faceplate display used for the Autotune object.
		(RA-BAS) Buit-In MMC Autotune-Help	(RA-BAS-ME) Built-In MMC Autotune-Help	Help information that is accessed from the MMC Autotune Help faceplate.
		(RA-BAS) Built-In MMC Faceplate	(RA-BAS-ME) Built-In MMC Faceplate	The faceplate display used for the MMC object.
		(RA-BAS) Built-In MMC Help	(RA-BAS-ME) Built-In MMC Help	Help information that is accessed from the MMC Help faceplate.
		(RA-BAS) Built-In MMC Quick	(RA-BAS-ME) Built-In MMC Quick	The Quick display used for the MMC object.
Graphics - Global Objects	GGFX	(RA-BAS) Common Faceplate Objects	(RA-BAS-ME) Common Faceplate Objects	Common global objects used on all Process Object faceplates.
		(RA-BAS) BuiltIn Faceplate Objects	(RA-BAS-ME) BuiltIn Faceplate Objects	Global objects used on IMC faceplates.
		(RA-BAS) BuiltIn Graphics Librarys	(RA-BAS-ME) BuiltIn Graphics Librarys	Builtin display elements used to build process graphics.
		(RA-BAS) BuiltIn Help Objects	(RA-BAS-ME) BuiltIn Help Objects	BuiltIn global objects used for all BuiltIn help displays.
Graphics - Images	BMP	All .png files in the images folder	All .png files in the images folder	These are the common icons used in the global objects and faceplates for all Process Objects.
HMI Tags	CSV	N/A	FTVME_PlantPAxLib_Tags_3_1_**00**.csv(1)	These tags must be imported into the FactoryTalk View ME project to support switching tabs on any Process Object faceplate.

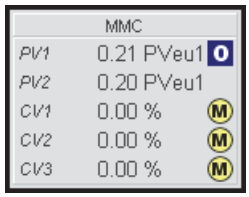
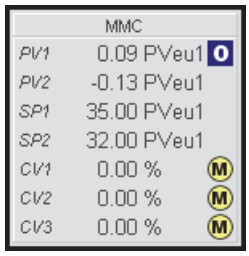
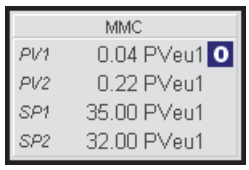
(1) The service release number (boldfaced) can change as service revisions are created.

Display Elements

A display element (global object) is created once and can be referenced multiple times on multiple displays in an application. When changes are made to the original (base) object, the instantiated copies (reference objects) are automatically updated. Use of global objects, in conjunction with tag structures in the ControlLogix system, aid consistency and save engineering time.



Ta **IMC) Display Elements Descriptions**

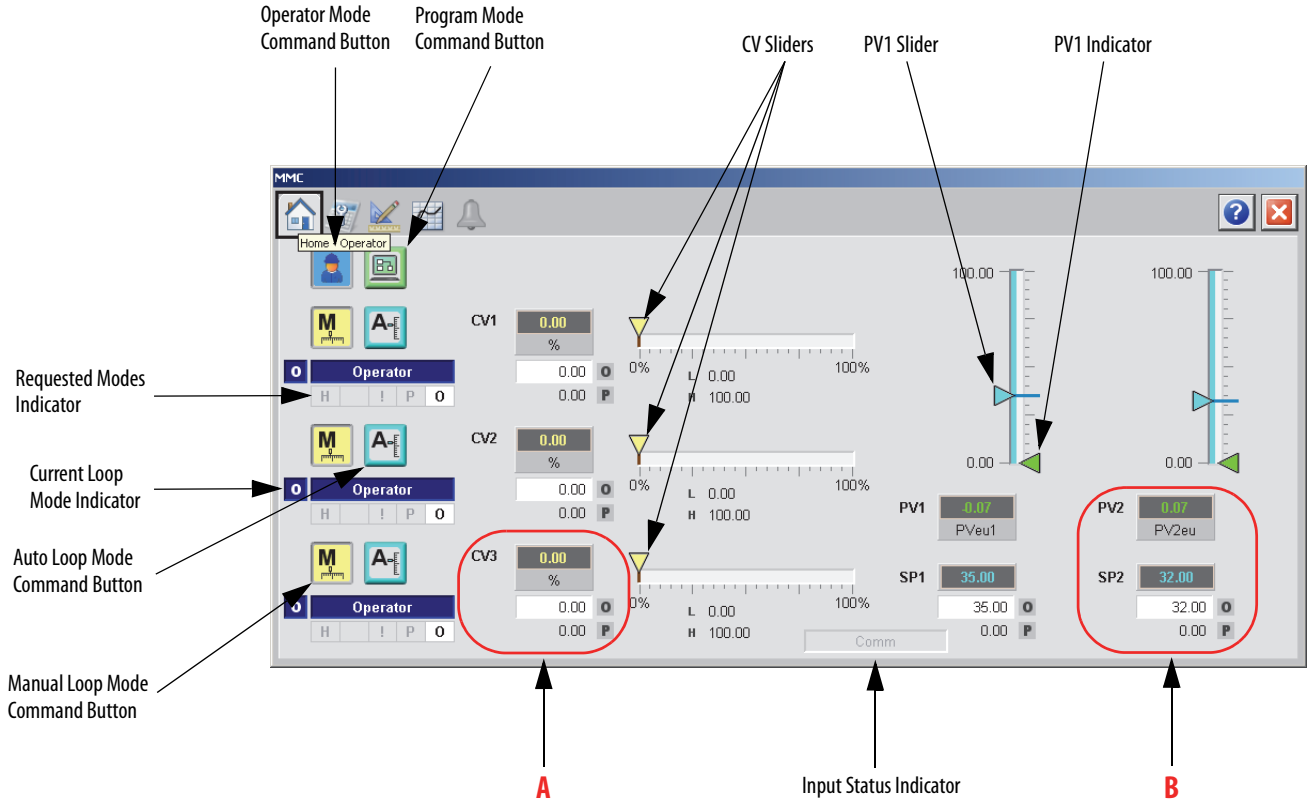
	Display Element	Description
		Modular Multivariable Control object with two process variables and three control variables.
		Modular Multivariable Control object with two process variables, two setpoints, and three control variables.
G0_BuiltIn_MMC2		Modular Multivariable Control object with two process variables and two setpoints.

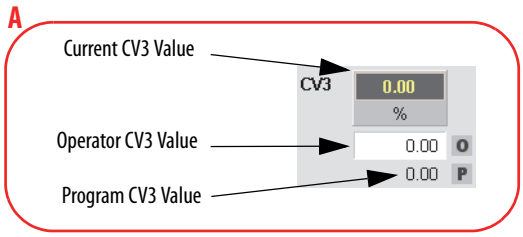
Operator Tab

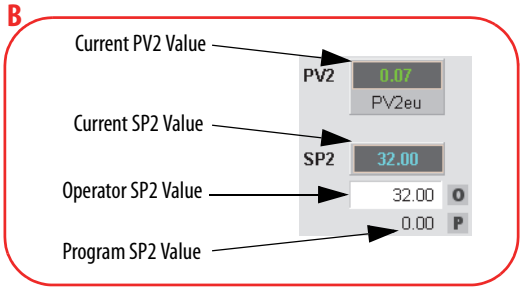
The faceplate initially opens to the Operator ('Home') tab. From here, an operator can monitor the device status and manually operate the device when it is in Operator mode.

The MMC Operator tab shows the following information:

- Requested modes indicator for CV1, CV2, and CV3
- Current mode indicator for CV1, CV2, and CV3
- Current process variable (PV1 and PV2) and bar graph
- Current CV (CV1, CV2, and CV3)
- Program CV (CV1, CV2, and CV3) and bar graph for each
- Current setpoint for PV1 and PV2
- Program setpoint for PV1 and PV2







The following table lists the functions on of the MMC Operator tab.

Table 64 - MMC Operator Tab Description

Function	Action	Security
	Click to request Operator mode.	Manual Device Operation (Code B)
	Click to request Program mode.	
	Click to request Auto Loop mode.	Normal Operation of Devices (Code A)

Table 64 - MMC Operator Tab Description

Function	Action	Security
	Click to request Manual Loop mode.	Normal Operation of Devices (Code A)
Operator CV Value (CV1, CV2, and CV3)	Type a value for CV1, CV2, or CV3.	
CV Slider (CV1, CV2, and CV3)	Move this slider to adjust the loop CV output.	Equipment Maintenance (Code C)
Operator Setpoint Value (SP1 and SP2	Type a value for the loop Setpoint.	Normal Operation of Devices (Code A)
PV Slider (PV1 and PV2)	Move this slider to adjust the loop PV value.	Equipment Maintenance (Code C)

Maintenance Tab

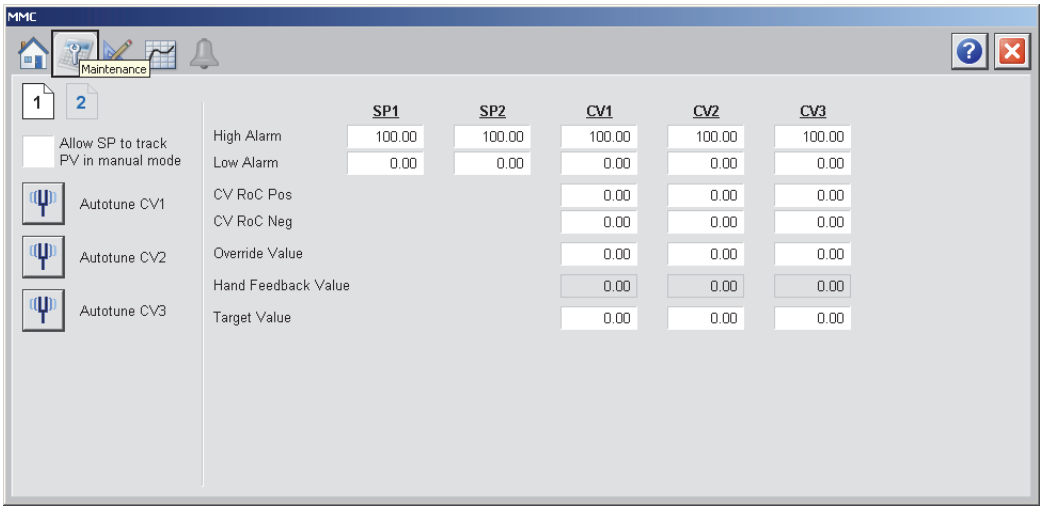
Maintenance personnel use the information and controls on the MMC Maintenance tab to make adjustments to device parameters, troubleshoot and temporarily work around device problems, and disable the device for routine maintenance.

The MMC Maintenance tab is divided into two tabs.

Maintenance Tab Page 1

Page 1 of the MMC Maintenance tab shows the following information:

- The Hand feedback value
- Autotune buttons for CV1, CV2, and CV3
- High and low alarms
- Operator inputs for high and low alarms, positive and negative rate of change, override, and target value



The following table shows the functions of page 1 of the Maintenance tab.

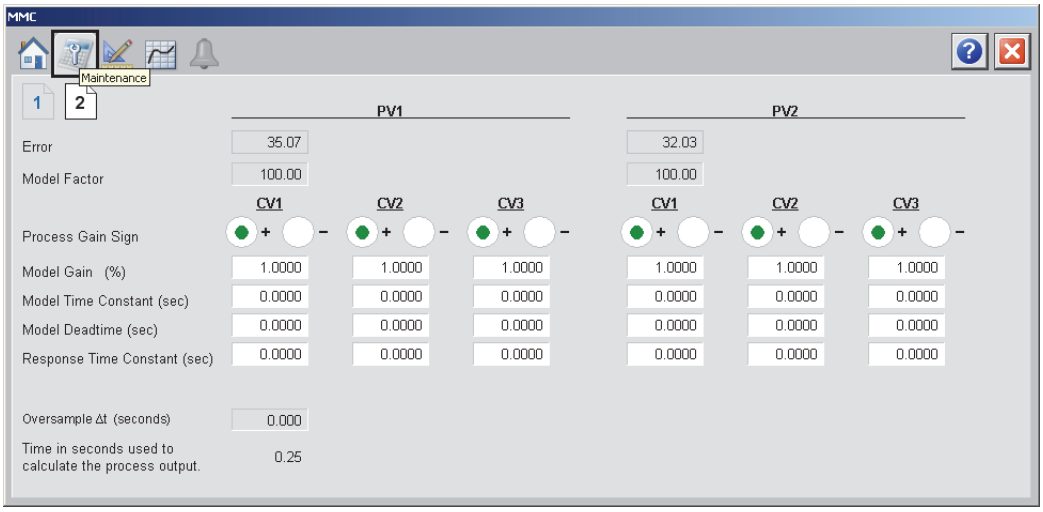
Table 65 - MMC Maintenance Tab Page 1 Description

Function	Action	Security	Configuration Parameters
Allow SP to track PV in manual mode	Check to enable SP to track PV. This is ignored when in Auto modes. SP tracks PV only when all three outputs are in manual. As soon as any output returns to Auto, PVTracking stops.	Equipment Maintenance (Code C)	.PVTracking
	Click to show the Autotune Operator faceplate.	None	None
Operator High Alarm	Type the High alarm value for SP1, SP2, CV1, CV2, or CV3.	Configuration and Tuning Maintenance (Code D)	• .SP1HLimit • .SP2HLimit • .CV1HLimit • .CV2HLimit • .CV3HLimit
Operator Low Alarm	Type the Low alarm limit value for SP1, SP2, CV1, CV2, or CV3.		• .SP1LLimit • .SP2LLimit • .CV1LLimit • .CV2LLimit • .CV3LLimit
CV RoC Pos	Type the positive Rate of Change limit value for CV1, CV2, or CV3. Rate of change limiting is used only when in Auto mode or in Manual mode if CVManLimiting is **true**. A value of zero disables CV1 ROC limiting.		• .CV1ROCPosLimit • .CV2ROCPosLimit • .CV3ROCPosLimit
CV RoC Neg	Type the negative Rate of Change limit value for CV1, CV2, or CV3. Rate of change limiting is used only when in Auto mode or in Manual mode if CVManLimiting is **true**. A value of zero disables CV2 ROC limiting.		• .CV1ROCNegLimit • .CV2ROCNegLimit • .CV3ROCNegLimit
Override Value	Type the Override value for CV1, CV2, or CV3. CV1, CV2, or CV3 is set to this value when in the Override mode.		• .CV1OverrideValue • .CV2OverrideValue • .CV3OverrideValue
Target Value	Type the Target calue for CV1, CV2, or CV3 output.		• .CV1Target • .CV2Target • .CV3Target

Maintenance Tab Page 2

Page 2 of the MMC Maintenance tab shows the following information:

- Error value for PV1 and PV2
- Model factor for PV1 and PV2
- Oversample Δt (seconds)
- Time in seconds used to calculate the process output
- Operator inputs for process gain time, model gain, model time constant, model dead time, and response time constant



The following table shows the functions of page 2 of the MMC Maintenance tab.

Table 66 - MMC Maintenance Tab Page 2 Description

Function	Action	Security	Configuration Parameters
Process Gain Sign	Click to select '+' or '-' as the Process Gain sign for PV1: CV1, CV2, or CV3, PV2: CV1, CV2, or CV3. This is used only for Autotune and is the sign of the process gain (ΔPV1/ΔCV1).	Engineering Configuration (Code E)	• .CV1PV1ProcessGainSign • .CV2PV1ProcessGainSign • .CV3PV1ProcessGainSign • .CV1PV2ProcessGainSign • .CV2PV2ProcessGainSign • .CV3PV2ProcessGainSign
Model Gain (%)	Type the Operator model gain for PV1: CV1, CV2, or CV3, PV2: CV1, CV2, or CV3. This is the internal model gain for CV# - PV#. enter a positive or negative gain depending on the process direction.	Configuration and Tuning Maintenance (Code D)	• .CV1PV1ModelGain • .CV2PV1ModelGain • .CV3PV1ModelGain • .CV1PV2ModelGain • .CV2PV2ModelGain • .CV3PV2ModelGain
Model Time Constant (seconds)	Type the internal model time constant for PV1: CV1, CV2, or CV3, PV2: CV1, CV2, or CV3 in seconds.		• .CV1PV1ModelTC • .CV2PV1ModelTC • .CV3PV1ModelTC • .CV1PV2ModelTC • .CV2PV2ModelTC • .CV3PV2ModelTC

Table 66 - MMC Maintenance Tab Page 2 Description

Function	Action	Security	Configuration Parameters
Model Deadtime	Type the internal model deadtime for PV1: CV1, CV2, or CV3, PV2: CV1, CV2, or CV3 in seconds.	Configuration and Tuning Maintenance (Code D)	• .CV1PV1ModelDT • .CV2PV1ModelDT • .CV3PV1ModelDT • .CV1PV2ModelDT • .CV2PV2ModelDT • .CV3PV2ModelDT
Response Time Constant (seconds)	Type the internal Response time constant for PV1: CV1, CV2, or CV3, PV2: CV1, CV2, or CV3. This is the tuning parameter that determines the speed of the control variable action for CV# - PV# in seconds.		• .CV1PV1RespTC • .CV2PV1RespTC • .CV3PV1RespTC • .CV1PV2RespTC • .CV2PV2RespTC • .CV3PV2RespTC

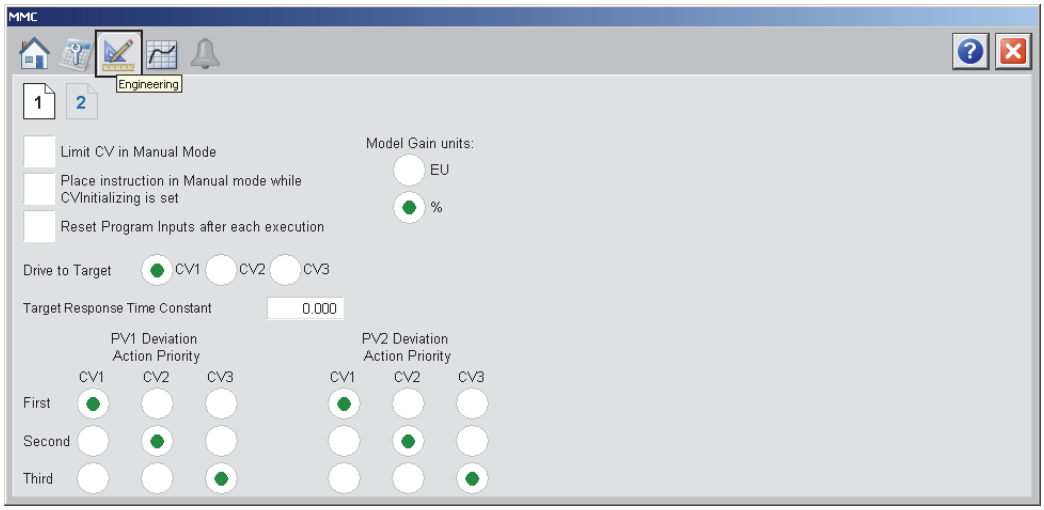
Engineering Tab

The Engineering tab provides access to device configuration parameters and ranges, options for device and I/O setup, displayed text, and faceplate-to-faceplate navigation settings, for initial system commissioning or later system changes.

The Engineering tab is divided into two pages.

Engineering Tab Page 1

Page 1 of the MMC Engineering tab has various Operator inputs/options for the CV, Cascade/Ratio mode, and Windup CV.



The following table shows the functions of page 1 of the MMC Engineering tab.

Table 67 - MMC Engineering Tab Page 1 Description

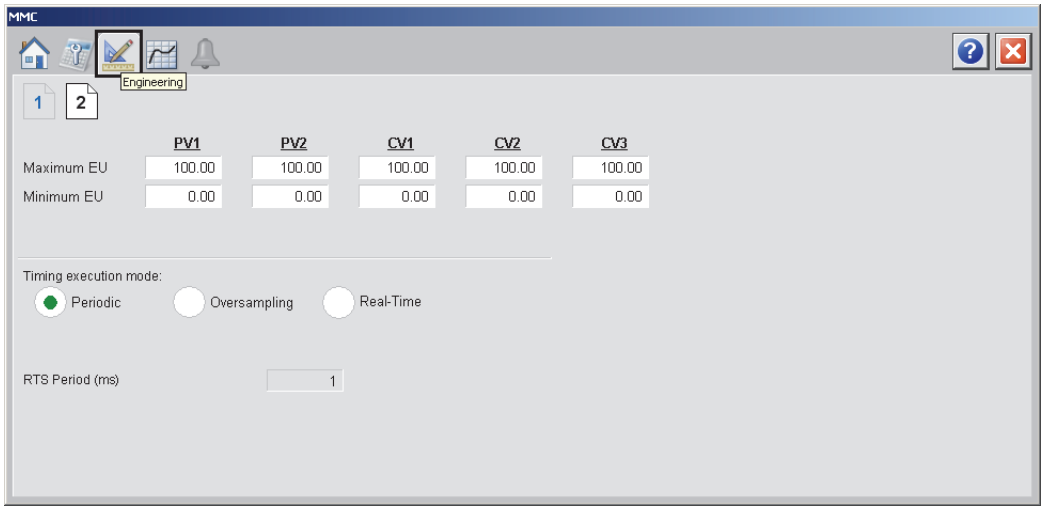
Function	Action	Security	Configuration Parameters
Limit CV in manual Mode	Limit CV1, CV2, or CV3 in Manual mode. If in Manual mode and CVManLimiting is **true**, CV1, CV2, and CV3 are limited by the CV1, CV2, and CV3 HLimit and CV1, CV2, and CV3 LLimit values.	Engineering Configuration (Code E)	.CVManLimiting
Place instruction in Manual mode while CVInitializing is set	Check to set the Loop mode to manual when CV initialization is requested. Clear the checkbox to leave the Loop mode unchanged when initialization is requested. When the initialization request clears, the loop resumes control in its previous Loop mode.		.ManualAfterInit
Reset Program Inputs after each execution	Check to reset Program control values after each execution.		.ProgValueReset
Model Gain units	Select either 'EU' or '%' for the Model Gain units in EU or '% of span'.		.GainEUSpan
Drive to Target: CV1 CV2 CV3	Click to select the CV to be driven to its target.		.TargetCV
Target Response Time Constant	Type the value that determines the speed with which the control variables approach their target values.		.TargetRespTC
PV1 Deviation Action Priority: First: CV1, CV2, or CV3 Second: CV1, CV2, or CV3 Third: CV1, CV2, or CV3	 Click to select the first CV to act to compensate for PV1-SP1 deviation. Click to select the second CV to act to compensate for PV1-SP1 deviation. Click to select the third CV to act to compensate for PV1-SP1 deviation.	Engineering Configuration (Code E)	• .PV1Act1stCV • .PV1Act2ndCV • .PV1Act3rdCV
PV2 Deviation Action Priority: First: CV1, CV2, or CV3 Second: CV1, CV2, or CV3 Third: CV1, CV2, or CV3	 Click to select the first CV to act to compensate for PV2-SP2 deviation. Click to select the second CV to act to compensate for PV2-SP2 deviation. Click to select the third CV to act to compensate for PV2-SP2 deviation.		• .PV2Act1stCV • .PV2Act2ndCV • .PV2Act3rdCV

Engineering Tab Page 2

Page 2 of the MMC Engineering tab has various Operator inputs/options for the PV, CV, and Timing Execution mode.

The following values are displayed:

- RTS period
- Operator inputs for Maximum and Minimum EU for PV1, PV2, CV1, CV2, and CV3.
- Operator options for Timing Execution mode.



The following table shows the functions of page 2 of the MMC Engineering tab.

Table 68 - MMC Engineering Tab Page 2 Description

Function	Action	Security	Configuration Parameters
Maximum/ Minimum EU: PV1 PV2 CV1 CV2 CV3	Type the maximum /minimum PV and CV values in engineering units.	Engineering Configuration (Code E)	• .PVEUMax • .CVEUMax • .PVEUMin • .CVEUMin
CV Value in Override Mode	Type the CV value when in Override mode.		.CVOverrideValue
Timing execution mode	Click to select Periodic, Oversampling, or Real-Time for the execution mode.		.TimingMode

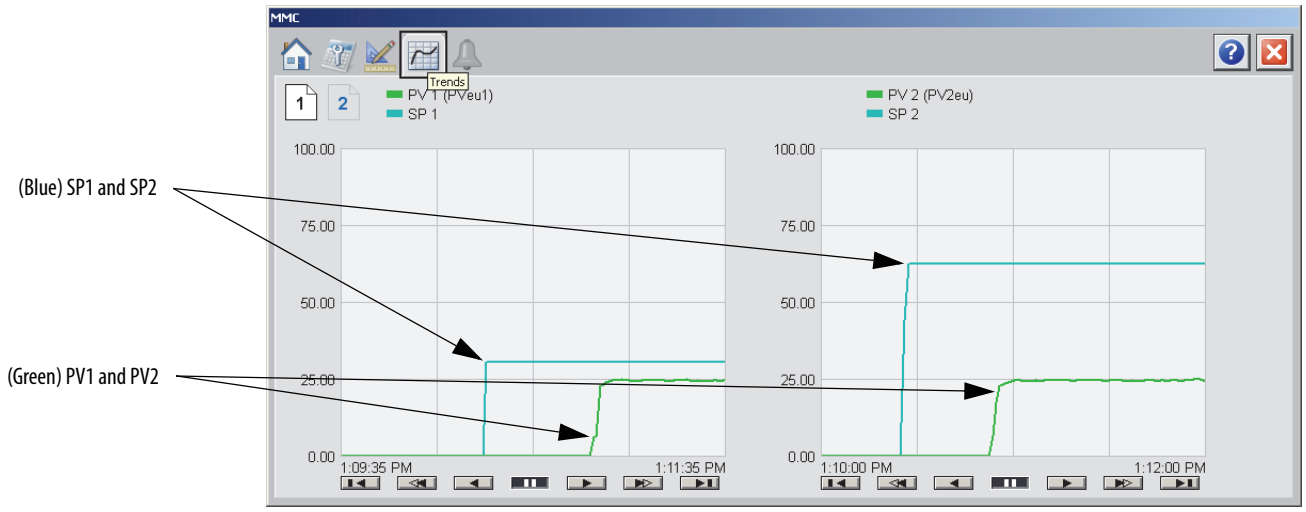
Trends Tab

The Trends tab shows trend charts of key device data over time. These faceplate trends provide a quick view of current device performance to supplement, but not replace, dedicated historical or live trend displays.

The Trends tab is divided into two pages.

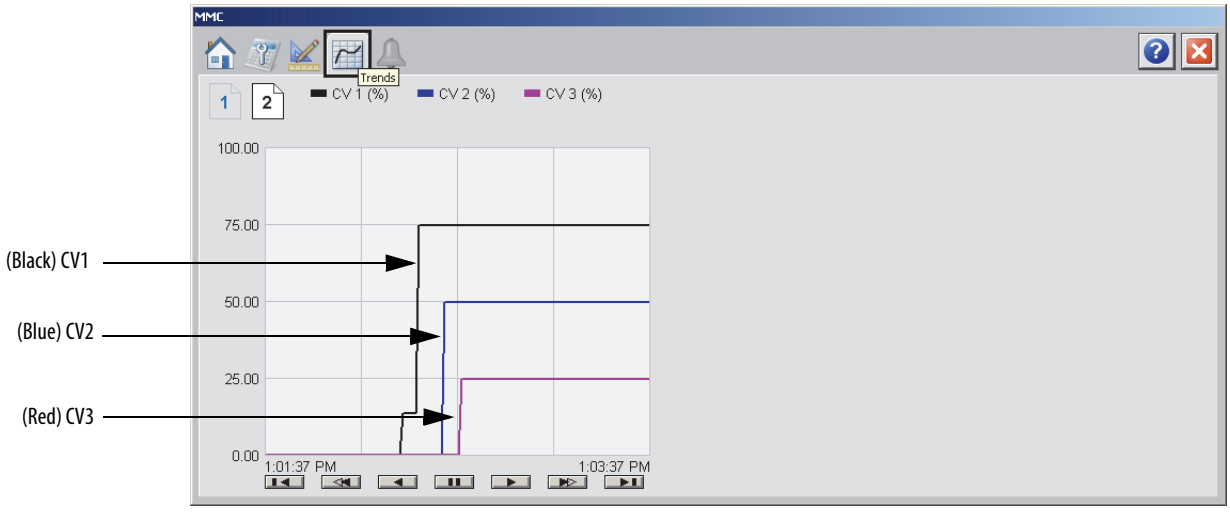
Trends Tab Page 1

Page 1 of the MMC Trends tab shows the relationship between PV1 (PVeu1) and SP1 for the same time period, and PV2 (PV2eu) and SP2 for the same time period.



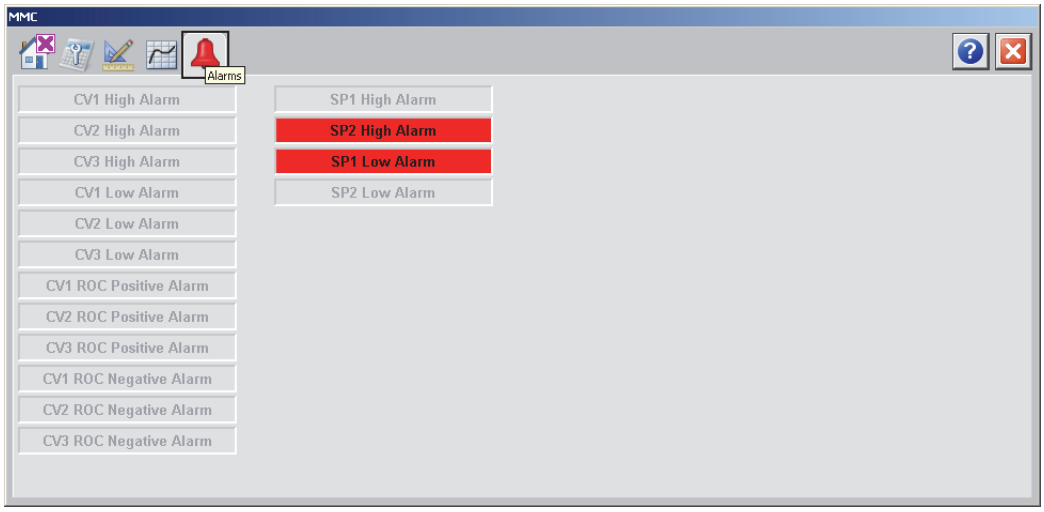
Trends Tab Page 2

Page 2 of the MMC Trends tab shows the waveforms for CV1, CV2, and CV3 for the same time period.



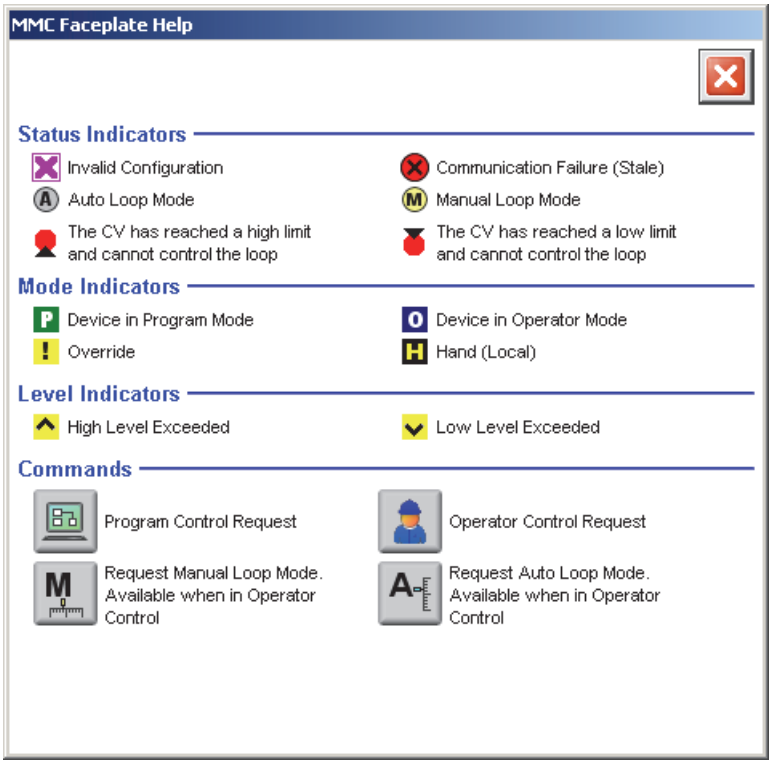
Alarms Tab

The MMC Alarms tab shows all the available alarms for the device and their current status.



Faceplate Help

The Faceplate Help page shows the indicators and command buttons that are used by the Modular Multivariable Control (MMC) function block.



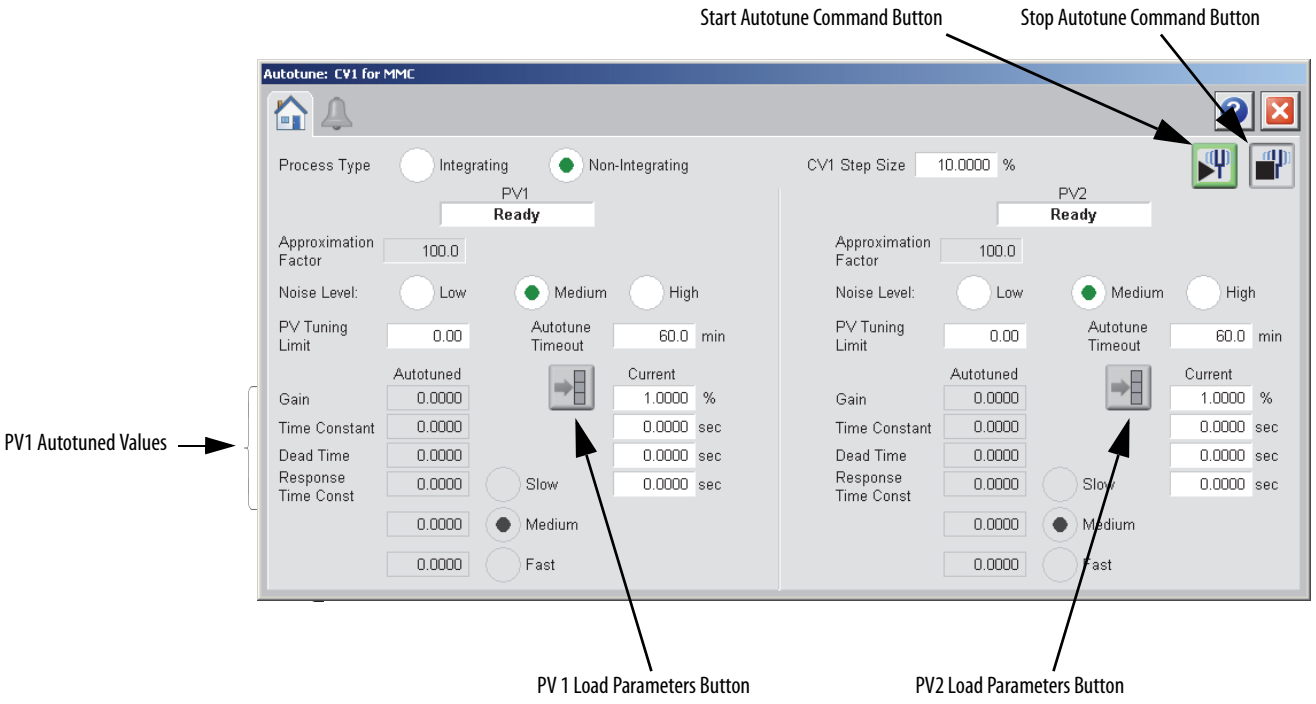
Modular Multivariable Control (MMC) Autotune

The faceplates in this section let you access all of the necessary parameters to autotune the MMC function block as well as hand-tune the instruction.

MMC Autotune Operator Tab

The MMC Autotune Operator tab shows the following information:

- Approximation Factor For PV1 and PV2
- Autotuned Gain, Time Constant, and Dead Time for PV1 and PV2.
- Response Time Constant for slow response, medium response, and fast response speeds for PV1 and PV2
- Load selected parameters into CV configuration parameters buttons for PV1 and PV2
- Start CV autotune button
- Stop CV autotune button



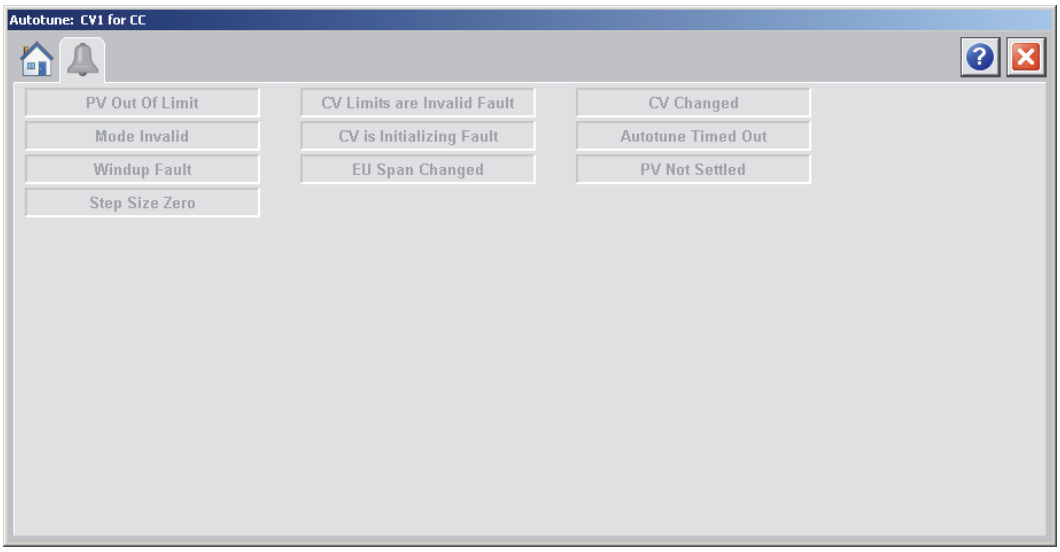
The following table lists the functions on of the MMC Autotune Operator tab.

Table 69 - MMC Autotune Operator Tab Description

Function	Action	Security
Process Type	Click Integrating or Non-inegrating to select the Process Type for PV1 and PV2.	Configuration and Tuning Maintenance (Code D)
Approximation Factor	Type a value for the non-integrating approximation factor for PV1 or PV2. **IMPORTANT:** You can enter this value only when the Process Type is 'Integrating'.	
Noise Level	Click to select an estimate of the noise level (Low, Medium, or High) expected on the PV1 or PV2 to compensate for it during tuning.	
PV Tuning Limit	Type a value for the PV1 or PV2 tuning limit scaled in the PV1 or PV2 units	Configuration and Tuning Maintenance (Code D)
Autotune Timeout	Type a value for the maximum time for PV1 or PV2 autotune to complete following the CV step change. When PV1 or PV2 autotune exceed this time, tuning is aborted.	
Currrent Gain	Type a value for the internal model gain for V1PV1, CV2PV1, CV3PV1, CV1PV2, CV2PV2, or CV3PV2.	
Current Time Constant	Type a value for the internal model time constant for CV1PV1, CV2PV1, CV3PV1, CV1PV2, CV2PV2, or CV3PV2.	
Currect Dead Time	Type a value for the internal model deadtime for V1PV1, CV2PV1, CV3PV1, CV1PV2, CV2PV2, or CV3PV2.	
Current Response Time Constant	Type a value for the tuning parameter that determines the speed of the CV action for V1PV1, CV2PV1, CV3PV1, CV1PV2, CV2PV2, or CV3PV2.	
Load Parameters	Click to copy selected values into the current IMC tags.	
Start Autotune	Click to start the autotune request.	
Stop Autotune	Click to abort the autotune request.	

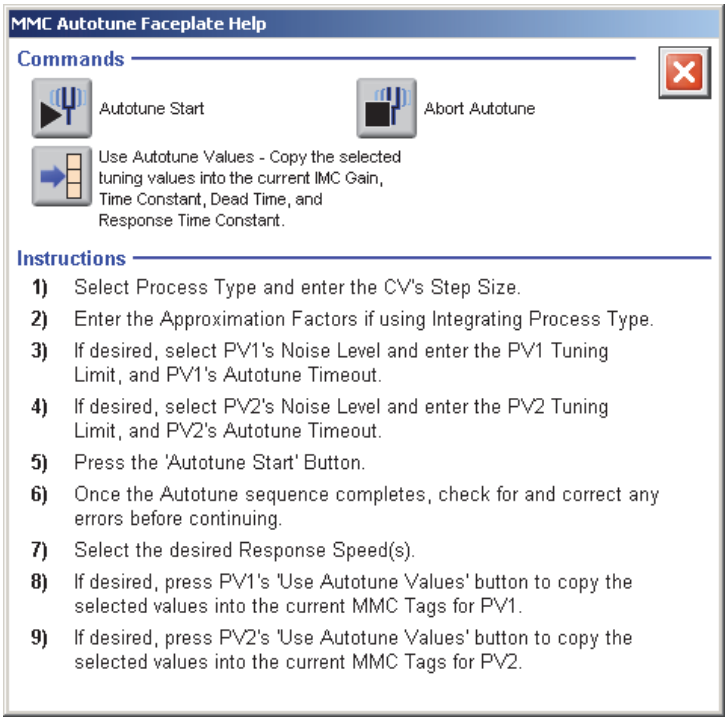
MMC Autotune Alarms Tab

The MMC Autotune Alarms tab shows all the available alarms for the device and their current status.



MMC Autotune Faceplate Help

The MMC Autotune Faceplate Help page shows the command buttons that are used by MMC Autotune. The faceplate also provides basic instruction on how to use Autotune.



Proportional + Integral + Derivative Enhanced (PIDE)

The Proportional + Integral + Derivative Enhanced (PIDE instruction provides enhanced capabilities over the standard PID instruction. The instruction uses the velocity form of the PID algorithm. The gain terms are applied to the change in the value of error or PV, not the value of error or PV.

Visualization Files

The following files are required to use the Built-In PIDE Object and can be downloaded from the Product Compatibility and Download Center at http://www.rockwellautomation.com/rockwellautomation/support/pcdc.page.

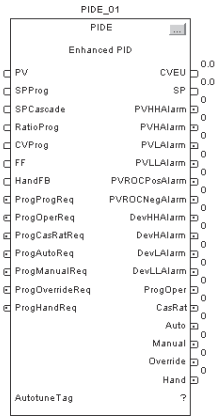
IMPORTANT Files must be imported in the following order: image files, then global object files, and then graphic files. This order is required to properly configure the visualization files.

Table 70 - Proportional + Integral + Derivative Enhanced (PIDE) Visualization File Types

Application Type	File Type	FactoryTalk View SE Software	FactoryTalk View ME Software	Description
Graphics - Displays	GFX	(RA-BAS) Common Analog-Edit	N/A	Faceplate used for analog input data entry. The FactoryTalk View ME faceplates use the native analog input data entry so no file is required.
		(RA-BAS) Built-In PIDE Faceplate	(RA-BAS-ME) Built-In PIDEFaceplate	The faceplate display used for the PIDE object.
		(RA-BAS) Built-In PIDE Help	(RA-BAS-ME) Built-In PIDEHelp	Help information that is accessed from the PIDE Help faceplate.
		(RA-BAS) Built-In PIDE Quick	(RA-BAS-ME) Built-In PIDE Quick	The Quick display used for the PIDE object.
Optional Graphic Displays		(RA-BAS) Built-In Autotune-Faceplate	(RA-BAS-ME) Built-In Autotune-Faceplate	The faceplate display used for the Autotune object.
		(RA-BAS) Buit-In Autotune-Help	(RA-BAS-ME) Buit-In Autotune-Help	Help information that is accessed from the PIDE Autotune Help faceplate.
Graphics - Global Objects	GGFX	(RA-BAS) Common Faceplate Objects	(RA-BAS-ME) Common Faceplate Objects	Common global objects used on all Process Object faceplates.
		(RA-BAS) BuiltIn Faceplate Objects	(RA-BAS-ME) BuiltIn Faceplate Objects	Global objects used on PIDE faceplates.
		(RA-BAS) BuiltIn Graphics Librarys	(RA-BAS-ME) BuiltIn Graphics Librarys	Builtin display elements used to build process graphics.
		(RA-BAS) BuiltIn Help Objects	(RA-BAS-ME) BuiltIn Help Objects	BuiltIn global objects used for all BuiltIn help displays.
Graphics - Images	BMP	All .png files in the images folder	All .png files in the images folder	These are the common icons used in the global objects and faceplates for all Process Objects.
HMI Tags	CSV	N/A	FTVME_PlantPAxLib_Tags_3_1_**00**.csv(1)	These tags must be imported into the FactoryTalk View ME project to support switching tabs on any Process Object faceplate.

(1) The service release number (boldfaced) can change as service revisions are created.

Display Elements



A display element (global object) is created once and can be referenced multiple times on multiple displays in an application. When changes are made to the original (base) object, the instantiated copies (reference objects) are automatically updated. Use of global objects, in conjunction with tag structures in the ControlLogix system, aid consistency and save engineering time.

s

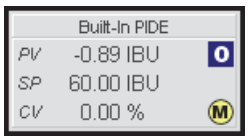
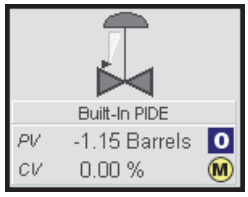
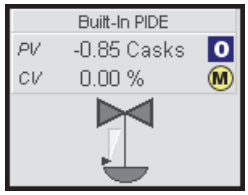
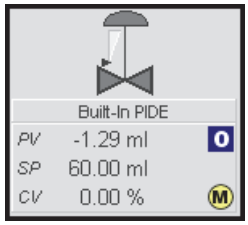
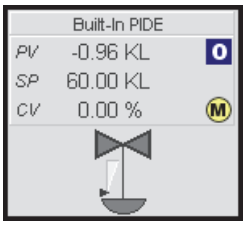
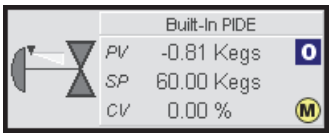
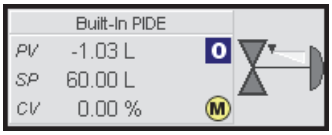
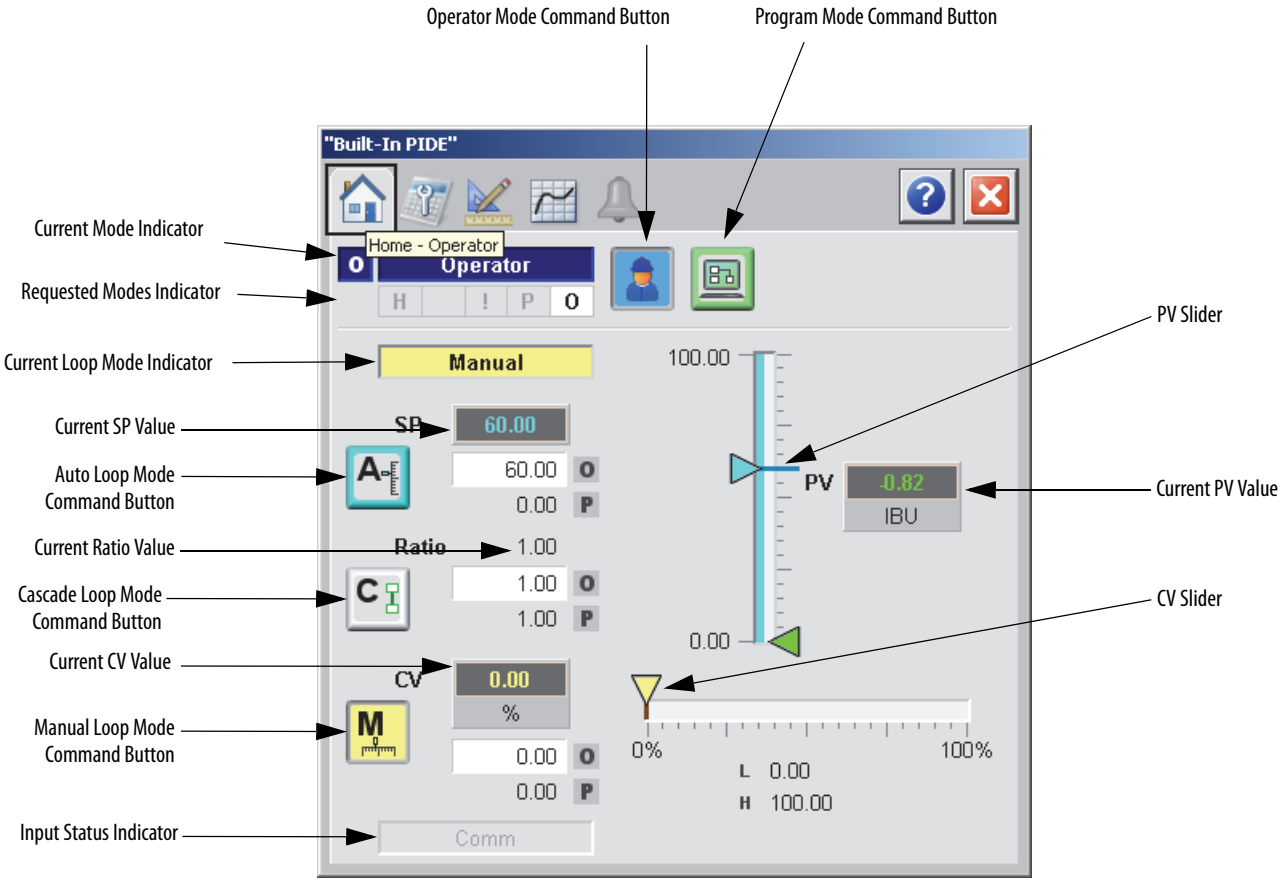
	Display Element	Description
		Proportional + Integral + Derivative Enhanced object with one Process Variables and one Control Variable.
		Proportional + Integral + Derivative Enhanced object with one Process Variable, one Setpoint, and one Control Variable.
GO_BuiltIn_PIDE_CV		Proportional + Integral + Derivative Enhanced (normal Control Valve for horizontal pipe) object with one Process Variable and one Control Variable.
GO_BuiltIn_PIDE_CV1		Proportional + Integral + Derivative Enhanced (inverted Control Valve for horizontal pipe) object with one Process Variable and one Control Variable.
GO_BuiltIn_PIDE_CV2		Proportional + Integral + Derivative Enhanced (Control Valve for vertical pipe to the left) object with one Process Variable and one Control Variable.
GO_BuiltIn_PIDE_CV3		Proportional + Integral + Derivative Enhanced (Control Valve for vertical pipe to the right) object with one Process Variable and one Control Variable.
GO_BuiltIn_PIDE_CV4		Proportional + Integral + Derivative Enhanced (normal Control Valve for horizontal pipe) object with one Process Variable, one Setpoint, and one Control Variable.

Table 71 - PIDE Display Elements Descriptions

Display Element Name	Display Element	Description
GO_BuiltIn_PIDE_CV5		Proportional + Integral + Derivative Enhanced (inverted Control Valve for horizontal pipe) object with one Process Variable, one Setpoint, and one Control Variable.
GO_BuiltIn_PIDE_CV6		Proportional + Integral + Derivative Enhanced (Control Valve for vertical pipe to the left) object with one Process Variable, one Setpoint, and one Control Variable.
GO_BuiltIn_PIDE_CV7		Proportional + Integral + Derivative Enhanced (Control Valve for vertical pipe to the right) object with one Process Variable, one Setpoint, and one Control Variable.

Operator Tab

The faceplate initially opens to the Operator ('Home') tab. From here, an operator can monitor the device status and manually operate the device when it is in Operator mode.



The following table lists the functions on of the CC Operator tab.

Table 72 - PIDE Operator Tab Description

Function	Action	Security
	Click to request Operator mode.	Manual Device Operation (Code B)
	Click to request Program mode.	
	Click to go to Manual Loop mode.	Normal Operation of Devices (Code A)
	Click to go to Cascade Loop mode.	

Table 72 - PIDE Operator Tab Description

Function	Action	Security
	Click to go to Auto Loop mode.	Normal Operation of Devices (Code A)
Operator Setpoint Value	Type a value for the loop Setpoint.	
Operator Ratio Value	Type a value for the ratio operator multiplier.	
Operator CV Value	Type a value for CV.	
CV Slider	Move this slider to adjust the loop CV output.	None
PV Slider	Move this slider to adjust the loop PV value.	

Maintenance Tab

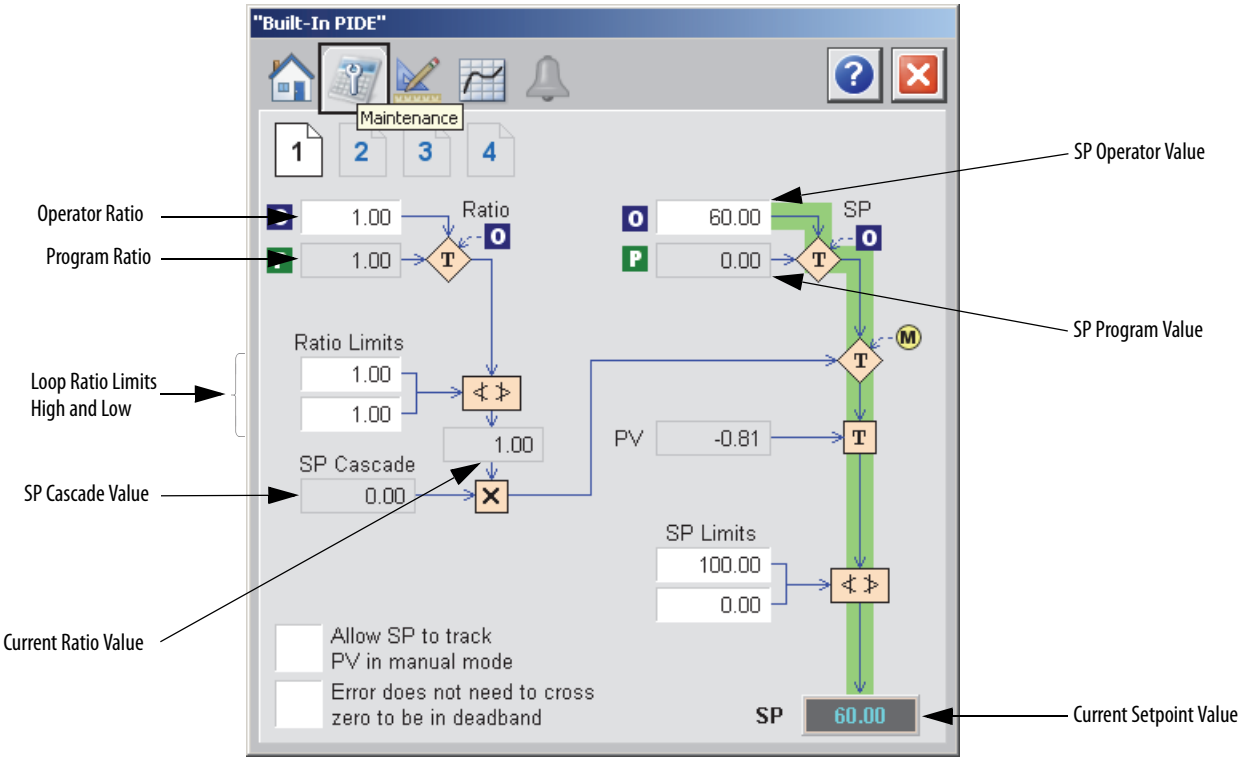
Maintenance personnel use the information and controls on the Maintenance tab to make adjustments to device parameters, troubleshoot and temporarily work around device problems, and disable the device for routine maintenance.

The Maintenance tab is divided into four tabs.

Maintenance Tab Page 1

Page 1 of the Maintenance tab shows the following information:

- Ratio program multiplier
- SP program value scaled in PV units
- Current ratio multiplier
- Current Setpoint value
- SP Cascade value scaled in PV units



The following table shows the functions of page 1 of the PIDE Maintenance tab.

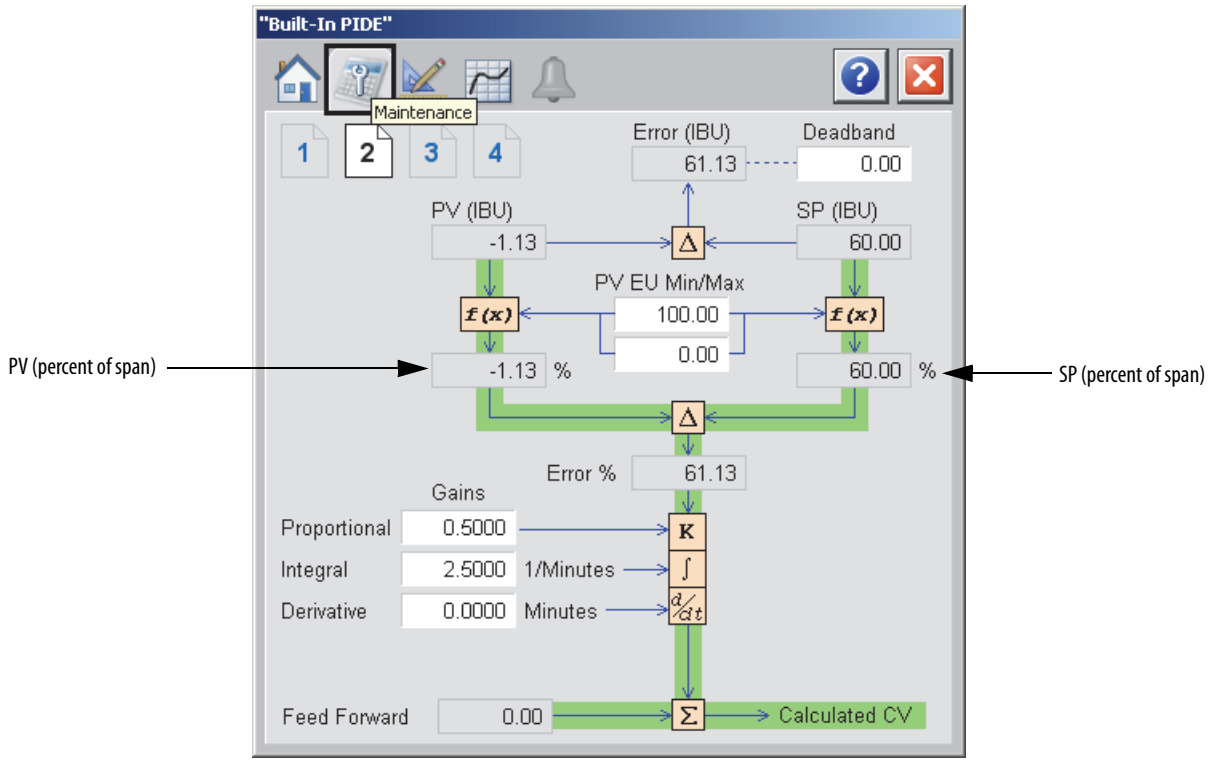
Table 73 - PIDE Maintenance Tab Page 1 Description

Function	Action	Security	Configuration Parameters
Operator Ratio	Type a ratio operator multiplier.	Normal Operation of Devices (Code A)	.RatioOper
Loop Ratio High and Low	Type values for the high and low ratio limits. These values limit the value of Ratio obtained from Operator or Program Ratio.	Configuration and Tuning Maintenance (Code D)	• .RatioHLimit • .RatioLLimit
SP Limits High and Low	Type the high and low limits for the setpoint.		• .SPHLimit • .SPLLimit
Allow SP to track PV in Manual mode	Click to have SP track PV when in Manual mode. This setting is ignored when in Cascade or Auto mode.	Equipment Maintenance (Code C)	.PVTracking
Error does not need to cross zero to be in deadband	Click to disable zero crossing for the deadband calculation.	Configuration and Tuning Maintenance (Code D)	.ZCOff

Maintenance Tab Page 2

Page 2 of the PIDE Maintenance tab shows the following information:

- Process error
- PV (scaled input)
- PV (expressed as a percent of span)
- SP (expressed as a percent of span)
- Error (expressed as a percent of span)
- Feed forward value



The following table shows the functions of page 2 of the Maintenance tab.

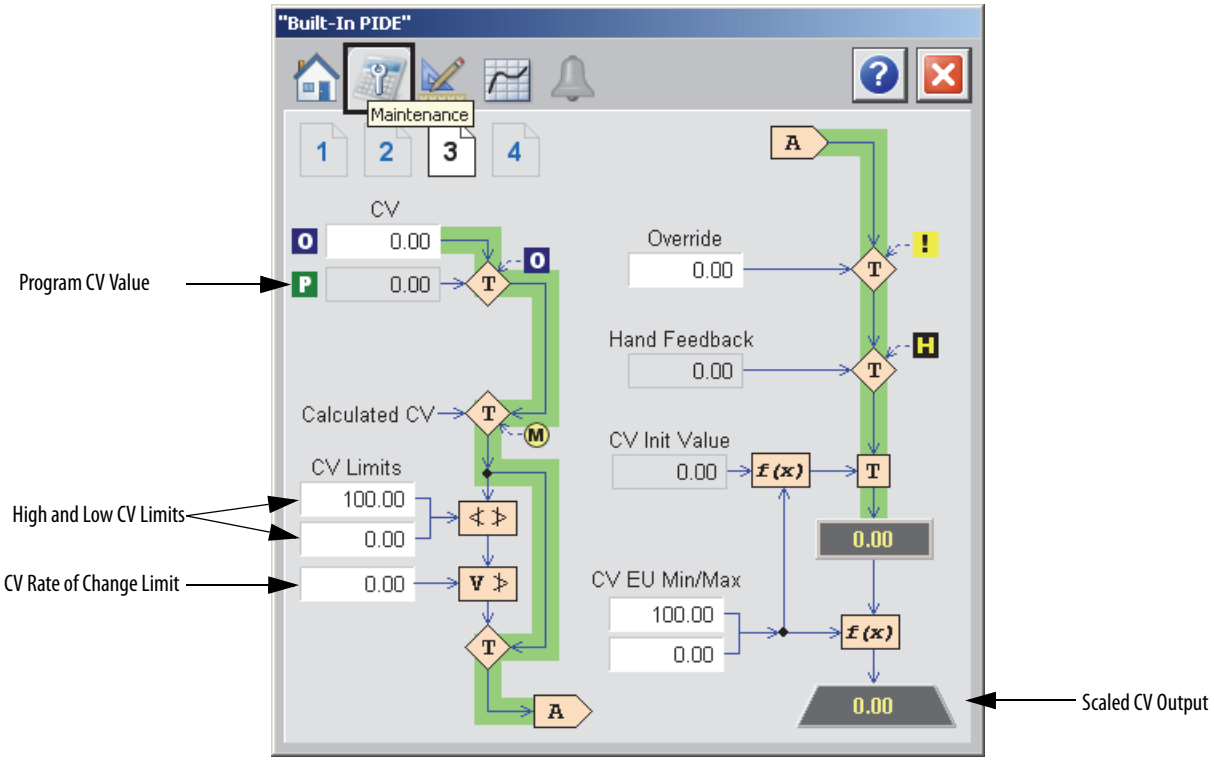
Table 74 - PIDE Maintenance Tab Page 2 Description

Function	Action	Security	Configuration Parameters
Deadband Value	Type a value for the zero crossing deadband range. Type zero to disable the zero crossing deadband checking.	Configuration and Tuning Maintenance (Code D)	.ZCDeadband
PV/EU Maximum and Minimum	Type a value for the maximum and minimum scaled values for PV.	Engineering Configuration (Code E)	• .PVEUMax • .PVEUMin
Gains: Proportional Integral Derivative	Type in a value for: Proportional gain Integral gain Derivative gain	Configuration and Tuning Maintenance (Code D)	• .PGain • .IGain • .DGain

Maintenance Tab Page 3

Page 3 of the PIDE Maintenance tab shows the following information:

- CV (when in the Program Manual mode)
- CV Hand feedback value (when in Hand mode and HandFBFault is clear)
- CV initial value
- Scaled CV output



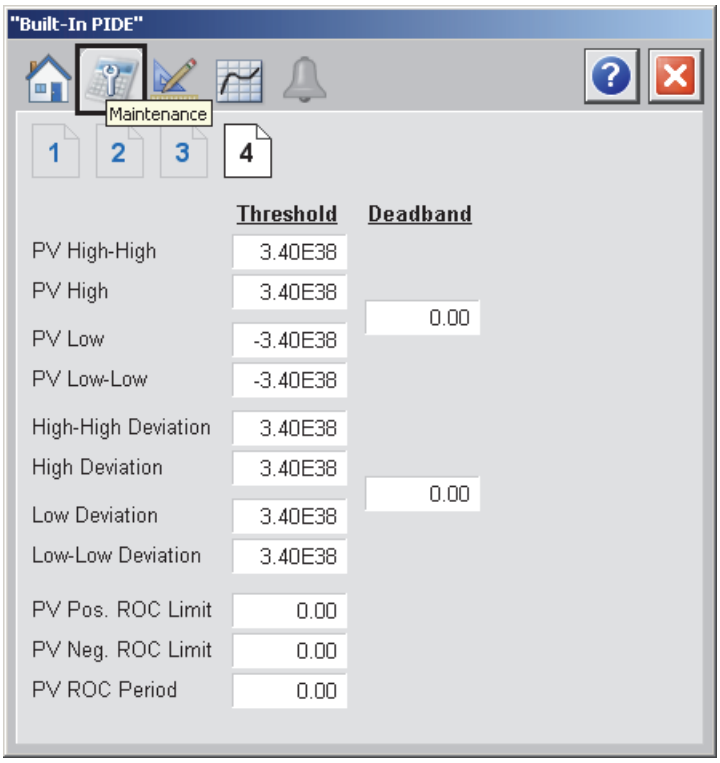
The following table shows the functions of page 3 of the Maintenance tab.

Table 75 - PIDE Maintenance Tab Page 3 Description

Function	Action	Security	Configuration Parameters
Operator CV Value	Type a value for CV when in the Operator Manual mode.	Normal Operation of Devices (Code A)	.CVOper
Override Value	Type a value for CV when in the Override mode.	Configuration and Tuning Maintenance (Code D)	.CVOverride
CV High and Low Limits	Type values for the CV high and low limits.		• .CVHLimit • .CVLLimit
CV Rate of Change Limit	Type a value for CV in percent per second.		.CVROCLimit
CV EU Minimum and Maximum	Type values for the maximum and minimum values for CVEU.	Engineering Configuration (Code E)	• .CVEUMax • .CVEUMin

Maintenance Tab Page 4

Page 4 of the PIDE Maintenance tab has Operator inputs for PVs, deviations, and Range of Change limits and period.



The following table shows the functions of page 4 of the Maintenance tab.

Table 76 - PIDE Maintenance Tab Page 4 Description

Function	Action	Security	Configuration Parameters
PV High-High	Type values for the PV high-high, high, low, and low-low alarm limits (scaled in PV units).	Disable Alarms (Code H)	• .PVHHLimit • .PVHLimit • .PVLLimit • .PVLLLimit
PV High			
PV Low			
PV Low-Low			
PV Deadband	Type a value for the PV alarm limit deadband (scaled in PV units).		.PVDeadband
High-High Deviation	Type values for the Deviation high-high, high, low, and low-low alarm limits (scaled in PV units).		• .DevHHLimit • .DevHLimit • .DevLLimit • .DevLLLimit
High Deviation			
Low Deviation			
Low-Low Deviation			
Deviation Deadband	Type a value for the Deviation alarm limit deadband (scaled in PV units).		.DevDeadband
PV Positive ROC Limit	Enter values for the positive and negative rates of change alarm limits.		• .PVROCPosLimit • .PVROCNegLimit
PV Negative ROC Limit			

Table 76 - PIDE Maintenance Tab Page 4 Description

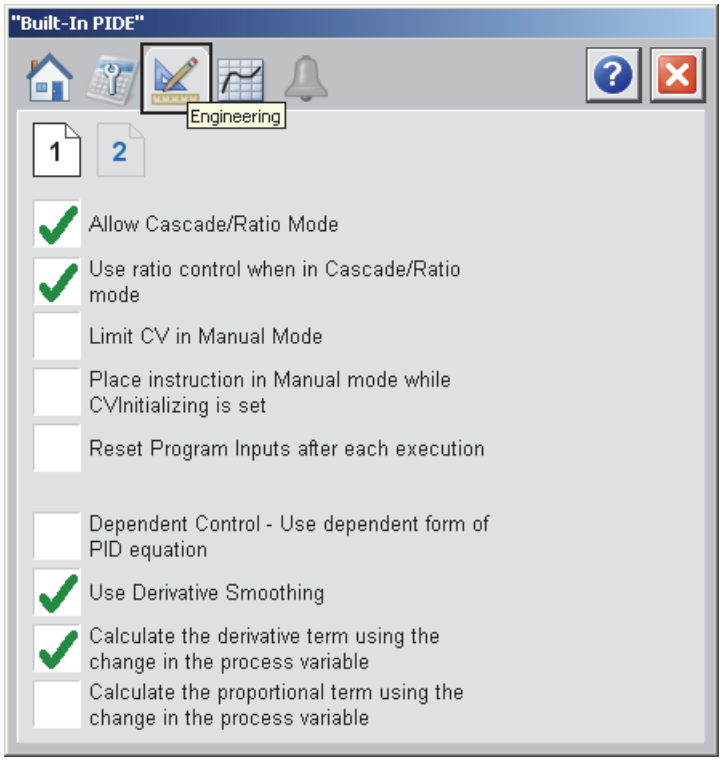
Function	Action	Security	Configuration Parameters
PV ROC Period	Type a value for the PV Rate of change sample period. This is the time period, in seconds, over which the rate of change for PV is evaluated. Type zero to disable the PV rate of change period checking.	Disable Alarms (Code H)	.PVROCPeriod

Engineering Tab

The Engineering tab provides access to device configuration parameters and ranges, options for device and I/O setup, displayed text, and faceplate-to-faceplate navigation settings, for initial system commissioning or later system changes.

The Engineering tab is divided into two tabs.

Engineering Tab Page 1



The following table shows the functions of page 1 of the Engineering tab.

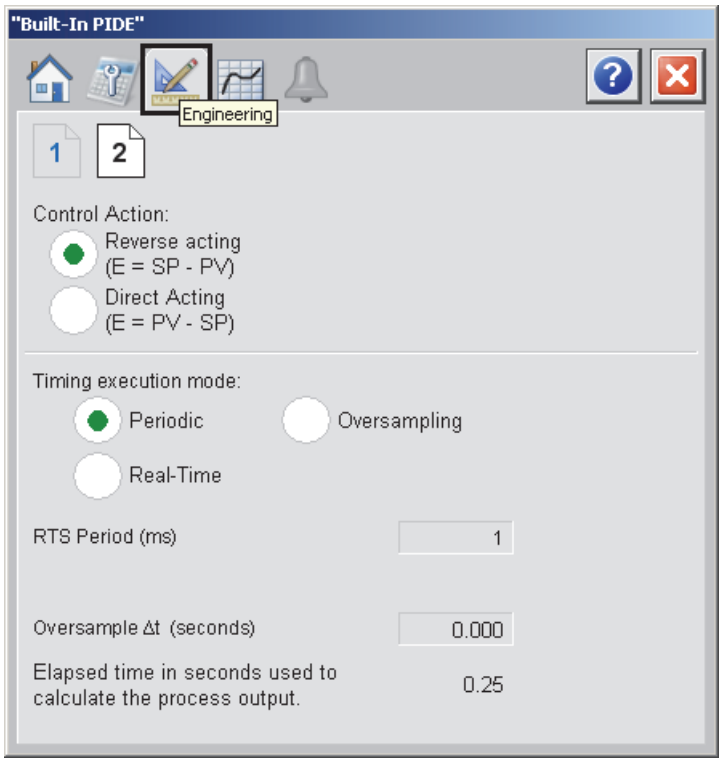
Table 77 - PIDE Engineering Tab Page 1 Description

Function	Action	Security	Configuration Parameters
Allow Cascade/ Ratio Mode	Check to enable Cascade/Ratio mod to be selected.	Engineering Configuration (Code E)	.AllowCasRat
Use ratio control when in Cascade/ Ratio mode	Check to enable ratio control when in Cascade/Ratio mode.		.UseRatio
Limit CV in Manual Mode	Check to limit CV in the Manual mode.		.CVManLimiting
Place instruction in Manual mode while CVInitializing is set	Check to set the Loop mode to manual when CV initialization is requested. Clear the checkbox to leave the Loop mode unchanged when initialization is requested. When the initialization request clears, the loop resumes control in its previous Loop mode.		.ManAfterInit
Reset program inputs after each execution	Click to clear all program request inputs after each execution of the instruction.		.ProgValueReset
Dependent Control - Use dependent form of PID equation	Click to use the dependent form of the PID equation. Clear this checkbox to use the independent form of the equations.		.DependIndepend
Use Derivative Smoothing	Click to smooth changes in the derivative term.		.DSmoothing
Calculate the derivative term using the change in the process variable	Click to calculate the derivative term (DeltaDTerm) by using the change in the process variable (PVPercent). Clear this checkbox to use the change in error (EPercent).		.PVEDerivative
Calculate the proportional term using the change in the process variable	Click to calculate the derivative term (DeltaDTerm) by using the change in process variable (PVPercent). Clear this checkbox to use the change in error (EPercent).		.PVEProportional

Engineering Tab Page 2

Page 2 of the PIDE Engineering tab shows the following information:

- RTS Period (milliseconds)
- Oversampling (seconds)
- Elapsed time in seconds used to calculate the process output
- Operator inputs for Control Action and Timing Execution mode



The following table shows the functions of page 2 of the Engineering tab.

Table 78 - PIDE Engineering Tab Page 2 Description

Function	Action	Security	Configuration Parameters
Control Action: Reverse acting (E = SP - PV) Direct Acting (E = PV - SP)	Click to select the method of calculating error.	Engineering Configuration (Code E)	.ControlAction
Timing execution mode: Periodic Oversampling Real-Time	Click to select the Timing execution mode.		.TimingMode

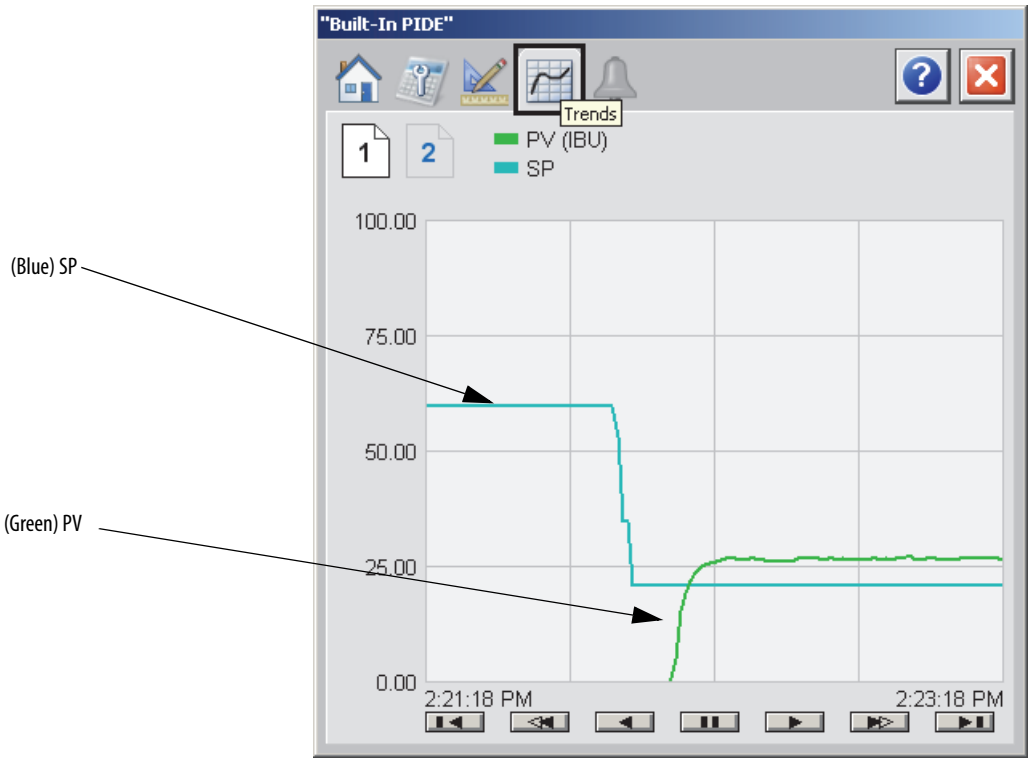
Trends Tab

The Trends tab shows trend charts of key device data over time. These faceplate trends provide a quick view of current device performance to supplement, but not replace, dedicated historical or live trend displays.

The Trends tab is divided into two pages.

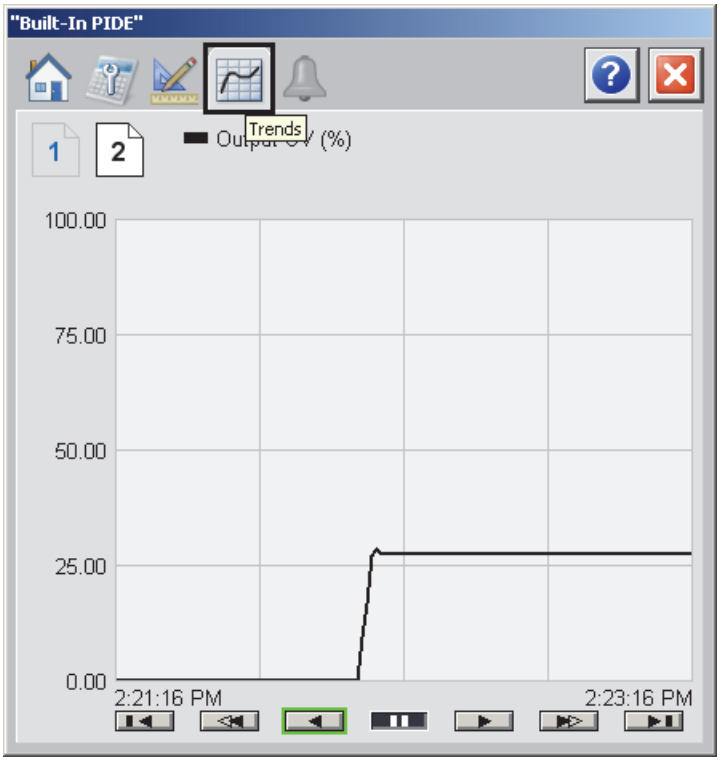
Trends Tab Page 1

Page 1 of the PIDE Trends tab shows the relationship between PV (IBU) and SP for the same time frame of a process.



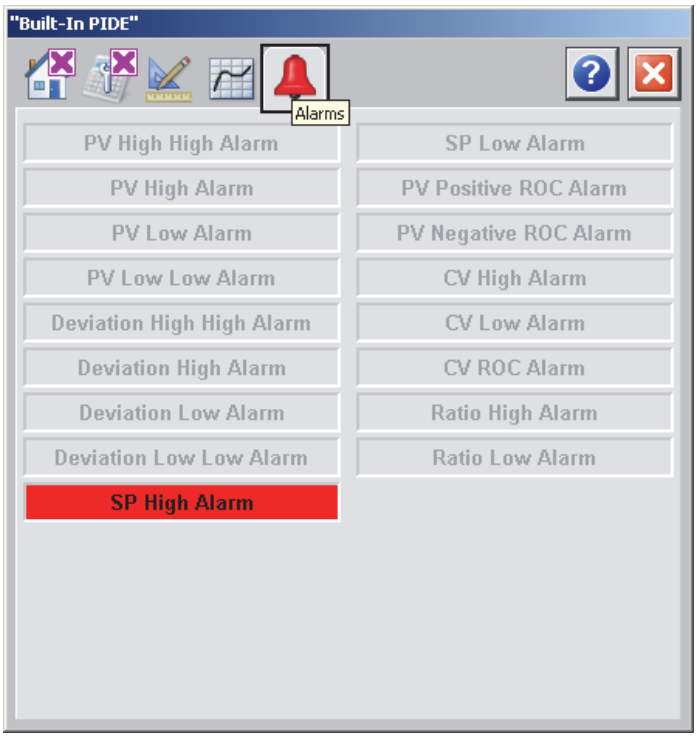
Trends Tab Page 2

Page 2 of the PIDE Trends tab shows the waveforms for the output CV.



Alarms Tab

The PIDE Alarms tab shows all the available alarms for the device and their current status.

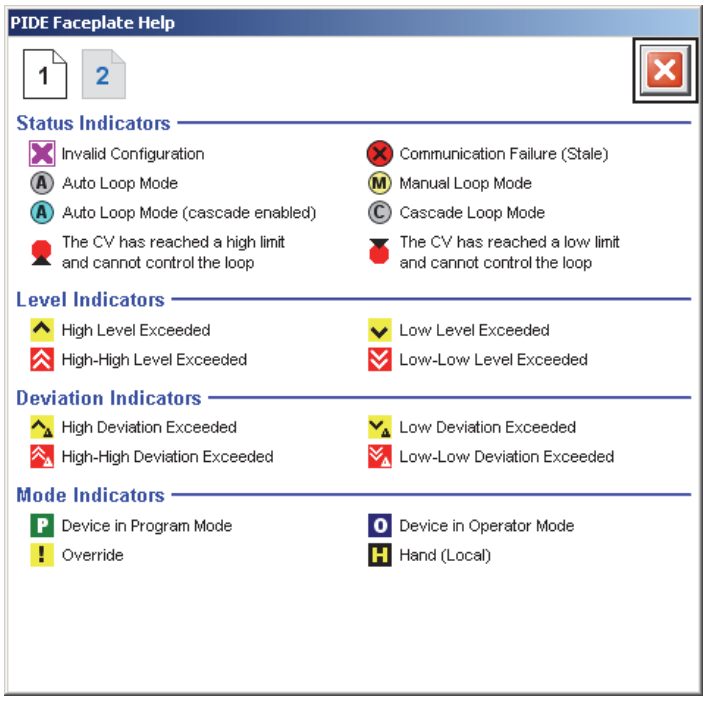


Faceplate Help

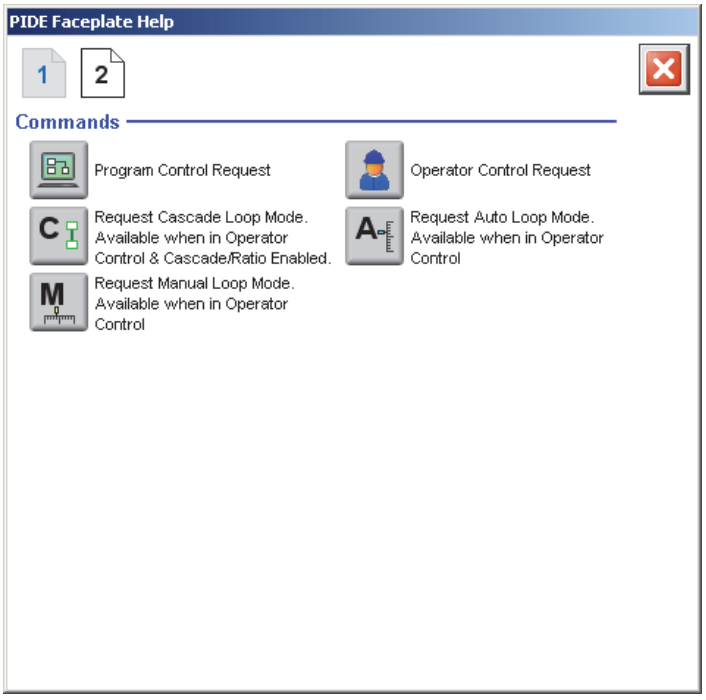
The Help faceplate shows the indicators and command buttons.

There are two Help pages.

Faceplate Help Page 1



Faceplate Help Page 2



Ramp/Soak (RMPS)

The Ramp/Soak (RMPS) instruction provides for a number of segments of alternating ramp and soak periods.

Visualization Files

The following files are required to use the Built-In RMPS Object and can be downloaded from the Product Compatibility and Download Center at http://www.rockwellautomation.com/rockwellautomation/support/pcdc.page.

IMPORTANT Files must be imported in the following order: image files, then global object files, and then graphic files. This order is required to properly configure the visualization files.

Table 79 - Ramp /Soak (RMPS) Visualization File Types

Application Type	File Type	FactoryTalk View SE Software	FactoryTalk View ME Software	Description
Graphics - Displays	GFX	(RA-BAS) Common Analog-Edit	N/A	Faceplate used for analog input data entry. The FactoryTalk View ME faceplates use the native analog input data entry so no file is required.
		(RA-BAS) Built-In RMPS-Faceplate	(RA-BAS-ME) Built-In RMPS Faceplate	The faceplate display used for the RMPS object.
		(RA-BAS) Built-In RMPS-Help	(RA-BAS-ME) Built-In RMPS Help	Help information that is accessed from the RMPS Help faceplate.
Graphics - Global Objects	GGFX	(RA-BAS) Common Faceplate Objects	(RA-BAS-ME) Common Faceplate Objects	Common global objects used on all Process Object faceplates.
		(RA-BAS) BuiltIn Faceplate Objects	(RA-BAS-ME) BuiltIn Faceplate Objects	Global objects used on RMPS faceplates.
		(RA-BAS) BuiltIn Graphics Librarys	(RA-BAS-ME) BuiltIn Graphics Librarys	Built-in display elements used to build process graphics.
		(RA-BAS) BuiltIn Help Objects	(RA-BAS-ME) BuiltIn Help Objects	Built-in global objects used for all Built-in help displays.
Graphics - Images	BMP	All .png files in the images folder	All .png files in the images folder	These are the common icons used in the global objects and faceplates for all Process Objects.
HMI Tags	CSV	N/A	FTVME_PlantPAxLib_Tags_3_1_**00**.csv(1)	These tags must be imported into the FactoryTalk View ME project to support switching tabs on any Process Object faceplate.

(1) The service release number (boldfaced) can change as service revisions are created.

Display Elements

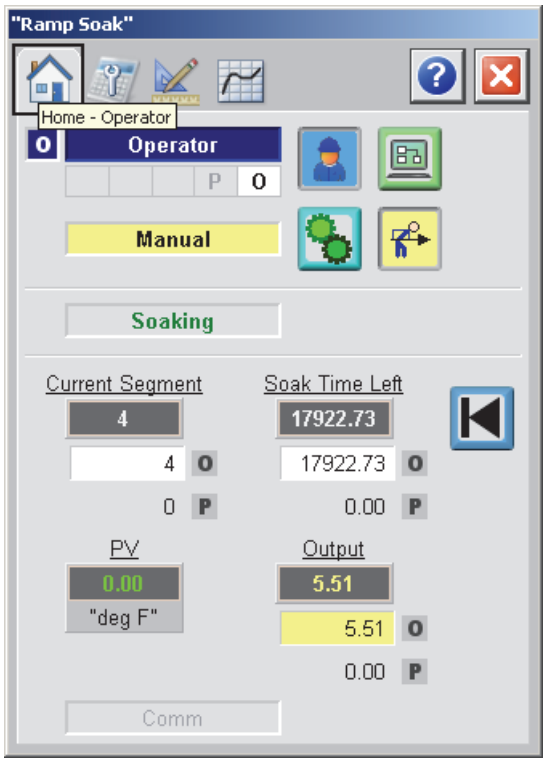
A display element (global object) is created once and can be referenced multiple times on multiple displays in an application. When changes are made to the original (base) object, the instantiated copies (reference objects) are automatically updated. Use of global objects, in conjunction with tag structures in the ControlLogix system, aid consistency and save engineering time.

Descriptions

Display Element	Description
	Ramp Soak global object.

Operator Tab

The faceplate initially opens to the Operator ('Home') tab. From here, an operator can monitor the device status and manually operate the device when it is in Operator mode.



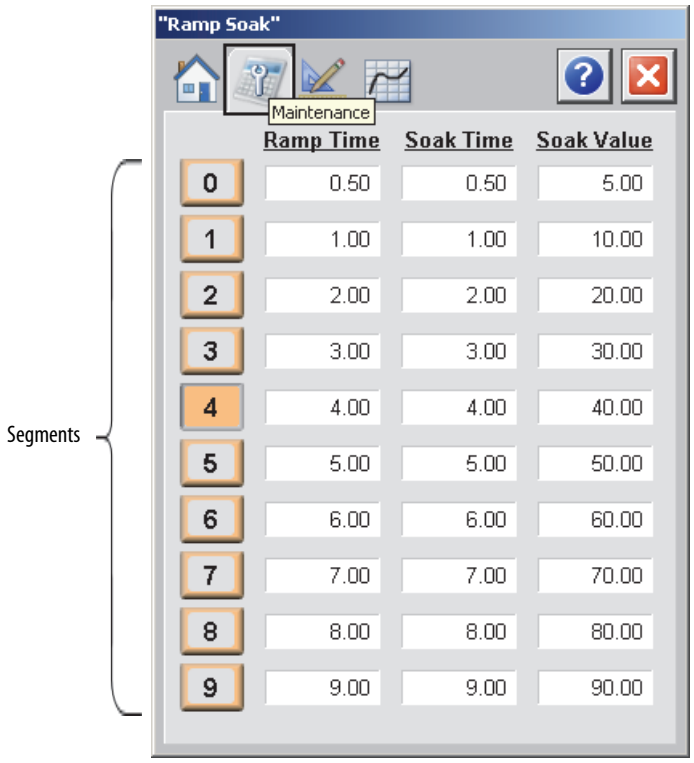
The following table lists the functions on of the RMPS Operator tab.

Table 81 - RMPS Operator Tab Description

Function	Action	Security
	Click to request Operator mode.	Manual Device Operation (Code B)
	Click to request Program mode.	
	Click to request Manual Loop mode.	Normal Operation of Devices (Code A)
	Click to request Auto Loop mode.	
	Click to initialize Current Segment and Soak Time Left.	
Operator Segment Value	Type a value for the Operator Segment. This value is used if Ramp/Soak is in the Manual mode.	
Operator Soak Time Left Value	Type a value for the Operator Soak Time Left. This value is used if Ramp/Soak is in the Manual mode.	
Operator Output Value	Type a value for the Operator output value. This value is used as the Output when Ramp/Soak is in the Manual mode.	

Maintenance Tab

The Maintenance tab has inputs for Ramp Time, Soak Time, and Soak Value for each segment.



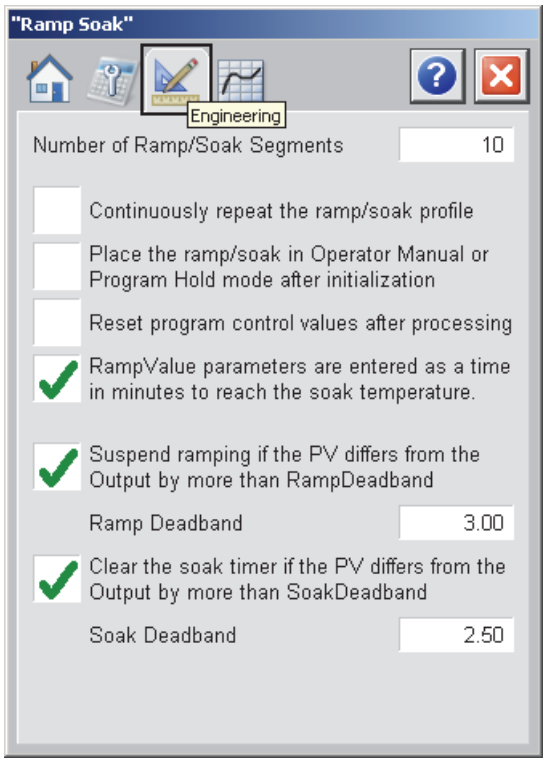
The following table shows the functions of the RMPS Maintenance tab.

Table 82 - RMPS Maintenance Tab Description

Function	Action	Security	Configuration Parameters
	Click to select the current segment.	Normal Operation of Devices (Code A)	.CurrentSeg
Ramp Time	Type a value for Ramp time, in minutes or units/minute, for the desired segment(s).	Configuration and Tuning Maintenance (Code D)	.RampValue
Soak Time	Type a value for Soak Time, in minutes, for the desired segment(s).		.SoakTime
Soak Value	Type a value for Soak Value for the desired segment(s).		.SoakValue

Engineering Tab

The Engineering tab provides access to device configuration parameters and ranges, options for device and I/O setup, displayed text, and faceplate-to-faceplate navigation settings, for initial system commissioning or later system changes.



The following table shows the functions of the RMPS engineering tab.

Table 83 - RMPS Engineering Tab Description

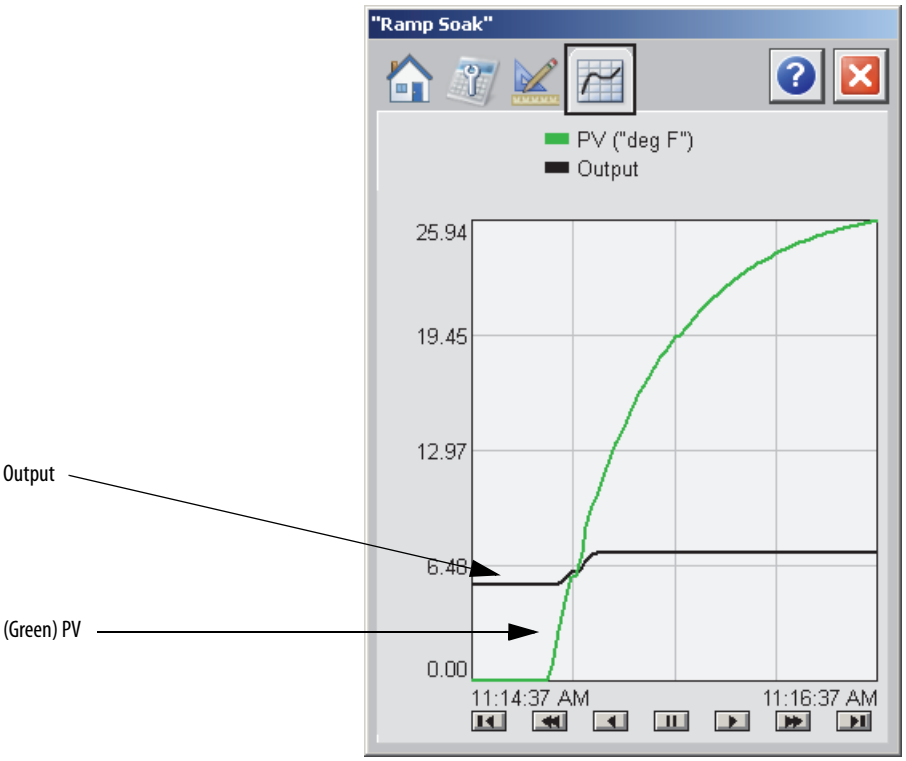
Function	Action	Security	Configuration Parameters
Number of Ramp/ Soak Segments		Engineering Configuration (Code E)	.NumberOfSegs
Continuously repeat the ramp/soak profile	Check to set for cyclic action. Clear this checkbox to set for single action. Cyclic action continuously repeats the ramp/soak profile. Single action performs the ramp/ soak profile once and then stops.		.CyclicSingle
Place the ramp/ soak in operator Manual or Program Hold mode after initialization	Click to set ramp/Soak in Manual or Program Hold mode after initialization. Clear this checkbox to have Ramp/Soak remain in it's previous mode after initialization completes.		.ManHoldAftInit
Reset program control values after processing	Click to set program control values.		.ProgValueReset

Table 83 - RMPS Engineering Tab Description

Function	Action	Security	Configuration Parameters
RampValue parameters are entered as a time in minutes to reach the soak temperature	Click to set if the RampValue parameter are entered in minutes. Clear this checkbox if the RampValue parameter is entered in units/minute.	Engineering Configuration (Code E)	.TimeRate
Suspend ramping if the PV differs from the Output by more than RampDeadband	Click to set Guaranteed Ramp. If set and the instruction is in Auto, ramping is temporarily suspended if the PV differs from the Output by more than RampDeadband.		.GuarRamp
Ramp Deadband	Type a value (Guaranteed Ramp Deadband) in engineering units that PV is allowed to differ from the output when GuarRamp is on.		.RampDeadband
Clear the soak timer if the PV differs from the Output by more than SoakDeadband	Click to clear the soak timer.		.GuarSoak
Soak Deadband	Type a value in engineering units that the PV is allowed to differ from the output when GuarSoak is on.		.SoakDeadband

Trends Tab

Page 1 of the PIDE Trends tab shows the relationship between PV (°F) and the output for the same time period.



Faceplate Help

The faceplate Help page shows the indicators and command buttons that are used by the Ramp Soak (RMPS) instruction.

Totalizer (TOT)

The TOT instruction provides a time-scaled accumulation of an analog input value.

Visualization Files

The following files are required to use the Built-In TOT Object and can be downloaded from the Product Compatibility and Download Center at http://www.rockwellautomation.com/rockwellautomation/support/pcdc.page.

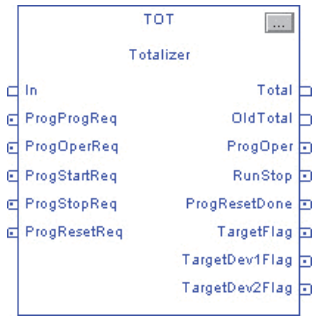
IMPORTANT Files must be imported in the following order: image files, then global object files, and then graphic files. This order is required to properly configure the visualization files.

Table 84 - Totalizer (TOT) Visualization File Types

Application Type	File Type	FactoryTalk View SE Software	FactoryTalk View ME Software	Description
Graphics - Displays	GFX	(RA-BAS) Common Analog-Edit	N/A	Faceplate used for analog input data entry. The FactoryTalk View ME faceplates use the native analog input data entry so no file is required.
		(RA-BAS) Built-In Totalizer-Help	(RA-BAS-ME) Built-In Totalizer-Help	Help information that is accessed from the Totalizer Help faceplate.
Optional Graphic Displays		(RA-BAS) Built-In Totalizer-Faceplate	(RA-BAS-ME) Built-In TotalizerFaceplate	The faceplate display used for the Totalizer object.
		(RA-BAS) Built-In TotalizerTgt-Faceplate	(RA-BAS-ME) Built-In TotalizerFaceplate	The faceplate display used for the Totalizer object.
Graphics - Global Objects	GGFX	(RA-BAS) Common Faceplate Objects	(RA-BAS-ME) Common Faceplate Objects	Common global objects used on all Process Object faceplates.
		(RA-BAS) BuiltIn Faceplate Objects	(RA-BAS-ME) BuiltIn Faceplate Objects	Global objects used on Totalizer faceplates.
		(RA-BAS) BuiltIn Graphics Librarys	(RA-BAS-ME) BuiltIn Graphics Librarys	Built-In display elements used to build process graphics.
		(RA-BAS) BuiltIn Help Objects	(RA-BAS-ME) BuiltIn Help Objects	Built-In global objects used for all BuiltIn help displays.
Graphics - Images	BMP	All .png files in the images folder	All .png files in the images folder	These are the common icons used in the global objects and faceplates for all Process Objects.
HMI Tags	CSV	N/A	FTVME_PlantPAxLib_Tags_3_1_**00**.csv(1)	These tags must be imported into the FactoryTalk View ME project to support switching tabs on any Process Object faceplate.

(1) The service release number (boldfaced) can change as service revisions are created.

Display Elements



A display element (global object) is created once and can be referenced multiple times on multiple displays in an application. When changes are made to the original (base) object, the instantiated copies (reference objects) are automatically updated. Use of global objects, in conjunction with tag structures in the ControlLogix system, aid consistency and save engineering time.

Descriptions

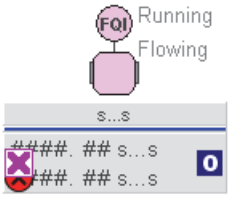
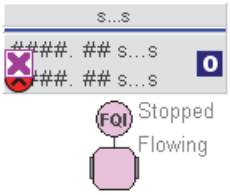
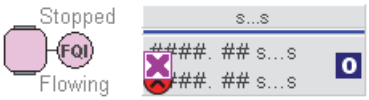
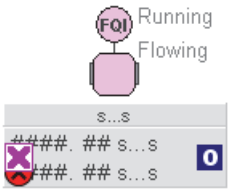
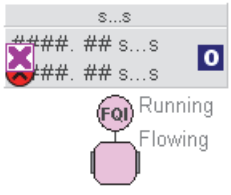
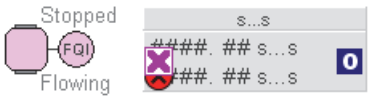
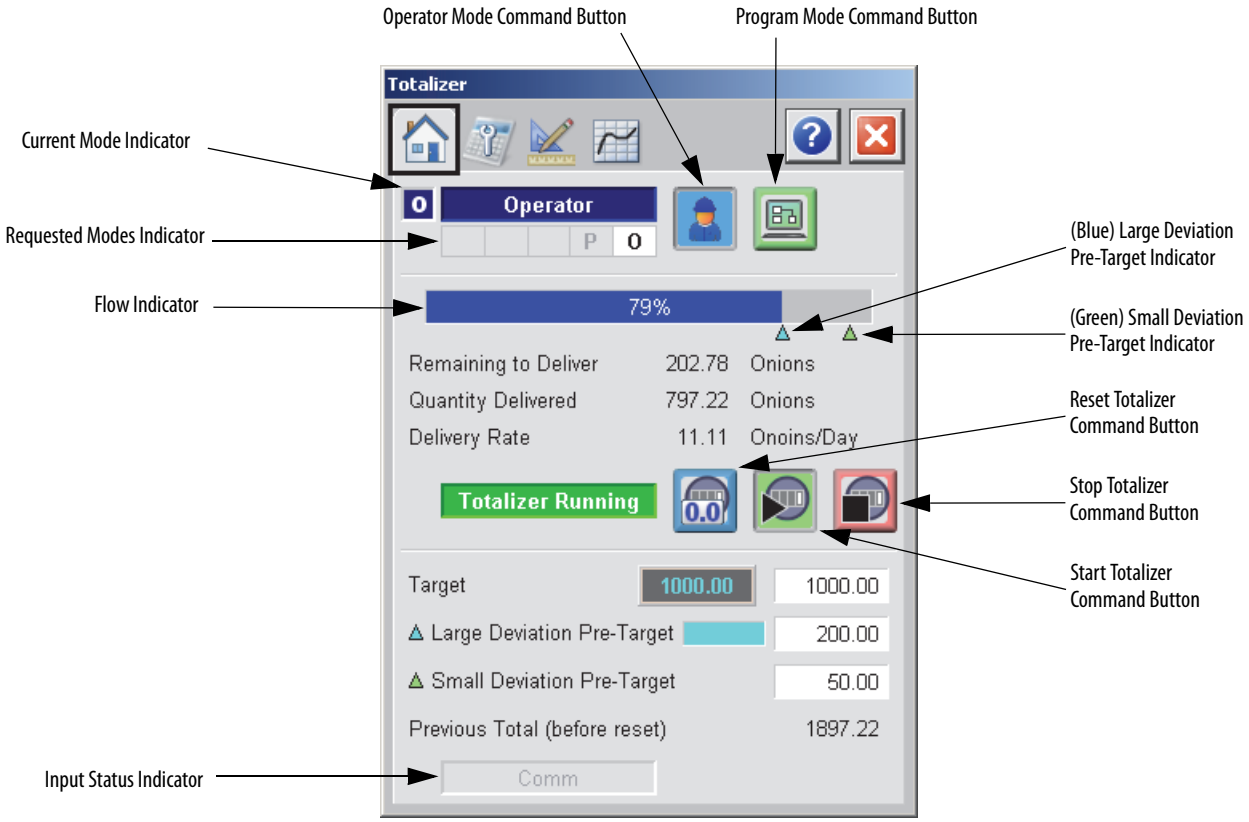
	Display Element	Description
		Totalizer with target, vertical orientation - top
GO_BuiltIn_TotalizerTgt1		Totalizer with target, vertical orientation - bottom
GO_BuiltIn_TotalizerTgt2		Totalizer with target, vertical orientation - right
GO_BuiltIn_TotalizerTgt3		Totalizer with target, vertical orientation - left
GO_BuiltIn_Totalizer		Totalizer, vertical orientation - top
GO_BuiltIn_Totalizer1		Totalizer, vertical orientation - bottom
GO_BuiltIn_Totalizer2		Totalizer, vertical orientation - right

Table 85 - Totalizer (TOT) Display Element Descriptions

Display Element Name	Display Element	Description
GO_BuiltIn_Totalizer3		Totalizer, vertical orientation - left

Operator Tab

The Faceplate initially opens to the Operator ('Home') tab. From here, an operator can monitor the device status and manually operate the device when it is in Operator mode.

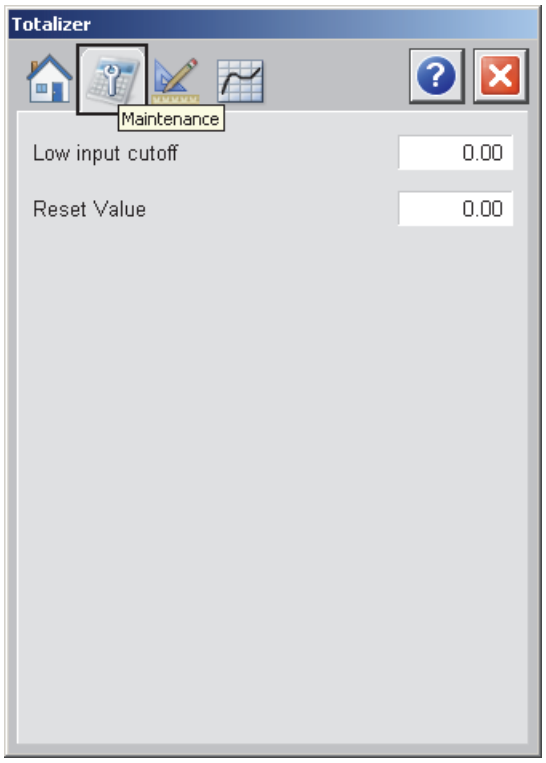


The following table lists the functions on of the TOT Operator tab.

Table 86 - TOT Operator Tab Description

Function	Action	Security
	Click to request Operator mode.	Manual Device Operation (Code B)
	Click to request Program mode.	
	Click to reset Totalizer.	Normal Operation of Devices (Code A)
	Click to start Totalizer.	
	Click to stop Totalizer.	
Target	Type the target value for the totalizer input.	
Large Deviation Pre-Target	Type a value for the large deviation pre-target value of Total compared to Target. This value is expressed as a deviation from the Target.	
Small Deviation Pre-Target	Type a value for the small deviation pre-target value of Total compared to Target. This value is expressed as a deviation from the Target.	

Maintenance Tab



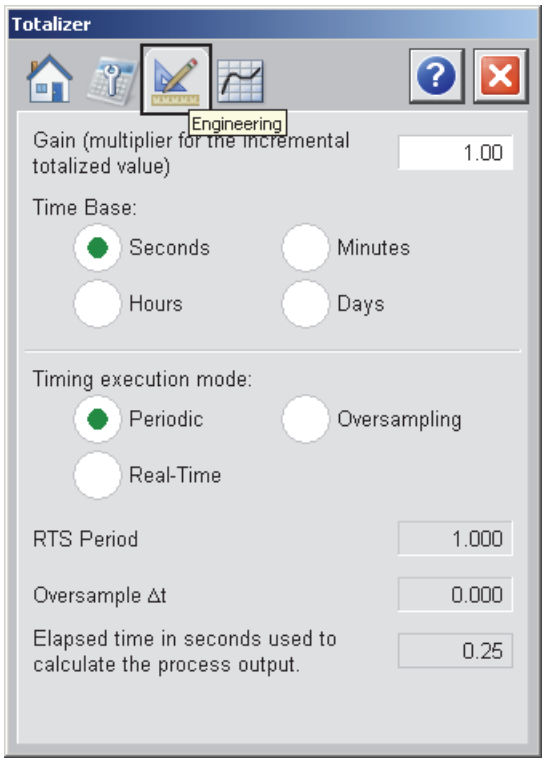
The following table shows the functions of the RMPS Maintenance tab.

Table 87 - TOT Maintenance Tab Description

Function	Action	Security	Configuration Parameters
Low input cutoff	Type a value for the low input cutoff. When the input is set at or below LowInCutoff, totalization stops.	Configuration and Tuning Maintenance (Code D)	.LowInCutoff
Reset Value	Type a value for reset input.		.ResetValue

Engineering Tab

The Engineering tab provides access to device configuration parameters and ranges, options for device and I/O setup, displayed text, and faceplate-to-faceplate navigation settings, for initial system commissioning or later system changes.



The following table shows the functions of the TOT engineering tab.

Table 88 - TOT Engineering Tab Description

Function	Action	Security	Configuration Parameters
Gain (multiplier for the incremental totalized value)	Type a value for the multiplier of the incremental totalized value.	Engineering Configuration (Code E)	.Gain
Time Base: Seconds Minutes Hours Days	Click to select the time base for the time base input.		.TimeBase
Timing execution mode: Periodic Oversampling Real-Time	Click to select the timing execution mode.	Engineering Configuration (Code E)	.TimingMode

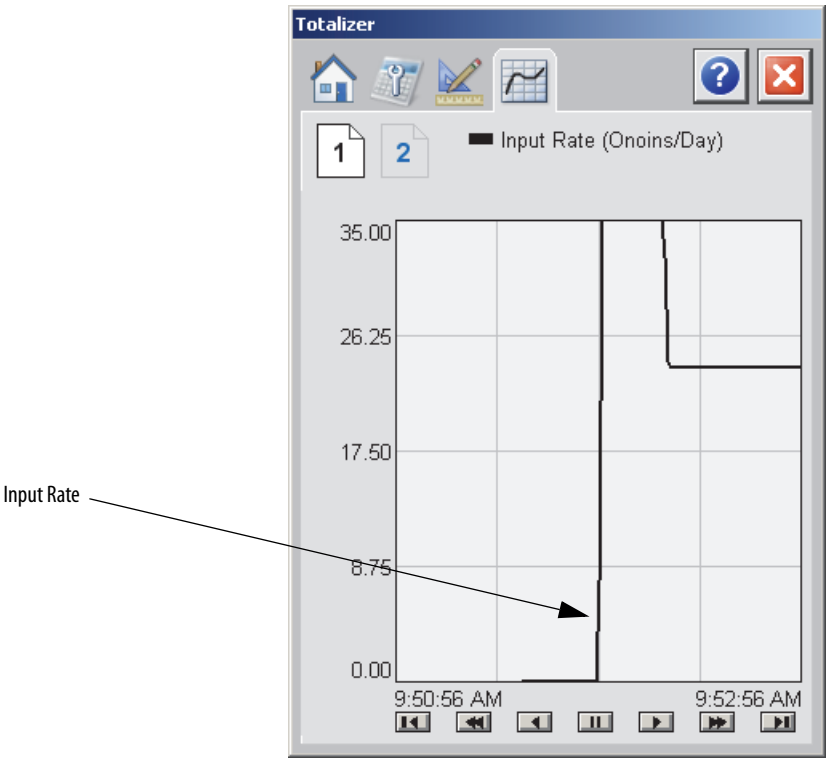
Trends Tab

The Trends tab shows trend charts of key device data over time. These faceplate trends provide a quick view of current device performance to supplement, but not replace, dedicated historical or live trend displays.

The Trends tab is divided into two pages.

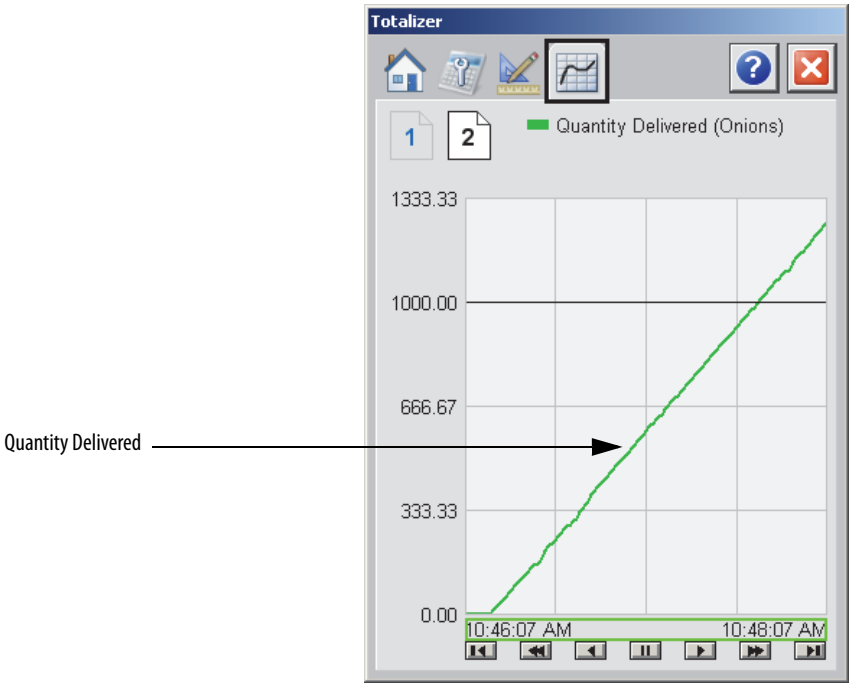
Trends Tab Page 1

Page 1 of the PIDE Trends tab shows the Input waveform.



Trends Tab Page 2

Page 2 of the Trends tab shows the quantity delivered.



Faceplate Help

The faceplate Help page shows the indicators and command buttons that are used by the Totalizer.

G

H

I

L

M

O

P

Q

R

S

T

U

V

X

Notes:

Rockwell Automation Support

Rockwell Automation provides technical information on the Web to assist you in using its products. At http://www.rockwellautomation.com/support you can find technical and application notes, sample code, and links to software service packs. You can also visit our Support Center at https://rockwellautomation.custhelp.com/ for software updates, support chats and forums, technical information, FAQs, and to sign up for product notification updates.

In addition, we offer multiple support programs for installation, configuration, and troubleshooting. For more information, contact your local distributor or Rockwell Automation representative, or visit http://www.rockwellautomation.com/services/online-phone.

Installation Assistance

If you experience a problem within the first 24 hours of installation, review the information that is contained in this manual. You can contact Customer Support for initial help in getting your product up and running.

United States or Canada	1.440.646.3434
Outside United States or Canada	Use the Worldwide Locator at http://www.rockwellautomation.com/rockwellautomation/support/overview.page, or contact your local Rockwell Automation representative.

New Product Satisfaction Return

Rockwell Automation tests all of its products to help ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning and needs to be returned, follow these procedures.

United States	Contact your distributor. You must provide a Customer Support case number (call the phone number above to obtain one) to your distributor to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for the return procedure.

Documentation Feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete this form, publication RA-DU002, available at http://www.rockwellautomation.com/literature/.

Rockwell Automation maintains current product environmental information on its website at http://www.rockwellautomation.com/rockwellautomation/about-us/sustainability-ethics/product-environmental-compliance.page.

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

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

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

Publication PROCES-RM002C-EN-P - November 2014
Supersedes Publication PROCES-RM002B-EN-P - June 2014