Burner Management Suite (BMS) Library

Release v1.00

Allen-Bradley
by ROCKWELL AUTOMATION

Reference Manual

Original Instructions

Important User Information

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Chapter 7

BMS Purge (raS_Opr_BMS_Purge)

Chapter 8

BMS Pilot (raS_Opr_BMS_Pilot)

Chapter 9

BMS Fuel (raS_Opr_BMS_Fuel)

Chapter 10

BMS 2-State Device (raS_Dvc_BMS_D2S)

Chapter 11

Implementation Checklist

Application Code Libraries Overview

Our Application Code Libraries enable you to easily develop and configure project code. The libraries contain tested, documented, and lifecycle-managed objects which can be used along with machine builder, process, and device libraries or as standalone components. Application Code Libraries from Rockwell Automation may include HMI faceplates for FactoryTalk®View ME/SE, FactoryTalk®Optix and/or Studio 5000 View Designer® software and provide a user interface that seamlessly integrates with the products. Platform support may differ between library offerings.

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

Application Code Libraries from Rockwell Automation may be downloaded from the Product Compatibility and Download Center. Search for "Library".

Application Code Manager

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

Enable more efficient project development with reusable libraries of code:

- Quickly create and deploy projects through our Application Content Libraries
- Import Rockwell provided application content libraries to expedite system development
- Build your own reusable code that can be managed and deployed across your entire enterprise
- Easily configure objects in bulk with reusable code to increase application development, no additional programming is necessary
- Consolidate content for Studio 5000 Logix Designer®, FactoryTalk® View Studio, Factory Talk® Optix, FactoryTalk® Alarms & Events, FactoryTalk® Historian to configure an object a single time and generate content for each of those software packages.

At this time, this library does not include support for Studio 5000® Application Code Manager. This functionality will be added in a future release.

Other Application Code Libraries

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

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

Item	Description
PlantPAx® Process Library	Rockwell Automation® Library of Process Objects provides application templates, Endress + Hauser library objects, Application Code Manager library objects, and tools and utilities for PlantPAx® DCS applications. Includes the following: Graphics for built-in instructions HMI images and Help files Logix diagnostic objects Process objects Control strategies Sequencer objects PlantPAx® Configuration Tools for Tags, Alarms and Historian Color Change Historian -- Asset Framework template and objects
Machine Builder Libraries	Tested, documented and life-cycle managed library objects and faceplates for use with Studio 5000® Application Code Manager for use primarily with OEM and discrete machine applications.
Common Application Libraries	Commonly used application library objects and faceplates for use with Studio 5000® Application Code Manager including basic functions like unit conversion and data collection.
Independent Cart Technology Libraries	ICT Libraries for iTRAK® and MagneMotion® including MagneMover® Lite™ and QuickStick® for Studio 5000® Application Code Manager
IO Device Library	Provides objects for Rockwell Automation® 1756, 1769, 1734, 1794, 1738, 1732E, 1719, 5032, 5069, 5094 I/O modules including pre-configured status and diagnostic faceplates
IO-Link Device Library	Provides IO-Link master and sensor objects including pre-configured status and diagnostic faceplates
Network Device Library	Provides objects for Stratix® switch and Device Level Ring network objects
Power Device Library	Provides objects for discrete, velocity, motion, and power monitor devices
Safety Device Library	Provides safety objects to interface with safety I/O
Condition Monitoring Device Library	Provides objects for Dynamix™-1444 Condition Monitoring modules used with pump applications.
Mettler Toledo Device Library	Provides objects for Mettler Toledo weigh scale modules.
Hardy Device Library	Provides objects for Hardy Process Solutions weigh scale modules.
Electrical Protection Device Library	Provides a standard to represent protection devices within your electrical distribution system

Libraries can be accessed from the
Product Compatibility and Download Center.

Software and Firmware Upgrades

When you update software or firmware revisions, we recommend that you verify the impact on performance and memory utilization before implementing the upgrade on the production system. For FactoryTalk® View, FactoryTalk® Optix, Studio 5000 View Designer® or ControlLogix® platforms, we recommend that you review the release notes and verify the impact of the upgrade on performance and memory utilization.

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

Rockwell Automation® Services and Support

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

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

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

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

Rockwell Automation® Burner Management Suite (BMS) Library

The Burner Management Suite Library is a tested, documented, and life cycle managed object library. The Burner Management library provides a configured Add-On Instruction (AOI) set and HMI faceplates for Rockwell Automation® industrial controllers. The Burner Management Objects may be used with Machine Builder, Process, and Packaged Libraries or as standalone components. Burner Management Library add-on instructions collect, process, and deliver data between hardware devices and application logic. The Burner Management Library includes Add-On Instructions (AOIs) and optional HMI Faceplates for Allen-Bradley® products including Compact GuardLogix 5380, and GuardLogix 5580 PACs.

The Burner Management System (BMS) Add-On Instruction (AOI) set provides the ability of easily implementing flame safety for a large segment of burner designs. The core pretense of this AOI set is making implementation of a single burner/single fuel system easy to implement with minimal code, while providing enough flexibility to support multi-fuel/multi-burner applications.

IMPORTANT It is the user's responsibility to determine if this AOI set is suitable for the intended application.

This AOI set has the following limitations:

- Single purge zone: All equipment under control of the AOI set must be feeding a single purge zone.
- Up to four main fuel trains: The fuel trains can be combined to feed a single burner with different fuels, feed multiple burners, or a combination of both.
- Up to four pilot fuel trains with spark igniter control: Pilot fuel trains and main fuel trains can be interrelated in several different configurations, such as 1:1 (each fuel has its own pilot) to multiple pilots are required to light off all main fuels. The pilot functionality allows for burners that have up to four spark-ignited main fuel(s).
- Optional Fuel Header: Using the same AOI as a main fuel train, control of a single fuel header is provided as an option.
- Due to specific timing requirements between the Safety AOIs that make up the system, all must reside in the same execution task so that all AOIs are scanned at the same rate in the same order every execution cycle.
- The Interface AOI does not reside in the Safety Logic so it is not required to be scanned synchronously. It does need to be scanned fast enough to prevent internal communication watchdog failures.

The BMS System is made up of 5 Safety-class AOIs, and one Standard-class AOI:

- **BMS Interface (raS_Opr_BMS_Interface)** (Standard-Class AOI): One per system. Primary command and status interface. Receives commands from external logic (Program) to start or stop the burner or receives commands from an HMI (Operator) to start, stop, or reset the burner. System level status information is also presented to external code and the HMI via this AOI. Predetermined data structures are communicated bi-directionally between the Interface and Master AOI.
- **BMS Master (raS_Opr_BMS_Master)** (Safety-Class AOI): One per system. Primary burner controller, receives requests from Interface and conveys requests to subordinate fuel/pilot AOIs once validated. Acts as a central collecting point for communication between all BMS system AOIs. Monitors all system level interlocks, providing safety trip and first-out capture for those signals directly monitored.
- **BMS Purge (raS_Opr_BMS_Purge)** (Safety-Class AOI): One per system. Monitors air flow to the burner, manages pre-ignition purge and post-firing purge timing. Monitors all purge and air flow-related interlocks, providing safety trip and first-out capture for those signals directly monitored. Predetermined data structures are communicated bi-directionally between the Purge and Master AOI.
- **BMS Pilot (raS_Opr_BMS_Pilot)** (Safety-Class AOI): Up to four pilot fuel trains can be included in a single BMS system, and at least one is required. Each pilot AOI monitors its own pilot fuel required interlock inputs and pilot flame sensor. It manages pilot ignition and flame proving and can be configured to perform a blocking valve closure-proving leakage test. Each pilot AOI supports at least one controlled device (single blocking valve), up to 5 devices (2 blocking valves, an auxiliary fuel vent or fuel recirc, an atomizing media valve, and a fuel scavenging valve). A single pilot AOI can support up to 3 spark ignitors and 3 flame sensors. Monitors all pilot fuel and ignition related interlocks, providing safety trip and first-out capture for those signals directly monitored. Predetermined data structures are communicated bi-directionally between each Pilot and Master AOI.
- **BMS Fuel (raS_Opr_BMS_Fuel)** (Safety-Class AOI): Up to four main fuel trains and one fuel header train can be included in a single BMS system, but a fuel AOI is not required for a complete system. Each fuel AOI monitors its own fuel required interlock inputs and main flame sensor. It manages main fuel ignition via requested pilot(s) and flame proving and can be configured to perform a blocking valve closure proving leakage test. Each fuel AOI supports at least one controlled device (single blocking valve), up to 5 devices (2 blocking valves, an auxiliary fuel vent or fuel recirc, an atomizing media valve, and a fuel scavenging valve). A single fuel AOI can support up to 2 flame sensors for distributed burners. Monitors all fuel related interlocks, providing safety trip and first-out capture for those signals directly monitored. When used as a Fuel Header, this AOI disables atomizing and scavenging and flame sensor inputs. Predetermined data structures are communicated bi-directionally between each Fuel and Master AOI.
- **BMS 2-State Device (raS_Dvc_BMS_D2S)** **(**Safety-Class AOI): Individual device control, designed to interface directly with the Master, Pilot, and Fuel AOIs. Applicable to devices that energize to activate, and de-energize for safe state. This AOI can be configured to monitor energized position feedback and de-energized position feedback, with timing adjustments to monitor for errant behavior. Under normal operating conditions there is no ability to interact with individual devices from external code or HMI, although extensive diagnostic information is available for monitoring. When the burner is not in operation, a System Test mode can be requested by the interface, enabling access to direct manipulation of each device individually for testing purposes only. The system is designed so this level of access is granted to only one device at a time. The device interacts with its host

AOI via an active heartbeat to ensure it is properly connected and executing.

The Burner Management Suite (BMS) Library may be downloaded from the Product Compatibility and Download Center. Search for Burner Management Suite (BMS) Library.

Compatibility

Compatible Software

- Studio 5000 Logix Designer® for PAC Application Development v36
- FactoryTalk® View Studio v14 and later for FactoryTalk® View Site Edition Application Development

Compatible Hardware

- GuardLogix® 5580 controller or Compact GuardLogix™ 5380 controller with v36 firmware
- FLEX 5000™ I/O modules
- 5069 Compact 5000™ I/O modules
- POINT Guard I/O Safety Modules
- 1756 ControlLogix® Digital Safety I/O modules

Certification

This section provides details on the certification for the BMS Add on instructions. Based on the assessment by the certified body, the Burner Management Suite is in compliance with IEC 61508 and meets the requirements specified Systematic Capability (SC) Level 3.

IMPORTANT IEC 61508 Certification applied exclusively to Studio 5000 Logix Designer® version 36.

See http://www.rockwellautomation.com/global/certification/overview.page for Declarations of Conformity, Certificates, and other certification details.

Summary of Changes

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

Topic	Page
Initial release	all

Footprint

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

- **Definition:** Estimated memory required to store the object definition, including all dependents
- **Instance:** Estimated memory required per object instantiated.

- **Execution:** Estimated execution time / scan footprint evaluated in 5069-L3100ERMS2

Object Footprint

Device Object	Standard Definition (blocks)	Safety Definition	Instance (blocks)	Execution (µs)
raS_Opr_BMS_Interface	8,796	N/A	736	10
raS_Opr_BMS_Master	15,216	107,576	1,600	83
raS_Opr_BMS_Purge	4,408	30,852	1,132	13
raS_Opr_BMS_Fuel	5,376	46,944	1,716	20
raS_Dvc_BMS_D2S	2,144	16,100	878	6
raS_Opr_BMS_Pilot	5,552	50,188	1,824	23

Safety Signatures

The Burner Management (BMS) AOIs are validated and certified and released with specific versions of the Logix Designer (refer to the Release Notes of the Burner Management (BMS) AOI Suite. Release Notes may be downloaded from the Product Compatibility and Download Center. The AOI Safety Signatures for each AOI for each software release (Vxx) exist to ensure the integrity of the AOI functionality. Users can use the Safety AOI Signature as part of their certification process to ensure proper GuardLogix configuration. Please refer to the Release Notes for the provided safety signatures.

Additional resources

For a complete understanding of Rockwell Automation® safety applications it is recommended to review the following documents.

Resource	Description
GuardLogix® Safety Application Instruction Set, publication 1756-RM095	This manual describes the Rockwell Automation® GuardLogix® Safety Application Instruction Set (e.g. DCS, CROUT, TSSM, TSAM, etc), which is type-approved and certified for safety-related function in applications up to and including Safety Integrity Level (SIL) 3 according to IEC61508, and Performance Level, PLe (Cat.4), according to ISO13849-1.
Rockwell Automation® Safety Function Documents	Collection of safety function examples provided to minimize risk and speed up development. Examples include set-up and wiring, bill of materials, wiring schemes, configuration, verification and validation plan, and calculation of performance level.
ControlLogix 5580 and GuardLogix® 5580 Controllers User Manual, publication 1756-UM543	Provides information on how to install, configure, and program ControlLogix and GuardLogix® 5580 controllers in the Logix Designer application.
CompactLogix™ 5380 and Compact GuardLogix® 5380 User Manual, publication 5069-UM001.	Provides information on how to install, configure, and program CompactLogix and Compact GuardLogix® 5380 controllers in the Logix Designer application.
GuardLogix® 5570 Controllers User Manual, publication 1756-UM022.	Provides information on how to install, configure, and program the GuardLogix® 5570 controllers in the Logix Designer application.
GuardLogix® 5580 and Compact GuardLogix® 5380 Controller Systems Safety Reference, publication 1756-RM012.	Describes the development, operation, and maintenance of GuardLogix® 5580 and Compact GuardLogix® 5380 controller-based safety systems in the Logix Designer application.
EtherNet/IP Network Devices User Manual, publication ENET-UM006	Describes how to configure and use EtherNet/IP devices to communicate on the EtherNet/IP network.
System Security Design Guidelines Reference Manual, publication SECURE-RM001	Provides guidance on how to conduct security assessments, implement Rockwell Automation products in a secure system, harden the control system, manage user access, and dispose of equipment.
Logix 5000 Controllers Common Procedures Programming Manual, publication 1756-PM001	Provides information on programming Logix 5000 controllers, including how to manage project files, organize tags, program and test routines, and handle faults.

Resource	Description
Logix 5000 Controllers I/O and Tag Data, publication 1756-PM004	Provides information on tag types, tag scoping, external tag access, and tag aliases.
Logix 5000 Controllers Add-On Instructions Programming Manual, publication 1756-PM010	Provides information on how to create and use standard and safety Add-On Instructions in Logix applications.
Logix 5000 Controllers Program Parameters, publication 1756-PM021	Provides information on linking tag data between program scoped tags, including definitions of the different parameter types and recommended usage of each.
Logix 5000 Controllers General Instructions Reference Manual, publication 1756-RM003	Provides information on the Logix 5000 instruction set that includes general, motion, and process instructions.
GuardLogix Safety Application Instruction Set Reference Manual, publication 1756-RM095	Provides information on the GuardLogix Safety Application instruction set.

Library Components

The Burner Management System (BMS) Library is a tested, documented, and life cycle managed object library. The Burner Management library provides a configured Add-On Instruction (AOI) set and HMI faceplates for Rockwell Automation® industrial controllers. The Burner Management Objects may be used with Machine Builder, Process, and Packaged Libraries or as standalone components. Burner Management Library add-on instructions collect, process, and deliver data between hardware devices and application logic. The Burner Management Library includes Add-On Instructions (AOIs) and optional HMI Faceplates for Allen-Bradley® products including Compact GuardLogix 5380, and GuardLogix 5580 PACs.

BMS Instructions

The Burner Management Suite Library includes the following safety-class instructions:

- Burner Master
- Purge Control
- Pilot Control
- Main Fuel/Header Control
- Device Control

The Burner Management Suite Library the following standard-class instructions:

- Interface

There are two types of instructions:

- Device (Dvc): instruction used for devices. (e.g. D2S)
- Operation (Opr): instructions used for operation or applied code such as sequencing, operational execution, and general application functions. (e.g. Purge)

All objects in this library include the prefix “raS” implying they belong to Rockwell Automation Safety applications.

BMS Instructions

Instruction	Version	Instruction Description	Common BMS Applications
raS_Opr_BMS_Interface	1.00	Interface	Standard class interface to safety class instructions for control & HMI.
raS_Opr_BMS_Master	1.00	Burner Master	Primary burner controller
raS_Opr_BMS_Purge	1.00	Purge Control	Burner airflow and purge control
raS_Opr_BMS_Fuel	1.00	Main Fuel/Header Control	Main fuel valve train interlocks and control
raS_Dvc_BMS_D2S	1.00	Device Control	Device interface with master, pilot and fuel
raS_Opr_BMS_Pilot	1.00	Pilot Control	Pilot Fuel valve train interlocks and control

Library Folders and Files

When you extract the library from the downloaded .zip folder, you will find the following folder and file structure. Note that some items are generalized with *TYPE* (e.g. Dvc, Opr, Tec), *OBJECT* (e.g. Interface, Master, etc) and *Class* (Std = Standard, Sfty = Safety). The major and minor versions are represented by X and Y respectively

Level 1	Level 2	Level 3	File Type	Description
Application Example			**Folder**	**Application Example Files**
	BMSApplication.ACD		ACD	Logix Designer Example Project
	BMSApplication_SE.apa		APA	FT View SE Project Archive
HMI - FactoryTalk View SE			**Folder**	**FactoryTalk® View SE Files**
	Displays - gfx		Folder	FT View SE display files
		(raS-X_YY-SE) raS_*TYPE_OBJECT*-faceplate.gfx	GFX	Object Faceplate display
	Global Objects - ggfx		Folder	FT View SE Global Object files
		(raS-X-SE) Graphic Symbols - LIBRARY.ggfx	GGFX	Graphic Symbol/Launch Button global objects
		(raS-X-SE) Toolbox - LIBRARY.ggfx	GGFX	Toolbox global objects
HMI FactoryTalk View Images - png			**Folder**	**FT View ME/SE image files**
	images.png		PNG	FTView ME/SE images
Reference Manuals			**Folder**	**Manuals**
	BMS-RM001X-EN-P.pdf		PDF	Reference manual
Studio 5000 Logix Designer Files - L5X			**Folder**	**Studio 5000® AOI and RUNG import files**
	raS_*TYPE_BMS_OBJECT*_X.YY_*Class*RUNG.L5X		L5X	Object rung import
	raS_*TYPE_BMS_OBJECT*_X.YY_*Class*ROUTINE.L5X		L5X	Object routine import
	raS_*TYPE_OBJECT*_X.YY_AOI.L5X		L5X	Object AOI import

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

Visualization Files

Each Add-On Instruction has associated visualization files that provide a common user interface. The BMS Library supports FactoryTalk® View SE (Site Edition) as an HMI platform.

FactoryTalk® View Visualization Files

You must import these files in the following order:

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

File Type Abbreviations	FactoryTalk® View SE	Description
Images (.png)	All .png files in the *HMI FactoryTalk View Images - png* folder. **IMPORTANT:** FactoryTalk® View application renames PNG files when they are imported with a .bmp file extension, but the files retain a .png format.	Common icons that are used in the Global Objects and standard displays for all objects.
Global objects (.ggfx)	(raS-1-SE) Graphic Symbols - BMS.ggfx	Graphic symbols or launch buttons used to open faceplate displays from other displays.
	(raS-1-SE) Toolbox - BMS.ggfx	Common objects used across multiple device faceplates.
Standard displays (.gfx)	(raS-1_XX-SE) precedes name of the display.	e.g. (raS-1_01-SE) raS_Dvc_BMS_D2S-Faceplate.gfx

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

Global objects serve two purposes:

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

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

Basic Faceplate Attributes

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

Faceplate Navigation

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


Active Tab


Inactive Tab

The common tabs are shown below.


Home Tab


Configuration Tab


Interlocks Tab

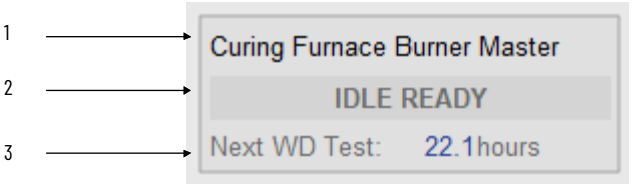

Permissives Tab

Diagnostics Tab


Help

Basic Launch Button Attributes

Common global object graphics are provided for the BMS Master, Purge, Pilot, and Fuel objects. Users can choose to use either a version including a navigation button to launch the related faceplate, or a non-navigation version. The D2S objects have unique graphic symbols which will be explained in that section of the manual.

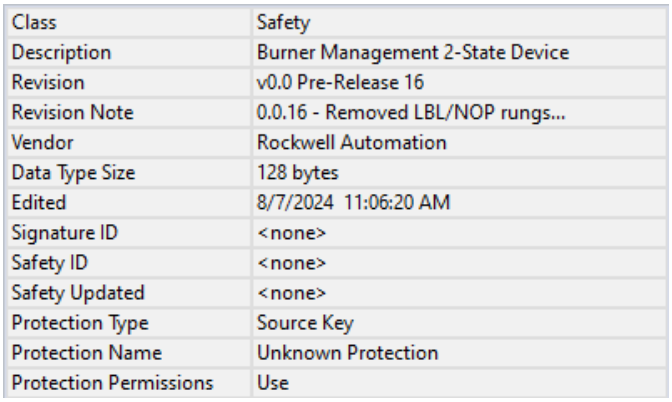
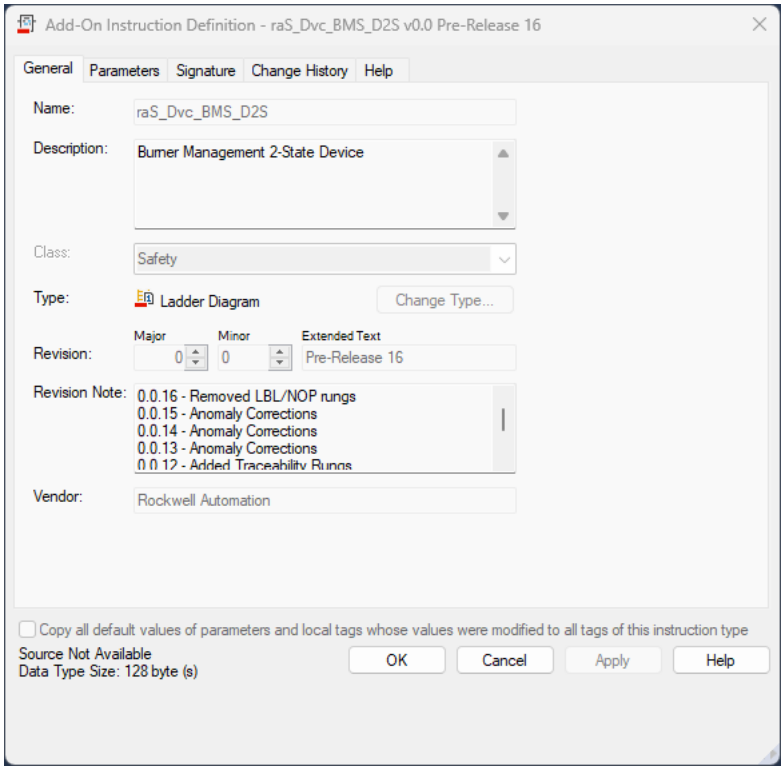
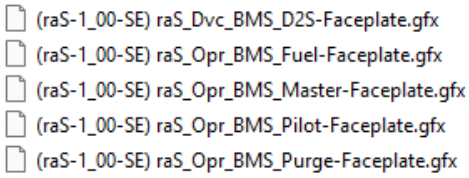


Item	Description
1	Tag information label (Inf_Label.@Description)
2	Operating State
3	Time to next external watchdog relay test (Master) or Time Remaining (Purge, Pilot, Fuel)

The Add-On Instruction parameter Inf_Label.@Description is used to assign a string to the button labels.

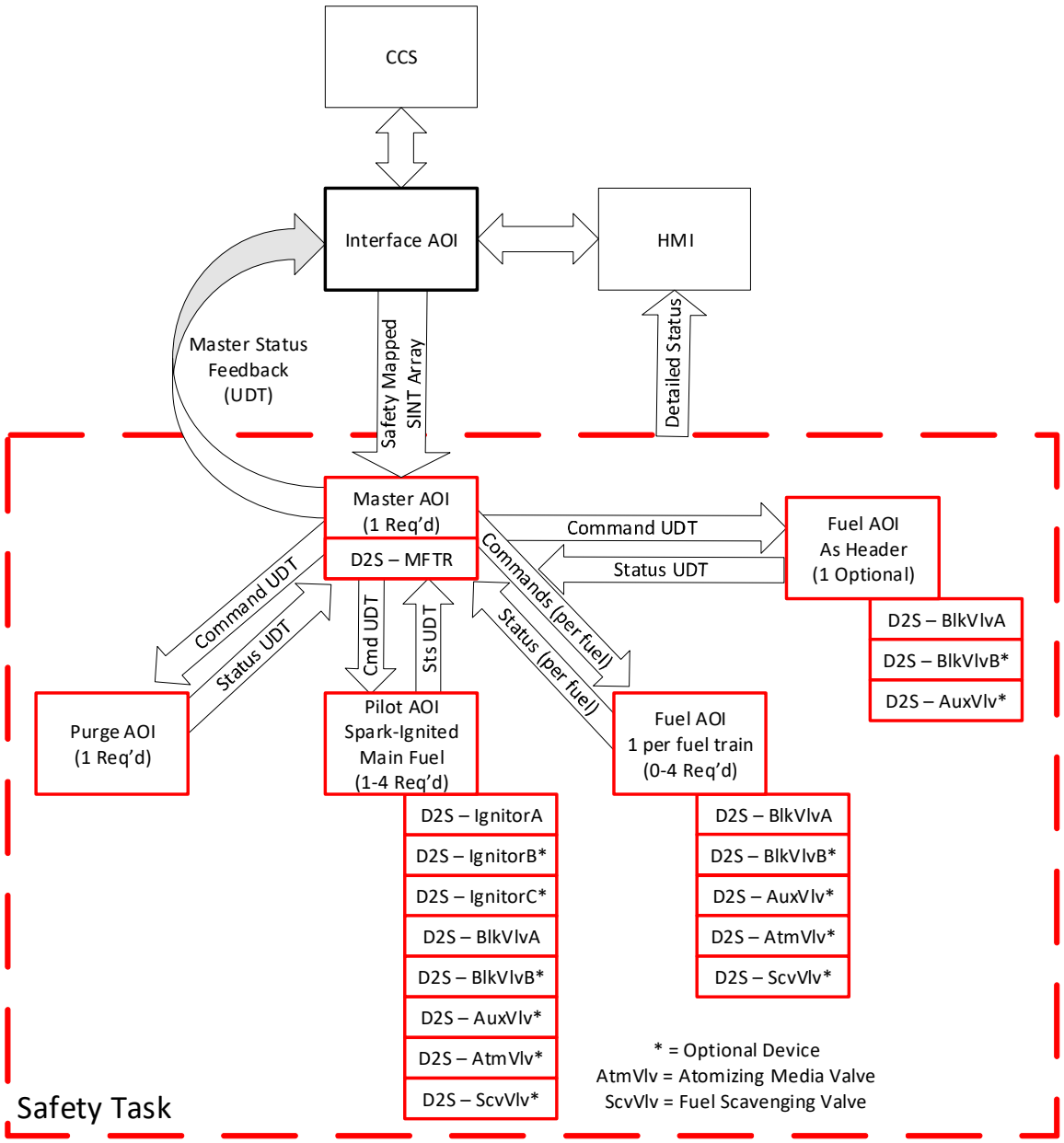
Library Versions

Each library object has a revision x.yy where: x is the Major Revision number and yy is the Minor Revision number. Each release of the library comes with release notes that describe the changes that were made since the last release. You can find the revision number of the object in a number of locations as shown in the following table.

Component	Example
The Add-On Instruction in Logix Designer application has revision information visible when the instruction is selected in the Controller Organizer.	
The Add-On Instruction Definition General tab shows the revision number along with basic revision notes. Refer to the release notes for complete revision notes.	
The revision number is shown in the file names for GFX, AOI.L5X, and RUNG.L5X files.	

Architectural Overview

The BMS inter-AOI communications and architecture is described in the following diagram.



Common Add-On Instruction Features

The following features apply to all Add-On Instructions (AOI) in the BMS Library:

- Required Boolean inputs provide immediate visibility of all safety device input conditions.
- Configuration validation.
- Interlocks - Each AOI provides functionally required interlocks, selectable interlocks based on configuration, and user-definable interlocks. All enabled interlock conditions must be resolved to allow burner functionality. Interlocks will generate a safety trip as appropriate if lost.
- Permissives - Each AOI provides functionally required permissive conditions, selectable permissives based on configuration, and user-

definable permissives. All enabled permissive conditions must be resolved to allow each AOI to start its assigned function but have no further effect once the AOI goes active.

- Safety Trip first-out identification and capture generated by lost interlock condition(s), requiring manual reset once triggered.
- Four operating modes:
  - Normal Mode - AOIs used must be in this mode to operate the burner. An out-of-service fuel in a multi-fuel application can be allowed to be Disabled as long as at least one fuel is available to run the burner.
  - TEST ENABLE Mode - Non-operating maintenance mode. The burner cannot operate in this mode. Individual devices can be selected for manual testing one at a time.
  - Test Active Mode - Non-operating maintenance mode. The burner cannot operate in this mode. One device is actively being tested for proper function. Once a single device is under test, no additional devices can be tested until the device under test is released and the system returns to TEST ENABLE Mode.
  - Disabled - The AOI is not in service. All devices under a Disabled AOI's control are deenergized and not available for testing.
- Bidirectional communication with the Master AOI.
  - Single UDT tag at Master with substructures for data to and from each component AOI.
  - Each component AOI has a Data To Master and Data From Master connection using its own substructures within the same tag used by the Master.
- Heartbeat and watchdog monitor for communication with Master AOI.
  - Heartbeat is monitored on both ends, generating a fault and interlock loss if heartbeat fails.

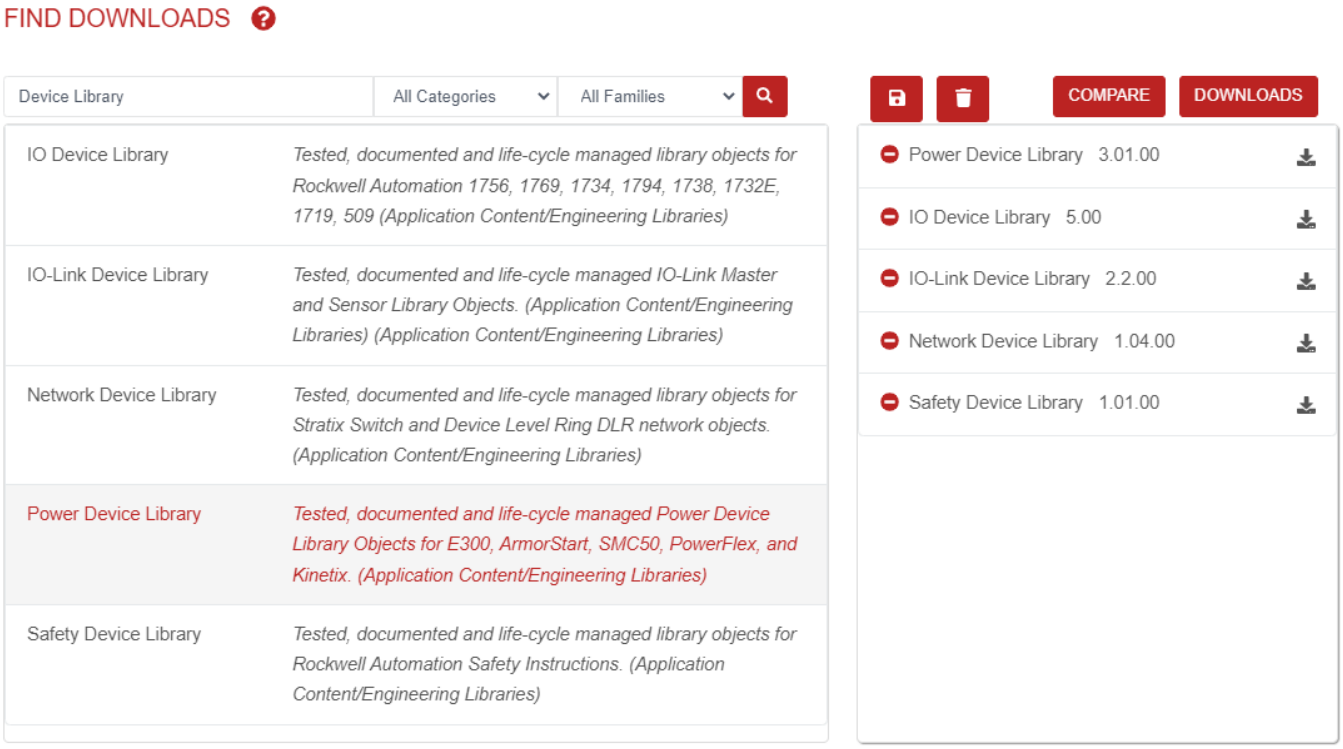
Using the Library

Install the Library

Download the Library

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

Search “Burner Management Suite (BMS) Library” or filter on Application Content to quickly find the library.



Importing Logic into Studio 5000 Projects

There are multiple methods to using the logic in a Studio 5000® Application. You can import the RUNG.L5X or ROUTINE.L5X files into your program and configure them manually.

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

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

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

Tool/Workflow	Description of when to use	Software Requirements
Import ROUTINE.L5X or RUNG.L5X File	Application Code Manager is not installed. Application has already been developed but some Device Objects need to be added. Familiar with rung import workflow.	Studio 5000 Logix Designer®
Import AOI.L5X File	Updating existing application that contains an older version of a Device Object AOI.	Studio 5000 Logix Designer®

Import Classes

The provided RUNG.L5X and ROUTINE.L5X files include a class identified:

- Sfty: Safety class rung/routine to be imported into the Safety class task.
- Std: Standard class rung/routine to be imported into any Standard class task.

Safety Class Imports

Import File	Description	Type
raS_Dvc_BMS_D2S_1.00_SftyRUNG.L5X	D2S Device	Rung import
raS_Dvc_BMS_D2S_w0Mon_1.00_SftyRUNG.L5X	D2S Device with Output Monitor (using DCM to monitor feedback from interposing relay)	Rung import
raS_Opr_BMS_Fuel_1.00_SftyROUTINE.L5X	Fuel Control	Routine import
raS_Opr_BMS_MasterPurge_1.00_SftyROUTINE.L5X	Master and Purge Control	Routine import
raS_Opr_BMS_Pilot_1.00_SftyROUTINE.L5X	Pilot Control	Routine import

Standard Class Imports

Import File	Description	Type
raS_Opr_BMS_Alarms_1.00_StdROUTINE.L5X	Safety Type Alarm Logic	Routine import
raS_Opr_BMS_Interface_1.00_StdROUTINE.L5X	BMS User Interface	Routine import

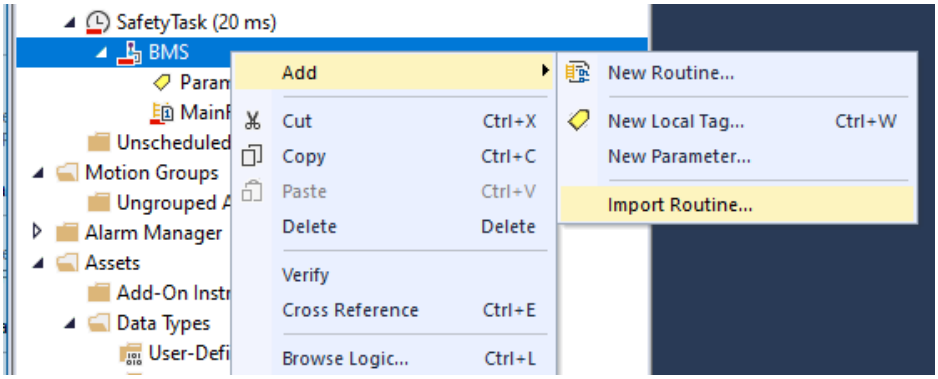
BMS Routine Import Example

The typical BMS implementation will require two programs, one in the Safety Task holding the BMS Safety AOIs and the other in a Standard Task to manage the user interface and alarms. Routine export files have been provided in L5X format to facilitate easy implementation of both programs. To use pre-engineered logic, import each desired ROUTINE L5X into the appropriate program within your controller project. Note that each ROUTINE L5X is identified as to whether it is used in the standard program or in the safety program. It is recommended that the Safety Program is named “BMS” to facilitate successful import of standard routines due to passing of Program Parameters between the programs. Once all desired routines are imported, the safety program can be renamed.

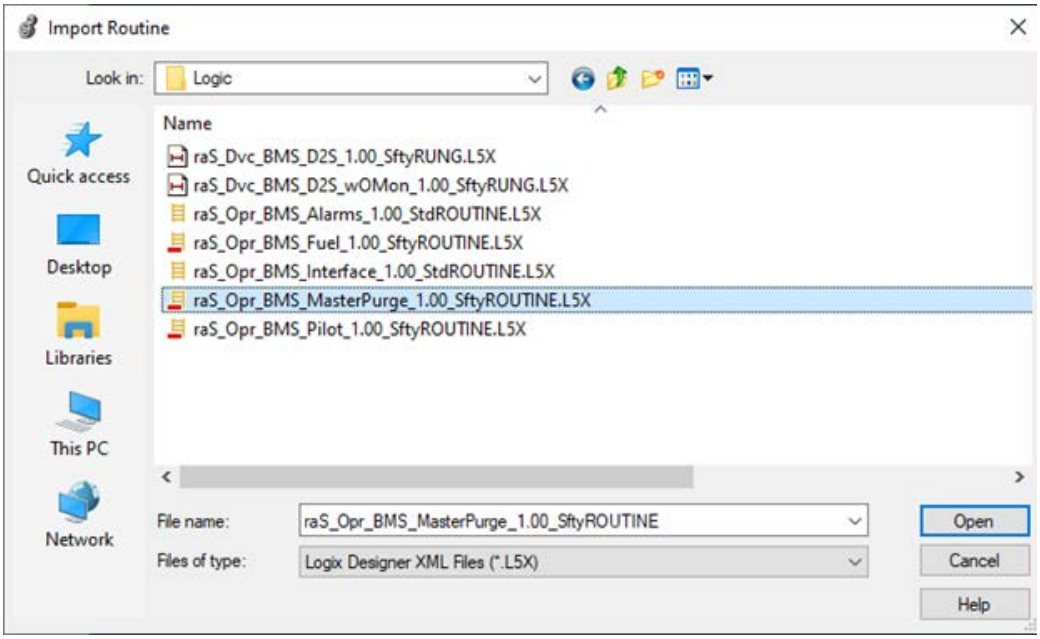
IMPORTANT Add-On Instruction definitions can be imported, but not updated, online.

Import BMS Program Routines

1. Import routines for the BMS Safety Program first. Every system requires one Master/Purge routine and one Pilot routine. Larger systems may have multiple Pilot routines and one or more Fuel routines. Note that each Pilot or Fuel routine needs a unique name, as do the tags each import creates.
2. Right click on the BMS Program, choose Add... then select Import Routine

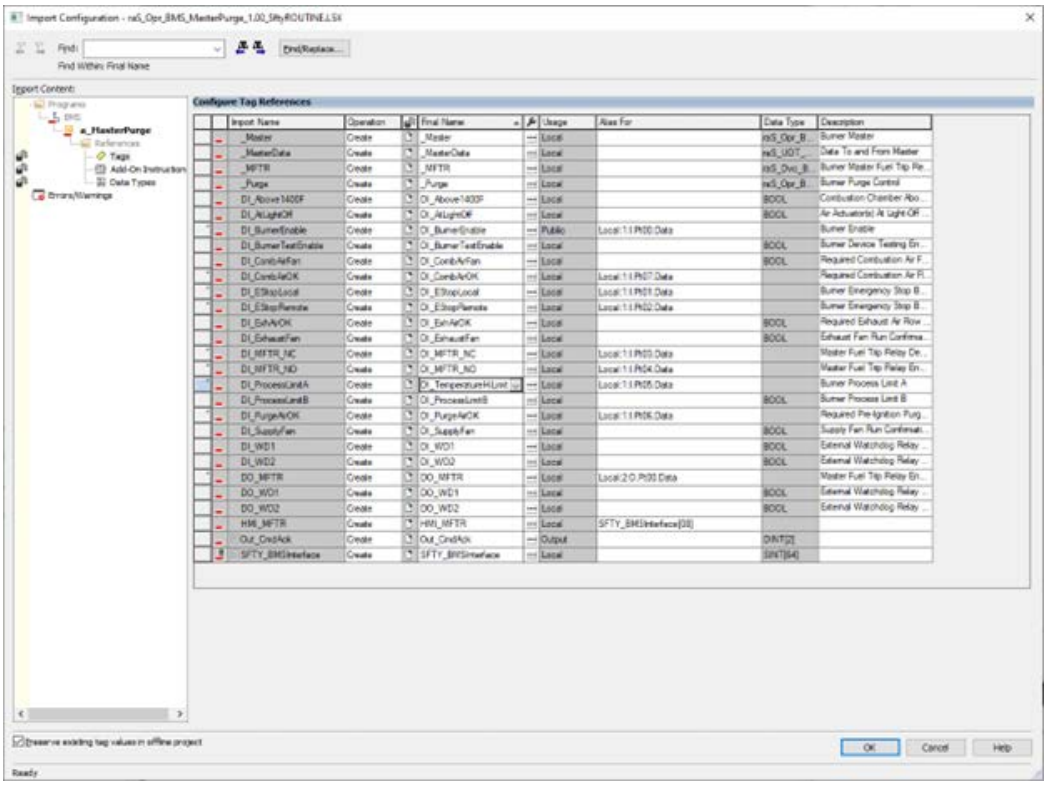


3. Select the Master/Purge ROUTINE import file.

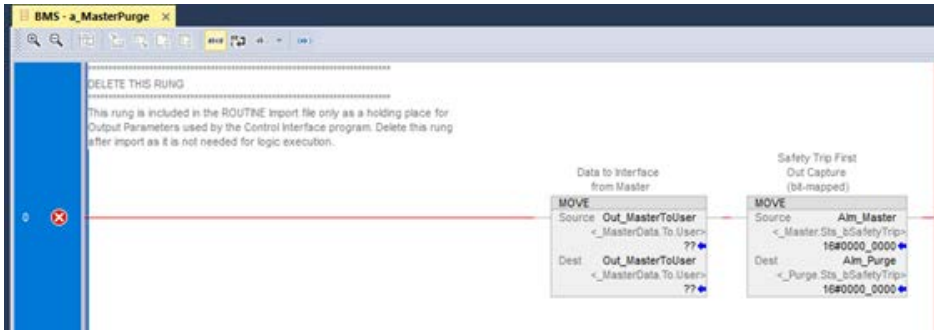


4. An Import Configuration dialogue window will open and display tag names contained within the import. Because all tags are generated at Program Scope, tags supporting the Master and Purge AOIs can remain as named in the import.
5. I/O tags, those prefixed with DI and DO, can be renamed as desired to better identify the connected device. The I/O tags can also be aliased to their Safety I/O channels by placing the Safety Input or Output data

point in the Alias for column. Only Safety I/O modules can be used for BMS Safety tag aliases.

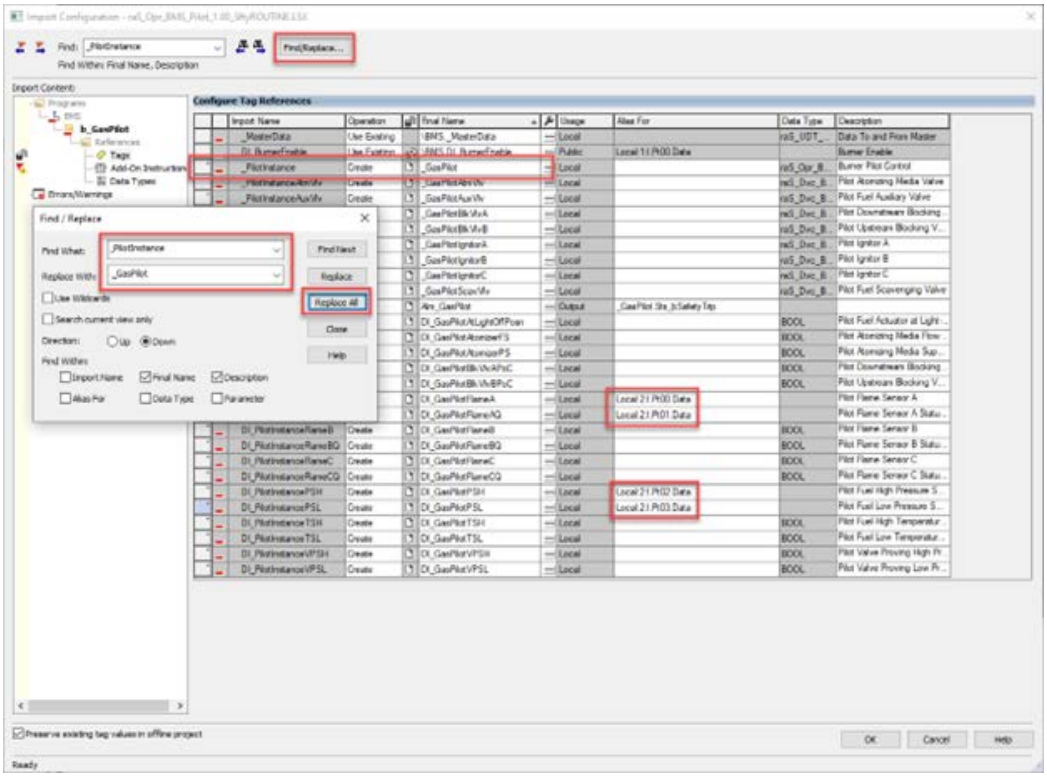


6. After importing, open the imported routine and delete the first rung. This rung is present only as a holding place for Output Parameters that need to be created to pass data to the BurnerControl program in the standard task.

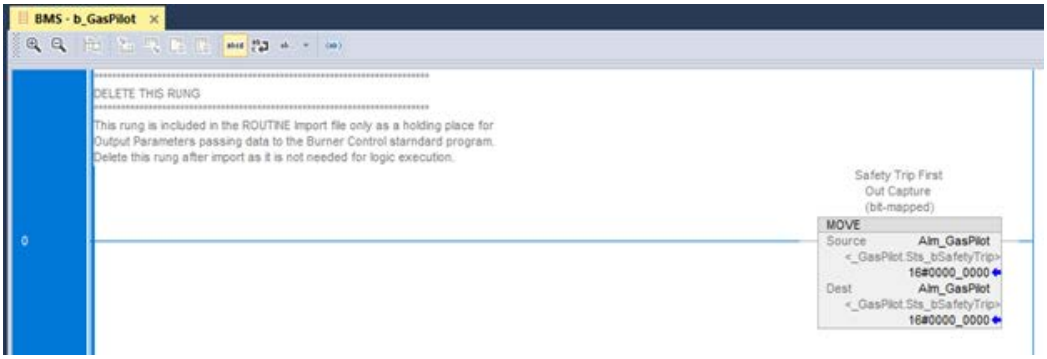


7. Import a Pilot routine for your burner application. Use the Find/ Replace feature to give the Pilot Instance and its tags unique names. As

with the Master/Purge import, I/O tags can be aliased to their physical channel data here.



8. After importing, open the imported routine and delete the first rung. This rung is present only as a holding place for Output Parameters that need to be created to pass data to the BurnerControl program in the standard task.

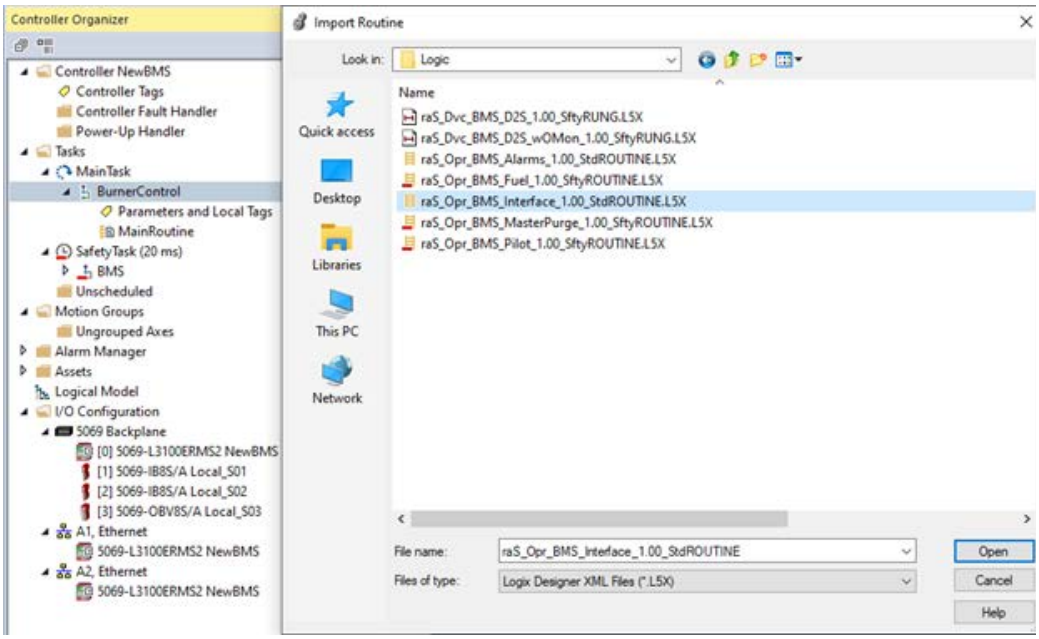


9. Repeat for remaining Pilot and Fuel instances needed for your burner. Ensure each instance has a unique routine name and all tags imported are unique to that instance.
10. Add a JSR in the MainRoutine for each routine created.

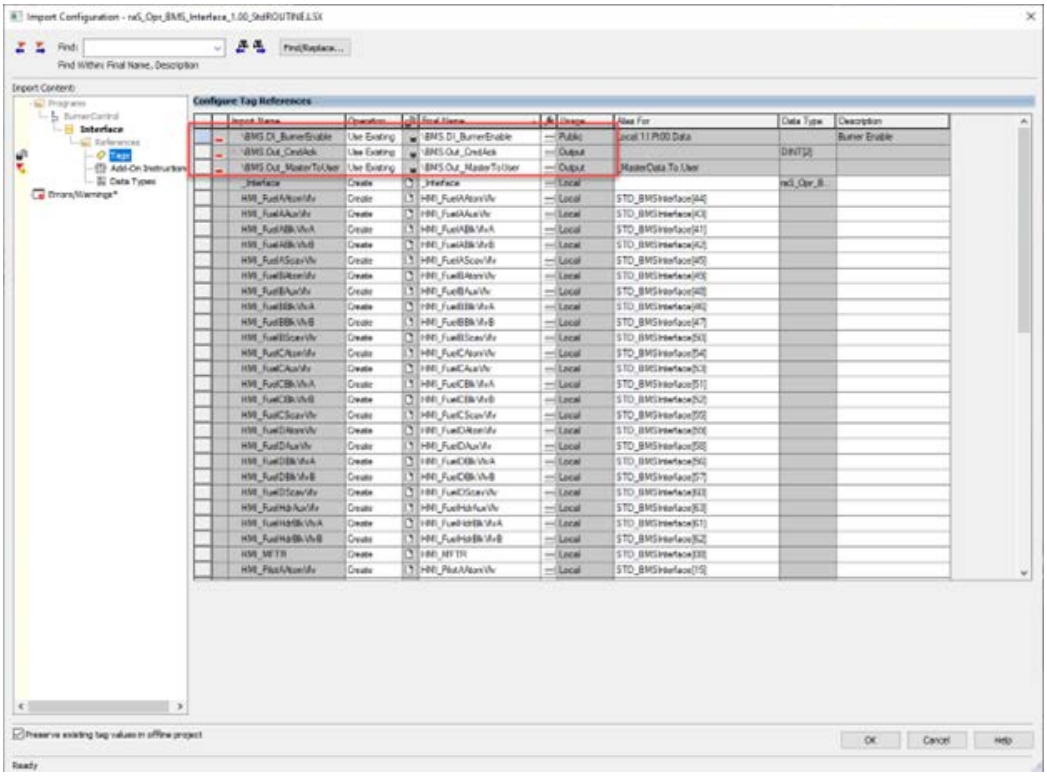
IMPORTANT This Library is intended to cover single burner applications and some multi-burner applications. Care must be taken in multi-burner applications to ensure proper Master AOI configuration, specifically that Cfg_AnyFlameOK setting is False as this setting is intended to support Single Burner applications only.

Import Burner Control Interface Routines

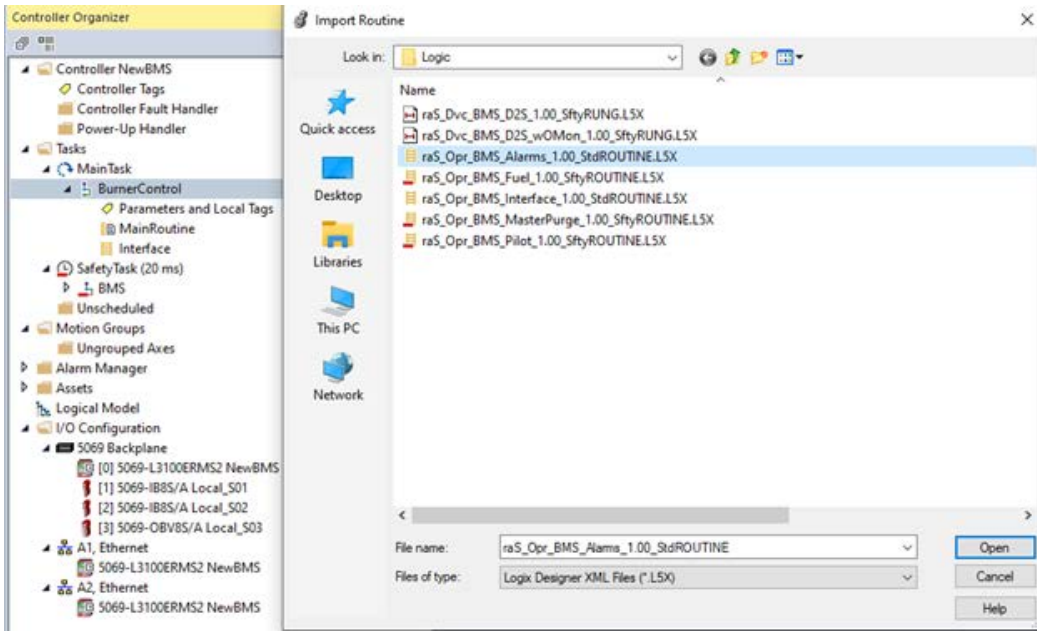
1. Import Interface routine into the standard BurnerControl program using the same techniques described above.



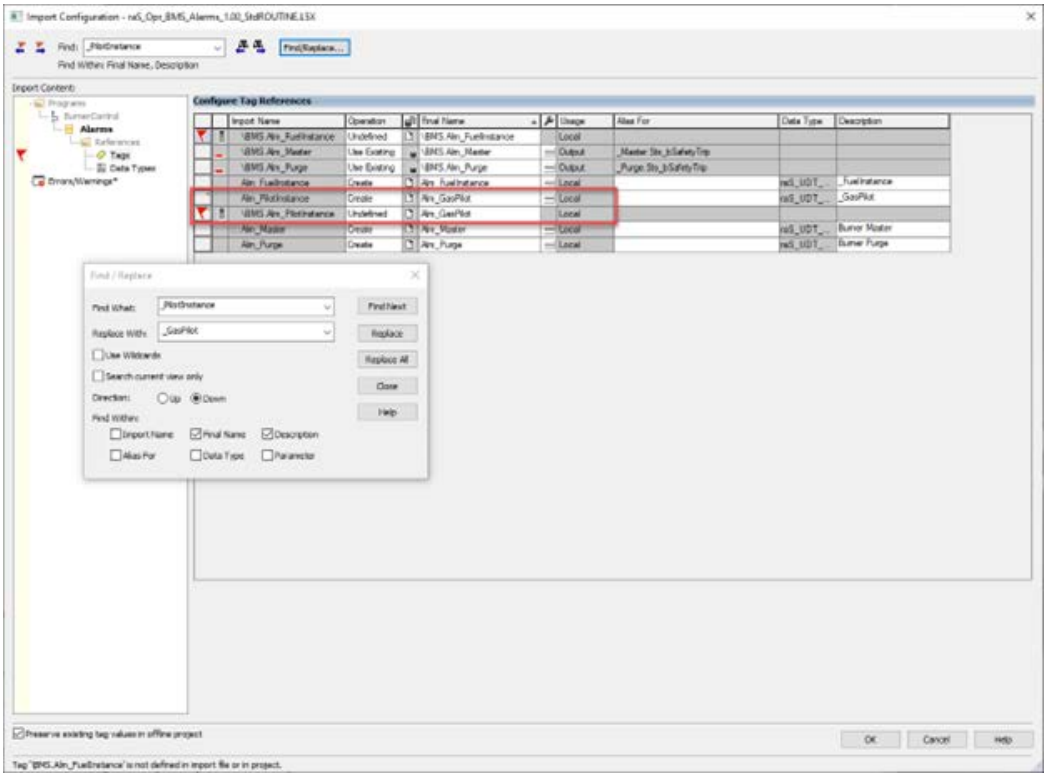
2. Ensure the Output Parameter references are valid. If not, ensure the burner management safety program is named “BMS” as this is the name used in the import file. The program name can be changed once both routine imports are complete.



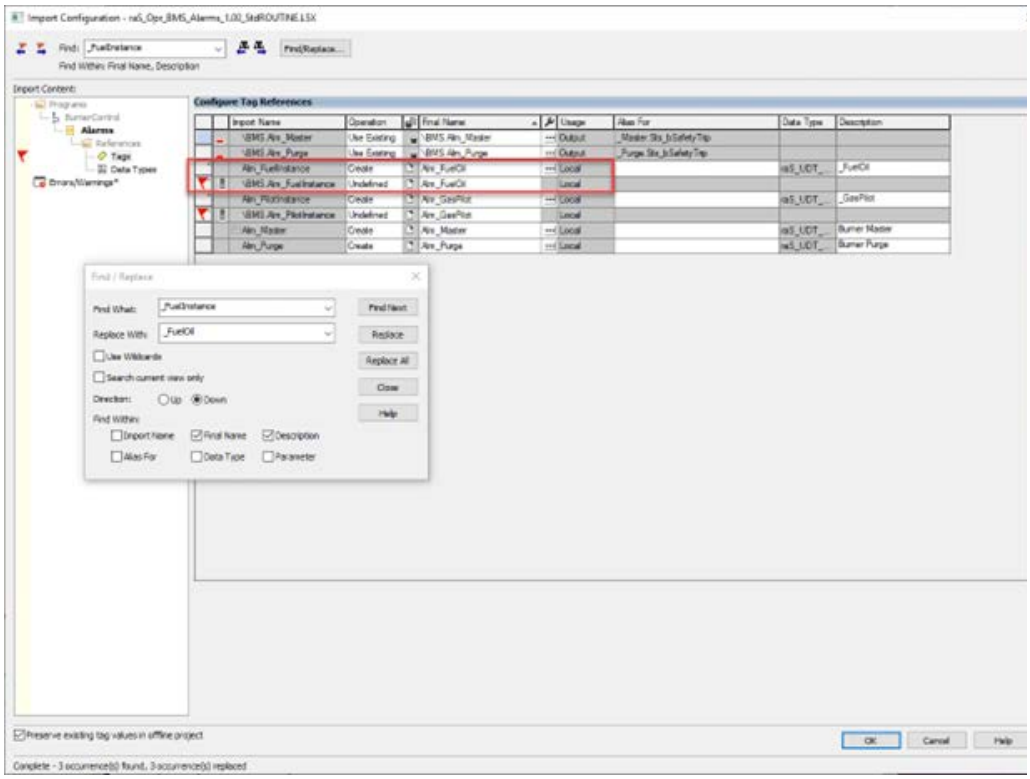
3. Import the Alarms routine into the standard BurnerControl program.



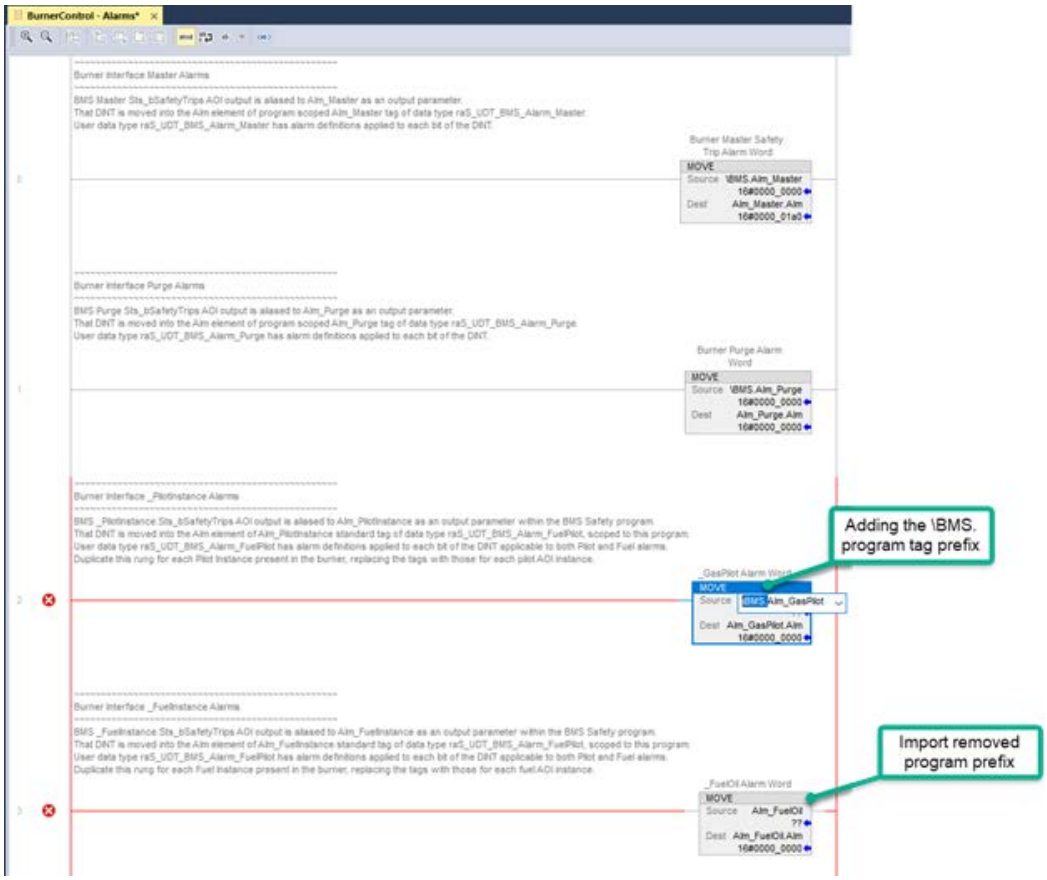
4. Update the alarm tags for the Pilot Instance previously created. Note that editing the Output Parameter reference removes the program name prefix. This will be added post-import.



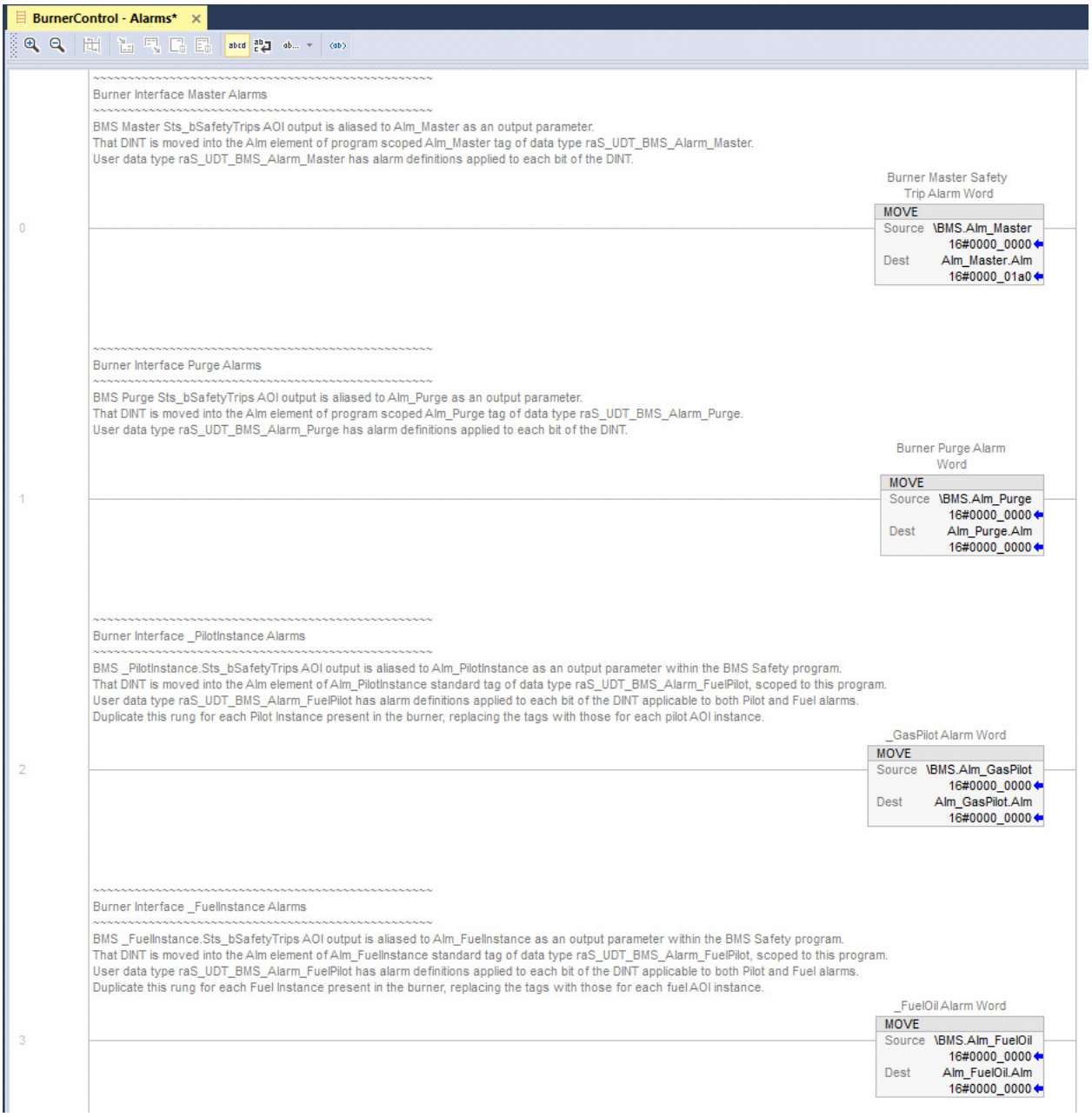
5. Update the alarm tags for the Fuel Instance previously created (_FuelOil in this case). Note that editing the Output Parameter reference removes the program name prefix. This will be added post-import.



6. Add the \BMS. Program reference as a prefix to the MOVE source tags for Fuel and Pilot.



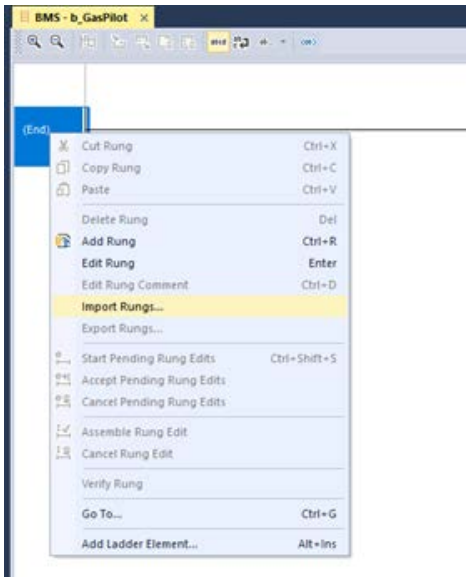
7. Replicate rungs and update tags for each additional Pilot or Fuel instance created in the BMS safety program. Routine after adding prefixes:



8. Add a JSR in the MainRoutine for each routine created.

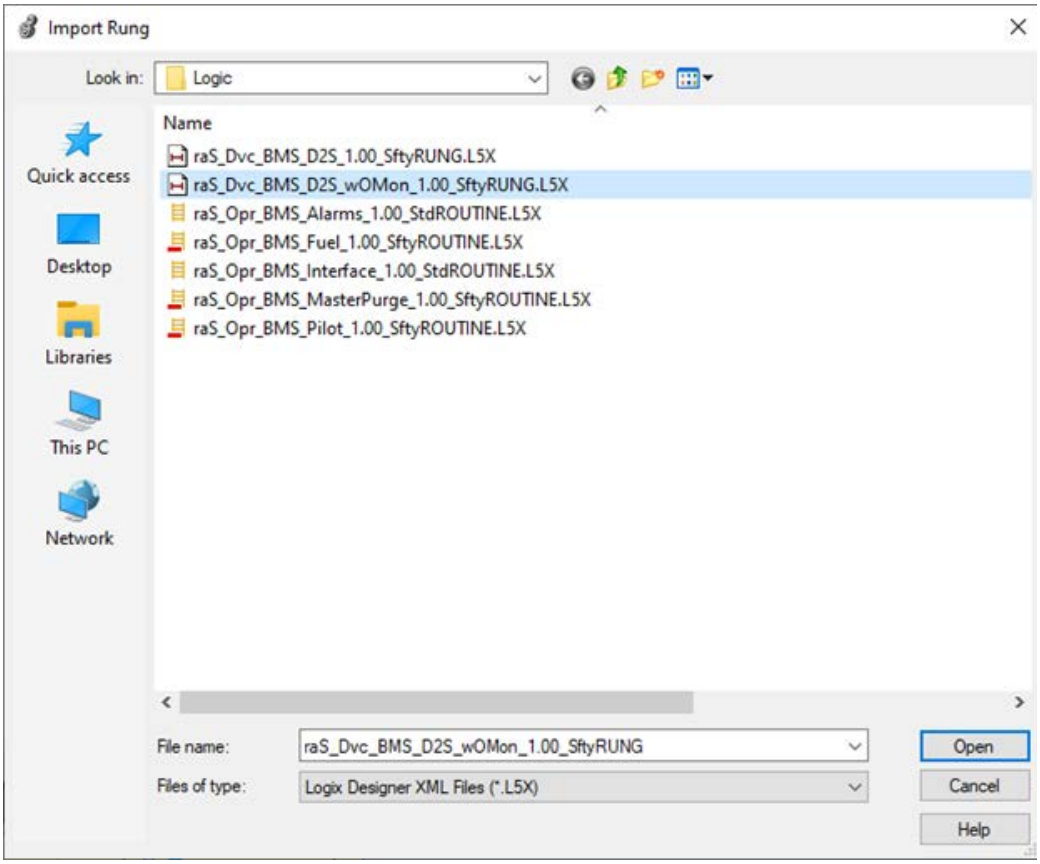
BMS Rung Import Example

1. Open the imported Pilot routine and import a RUNG L5X for each device controlled by the Pilot.

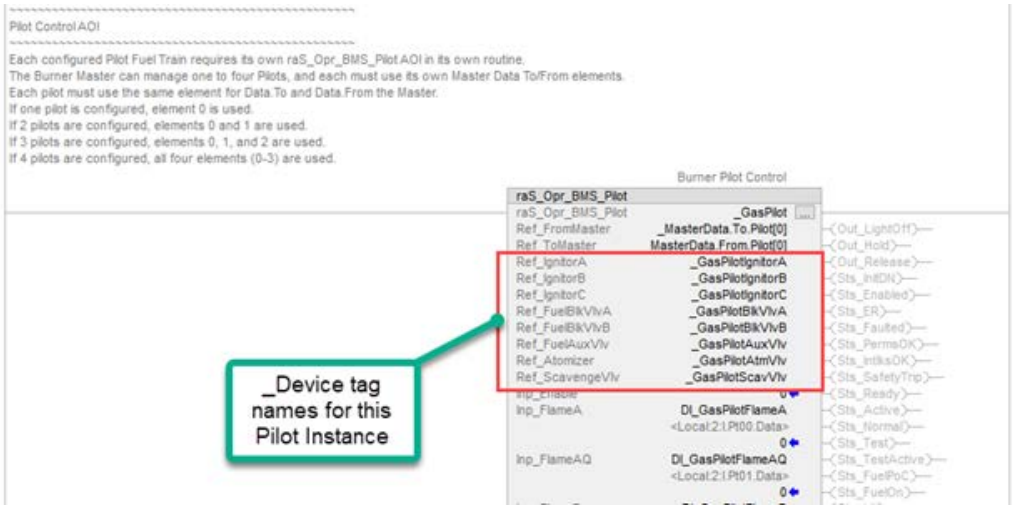


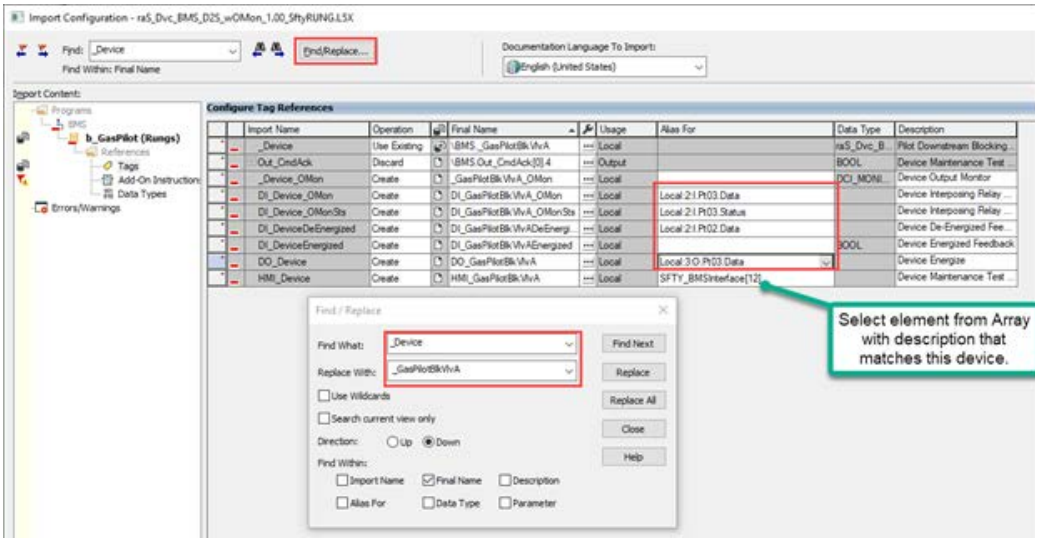
2. Two rung logic options are available: one as a bare raS_Dvc_BMS_D2S AOI and another with the D2S AOI and a DCM instruction to monitor

interposing relay health. The rung including the DCM instruction is imported here.

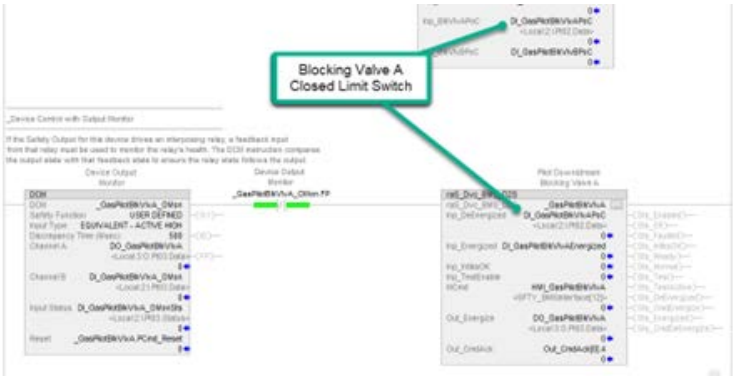


3. Substitute the generic "_Device" name with the actual device tag name as imported with the Pilot AOI.





4. Device control rung with output monitor after import:



5. Repeat for remaining devices

BMS Alarm Logic Example

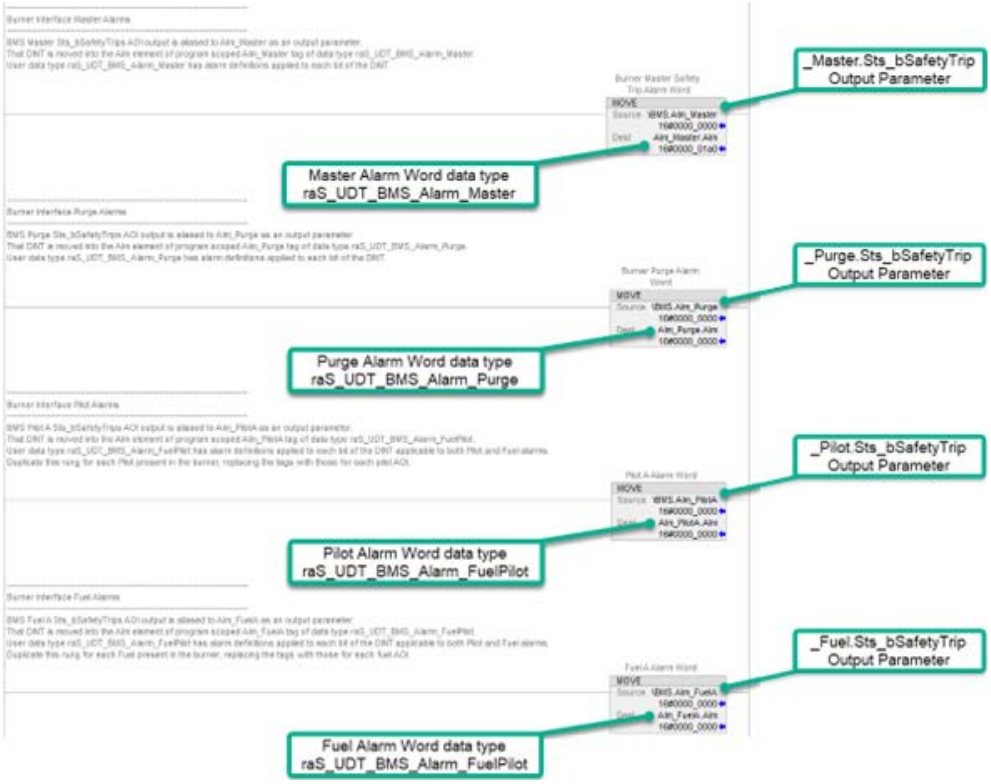
The Burner Management Suite Library includes a routine export and UDTs supporting Safety Trip alarms generated by raS_Opr_BMS_Master, raS_Opr_BMS_Purge, raS_Opr_BMS_Pilot, and raS_Opr_BMS_Fuel AOIs. This example demonstrates how to implement safety trip alarms for a simple system consisting of one Fuel and one Pilot, along with the required Master and Purge. The logic and tags shown are all that is necessary to generate and manage these alarms when employing FTView SE, PanelView 5000, or FactoryTalk Optix. Other HMI platforms may require additional logic to generate or manage alarms.

Three User-defined Data Types (UDTs) are provided with the library that mirror the Sts_bSafetyTrip output parameter for the four AOIs mentioned above. Each UDT also contains Logix Alarm Definitions for each trip condition such that a standard tag created using one of the UDTs also creates all alarms associated with the safety trips for that AOI. Note that the alarms generated by Pilot Safety Trips and Fuel Safety Trips are so similar that alarms for both AOIs can be covered by a single UDT.

1. Output Parameters within the BMS Safety Program for each AOI producing Safety Trip Alarms, shown below, are created by aliasing to respective Sts_bSafetyTrip parameters.

Alm_Master	Output	_Master.Sts_bSafetyTrip	_Master.Sts_bSafetyTrip	DINT	Safety
Alm_Purge	Output	_Purge.Sts_bSafetyTrip	_Purge.Sts_bSafetyTrip	DINT	Safety
Alm_PilotA	Output	_Pilot.Sts_bSafetyTrip	_Pilot.Sts_bSafetyTrip	DINT	Safety
Alm_FuelA	Output	_Fuel.Sts_bSafetyTrip	_Fuel.Sts_bSafetyTrip	DINT	Safety

2. Logic in the BurnerControl Standard Program receives data from the above output parameters, moving that data into alarm tags created using the library's alarm UDTs.

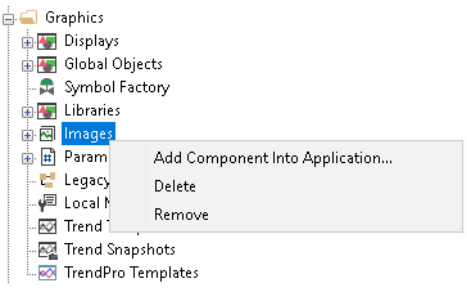


Using FactoryTalk® View Studio

Import FactoryTalk® View Visualization Files

There are several components to import for the visualization files. You import files from the downloaded Rockwell Automation® library files via FactoryTalk® View SE.

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



Import files in this order:

1. Import HMI Images files.

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

2. Import Global Object files

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

3. Import HMI Faceplates

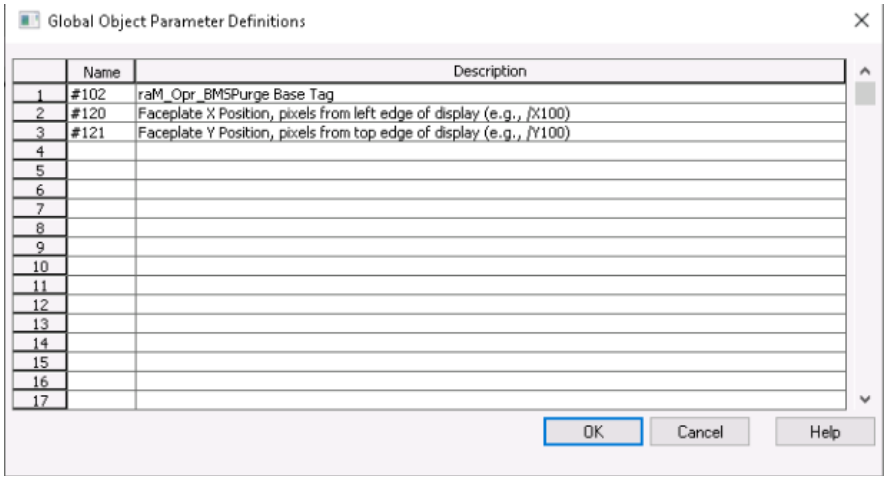
Select the faceplate (.gfx) files from the *HMI - FactoryTalk View SE\Displays - gfx* folder

Configuring FactoryTalk® View Objects

Once the files have been imported into the FactoryTalk® View Studio project, you can begin using them in your application. Open the *(raS-1-SE) Graphic Symbols - BMS.ggfx*. Copy the desired launch button style and paste it into a display in your application where you would like to open the faceplate. For more information on graphic symbols, refer to the Graphic Symbols section of the specific object type chapter in this manual.

To configure the graphic symbol launch button, right-click and select *Global Object Parameter Values*. The Global Object Parameter values for the AOI Instance (#102) is mandatory. You can browse for the tag in your controller project by clicking '...' or manually type them in. These parameters may vary depending on the graphic symbol used, please refer to the Graphic Symbols section of the device type for detailed information.

Navigation graphic symbols also include parameters #120 and #121 to define the faceplate x/y co-ordinates. This allows you to specify the location within the screen where the faceplate is opened.



Library Upgrades

Add-On Instruction Upgrades

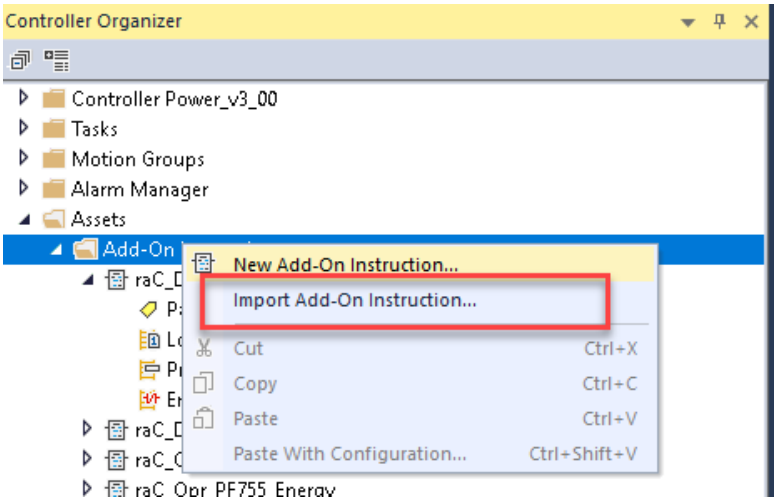
There are two methods to upgrading existing device object add-on instructions in a project. You can do this by importing individual add-on instruction AOI.L5X files.

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

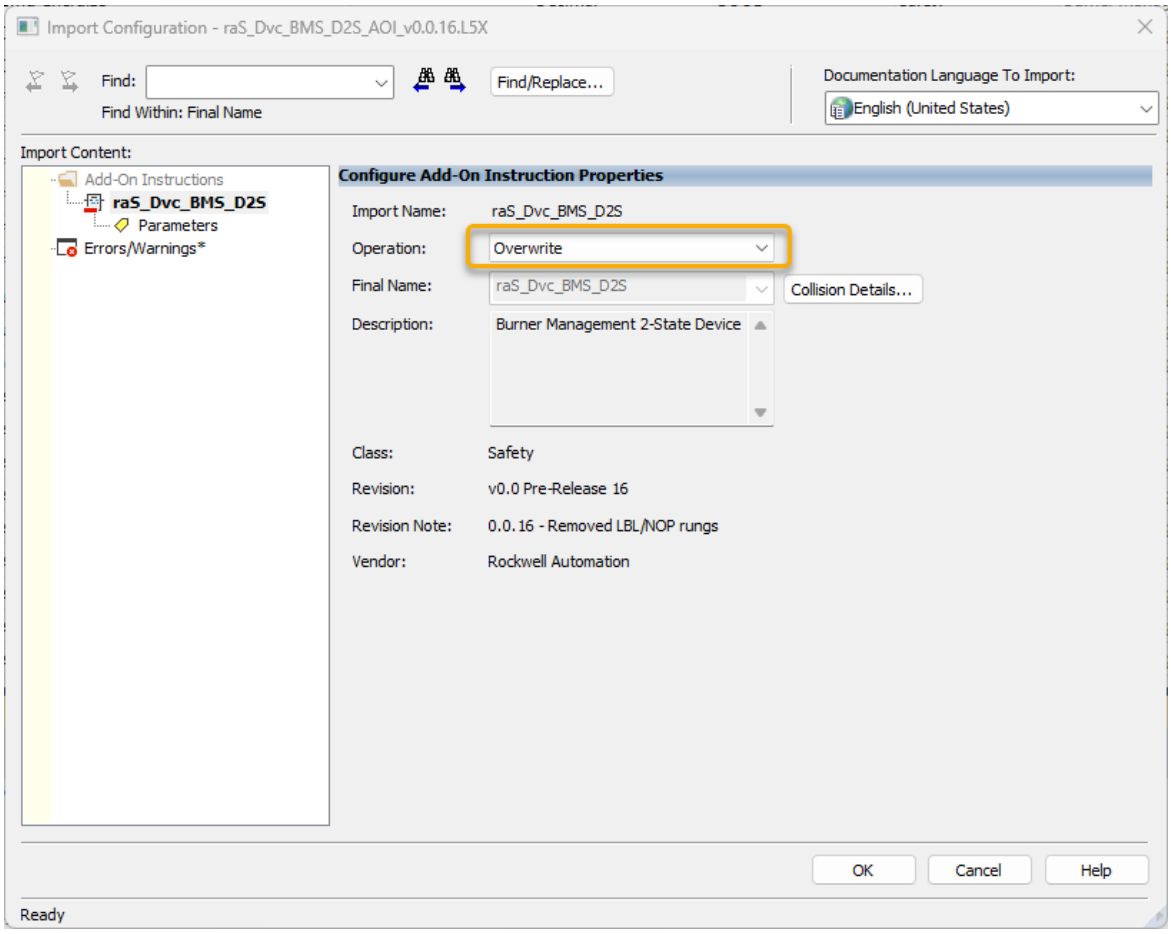
Upgrades by Importing AOI.L5X Files

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

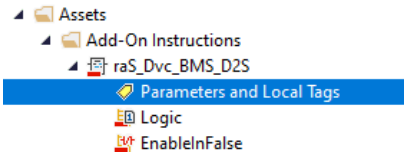
- Open the controller file. Note changes must be done offline.
- In the *Controller Organizer* pane right-click on *Assets* > *Add-On Instructions* and select *Import Add-On Instruction*. Navigate to the AOI.L5X file in the *Studio 5000 Logix Designer® Files - L5X* and Open.



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

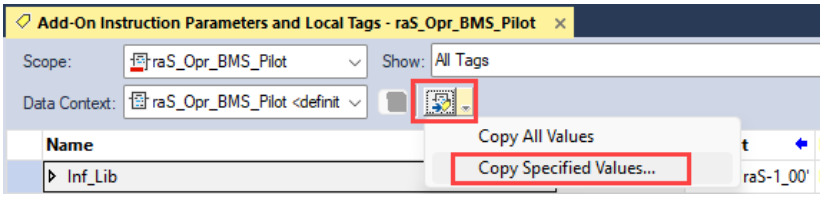


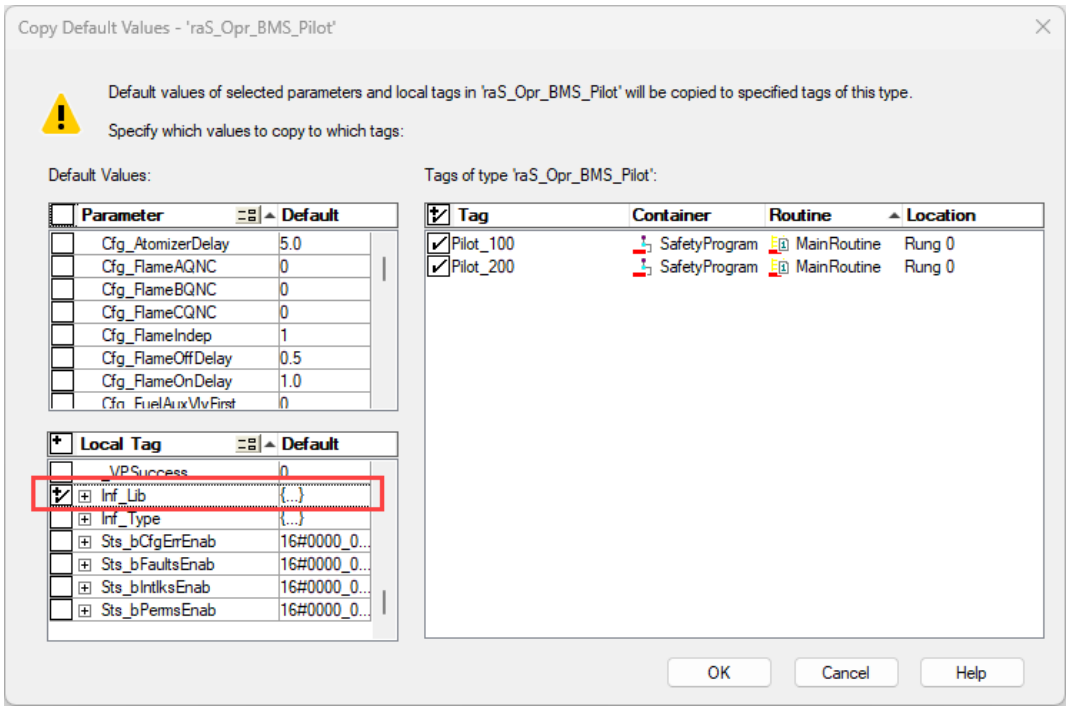
- In order to ensure the HMI faceplate still works properly you mayneed to update the object's library information stored in the Inf_Lib tag. In the *Controller Organizer* pane under *Assets > Add-On Instructions* expand the object that was updated. Double-click on *Parameters and Local Tags* to open up the instructions tags.



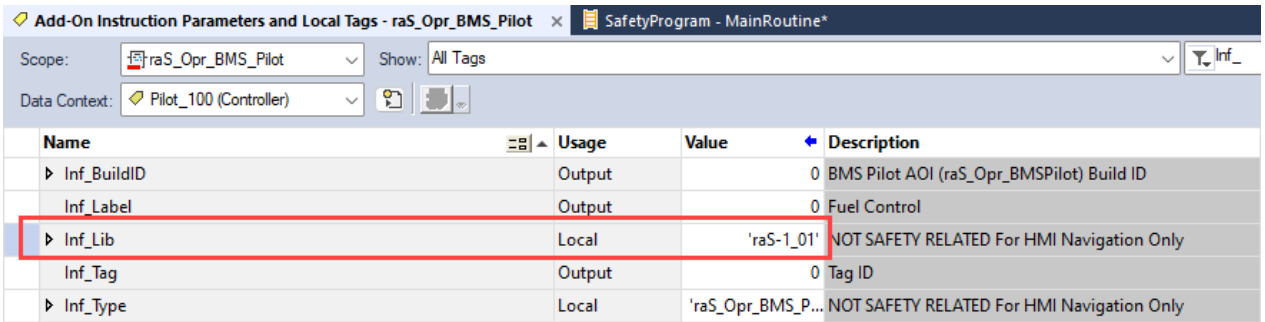
- In the *Add-On Instruction Parameters and Local Tags* window, you may notice that the Inf_Lib tag in the add-on instruction definition matches the new library revision number. Click on the down-arrow to the right of the copy button and select *Copy Specified Values...*

•





- In the *Copy Default Values* window, be sure to **first uncheck all Parameters and Local Tags** by clicking the +/- box in the top right. Failure to do so may result in overwriting settings in the existing objects.
- Check only *Inf_Lib* in the *Local Tag* area. On the right, all affected objects should be selected. Click OK.
- You can now confirm that the *Inf_Lib* tag has been updated to the current library (e.g. 'raS-1_01') by changing the *Data Context* drop-down to a specific object.



FactoryTalk® View Upgrades

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

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

Using the BMS Library with Other Application Code Libraries

Application Code Libraries

The Safety Device Library can be used alongside other Application Code Libraries.

You may consider using the Safety Device Library to apply AOIs and HMI faceplates to interface with other safety-class instructions available in GuardLogix and Compact GuardLogix PACs.

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

PlantPAx® Process Object Library objects may be used within the same FactoryTalk® View SE HMI application; however it should be noted that the Process Object Library v5.x is only compatible with Process class Logix controllers (1756-L8xEP,5069-L3x0ERP) and the BMS Library is only compatible with safety-class controllers (1756-L8xES, 5069-L3xxERSx) therefor these must be implemented in separate controllers.

BMS Interface (raS_Opr_BMS_ Interface)

Overview

Standard logic interface to BMS System, providing Program and Operator command inputs for normal operation requests. Maintenance test interface is also provided. Fuel selection is done via Boolean inputs, one input requesting each configured fuel. Transfer between fuels is requested by the interface and rules for transfer are dictated by the Master AOI and its configuration.

The BMS Interface also gathers status information from the burner's Combustion Control System (CCS), confirming that the active or selected fuels are included in the CCS' air/fuel ration calculations and the current air/fuel ratio will support combustion of all fuels passed to the burner. This information is used by the BMS Master AOI during fuel selection, live transfer, and subsequent fuel firing activities.

Limitations

The following limitations must be considered when implementing this instruction within a BMS Library application.

AOI Class	Standard
Minimum Instances Required	1
Maximum Instances Allowed	1

System Interaction

The Interface AOI exchanges data with the BMS Master AOI using a Safety Mapped SINT array to pass data Interface-to-Master and Safety Program Output-to-Input parameter linkage for Master-to-Interface data.

Interface to Master Data

Data Type: SINT array, 8 elements minimum. Note that individual SINT elements for device HMI interaction can be added to the SINT array to simplify Safety Tag Mapping but are not required as part of the Interface-to-Master data transfer. Data layout for the SINT array is shown in the table below.

Data Elements from Interface To Master

Starting Element	Data Type	Description
Ref_ToMaster[0]	SINT[4]	Command bits to Master
Ref_ToMaster[0].0	BOOL	Burner START Request
Ref_ToMaster[0].1	BOOL	Burner STOP Request
Ref_ToMaster[0].2	BOOL	Burner RESET Request
Ref_ToMaster[0].3	BOOL	Burner PURGE HOLD Request
Ref_ToMaster[0].4	BOOL	Burner PILOT HOLD Request
Ref_ToMaster[0].5	BOOL	Burner TEST MODE Request

Data Elements from Interface To Master

Starting Element	Data Type	Description
Ref_ToMaster[0].6	BOOL	Burner NORMAL MODE Request
Ref_ToMaster[0].7	BOOL	<reserved>
Ref_ToMaster[1].0	BOOL	<reserved>
Ref_ToMaster[1].1	BOOL	<reserved>
Ref_ToMaster[1].2	BOOL	<reserved>
Ref_ToMaster[1].3	BOOL	Pilot A Valve Proving Start Request
Ref_ToMaster[1].4	BOOL	Pilot B Valve Proving Start Request
Ref_ToMaster[1].5	BOOL	Pilot C Valve Proving Start Request
Ref_ToMaster[1].6	BOOL	Pilot D Valve Proving Start Request
Ref_ToMaster[1].7	BOOL	Fuel A Valve Proving Start Request
Ref_ToMaster[2].0	BOOL	Fuel B Valve Proving Start Request
Ref_ToMaster[2].1	BOOL	Fuel C Valve Proving Start Request
Ref_ToMaster[2].2	BOOL	Fuel D Valve Proving Start Request
Ref_ToMaster[2].3	BOOL	Fuel Header Valve Proving Start Request
Ref_ToMaster[2].4	BOOL	<reserved>
Ref_ToMaster[2].5	BOOL	Pilot A Scavenge Start Request
Ref_ToMaster[2].6	BOOL	Pilot B Scavenge Start Request
Ref_ToMaster[2].7	BOOL	Pilot C Scavenge Start Request
Ref_ToMaster[3].0	BOOL	Pilot D Scavenge Start Request
Ref_ToMaster[3].1	BOOL	Fuel A Scavenge Start Request
Ref_ToMaster[3].2	BOOL	Fuel B Scavenge Start Request
Ref_ToMaster[3].3	BOOL	Fuel C Scavenge Start Request
Ref_ToMaster[3].4	BOOL	Fuel D Scavenge Start Request
Ref_ToMaster[3].5	BOOL	<reserved>
Ref_ToMaster[3].6	BOOL	<reserved>
Ref_ToMaster[3].7	BOOL	<reserved>
Ref_ToMaster[4]	SINT[2]	Burner Fuel Selection and CCS Feedback
Ref_ToMaster[4].0	BOOL	Burner Fuel A Select
Ref_ToMaster[4].1	BOOL	Burner Fuel B Select
Ref_ToMaster[4].2	BOOL	Burner Fuel C Select
Ref_ToMaster[4].3	BOOL	Burner Fuel D Select
Ref_ToMaster[4].4	BOOL	<reserved>
Ref_ToMaster[4].5	BOOL	Burner Fuel A Primary
Ref_ToMaster[4].6	BOOL	Burner Fuel B Primary
Ref_ToMaster[4].7	BOOL	Burner Fuel C Primary
Ref_ToMaster[5].0	BOOL	Burner Fuel D Primary
Ref_ToMaster[5].1	BOOL	<reserved>
Ref_ToMaster[5].2	BOOL	CCS Fuel A Included
Ref_ToMaster[5].3	BOOL	CCS Fuel B Included
Ref_ToMaster[5].4	BOOL	CCS Fuel C Included
Ref_ToMaster[5].5	BOOL	CCS Fuel D Included
Ref_ToMaster[5].6	BOOL	<reserved>
Ref_ToMaster[5].7	BOOL	CCS Fuel/Air Ratio OK
Ref_ToMaster[6]	SINT[2]	Interface AOI Heartbeat Echo
Additional Elements for Individual Device Control from HMI (TEST ENABLE Mode only)		
Ref_ToMaster[8]	SINT (qty as req'd)	Individual Device Control from HMI (1 SINT per device)
Ref_ToMaster[8].0	BOOL	Test Operator Command Acquire Request
Ref_ToMaster[8].1	BOOL	Test Operator Command Relinquish Request
Ref_ToMaster[8].2	BOOL	Test Operator Command Reset Request
Ref_ToMaster[8].3	BOOL	Test Operator Command DeEnergize Request
Ref_ToMaster[8].4	BOOL	Test Operator Command Energize Request

Data Elements from Interface To Master

Starting Element	Data Type	Description
Ref_ToMaster[8].5	BOOL	Test Operator Command <reserved>
Ref_ToMaster[8].6	BOOL	Test Operator Command <reserved>
Ref_ToMaster[8].7	BOOL	Test Operator Command <reserved>

Master to Interface Data

Data Type: raS_UDT_BMS_Master_ToInterface (16 bytes). This UDT is predominantly Boolean data, indicating status and configuration of all pilots and fuels. The UDT also includes system-wide status information provided by the Master to all other BMS AOIs. A list of the UDT elements is shown below.

Data Elements from Master to Interface

Element Name	Data Type	Description
SysSts	raS_UDT_BMS_Master_SysSts	System-wide common status information.
SysSts.Enable	BOOL	Burner System Enable
SysSts.IntlksOK	BOOL	Burner System Interlocks OK
SysSts.MFTR	BOOL	Master Fuel Trip Relay Energized
SysSts.Normal	BOOL	Burner System in NORMAL Operating Mode
SysSts.TestMode	BOOL	Burner System in TEST Mode
SysSts.TestActive	BOOL	Burner in TEST Mode with a Device Test Active
SysSts.NoFlame	BOOL	Pre-Purge NOT Completed – No Flame Allowed
SysSts.eState	SINT	Burner System Master State Value
SysSts.HB	INT	Master Communication Heartbeat
Sts_SafetyTrip	BOOL	Burner Safety Trip
Sts_Ready	BOOL	Burner Ready for Start Request
Sts_PurgeHold	BOOL	Burner Purge and Hold
Sts_PilotHold	BOOL	Burner Hold at Pilot
Sts_AllFuelPoC	BOOL	All Fuel Trains Proof of Closure
Sts_PurgeActive	BOOL	Burner Purge Sequence Active
Sts_PilotActive	BOOL	Any Pilot Sequence Active
Sts_FuelLit	BOOL	Any Fuel Active and Lit
Sts_Modulate	BOOL	Burner Release for Modulation
Sts_FuelTransfer	BOOL	Fuel Transfer In Progress
Sts_FlameSensed	BOOL	Any Flame Sensed at Burner
Sts_VPRequired	BOOL	Any Pilot/Fuel Requires Valve Proving
Sts_FuelSelPending	BOOL	Fuel Selection Change is Pending
Sts_InvalidFuelSelect	BOOL	Requested Fuel Selection is Invalid
Sts_ManReqFail	BOOL	Manual Action Request Failed
Ack_Request	BOOL	Acknowledge Request from Interface
Sts_FuelSelectA	BOOL	Main Fuel A Selected
Sts_FuelSelectB	BOOL	Main Fuel B Selected
Sts_FuelSelectC	BOOL	Main Fuel C Selected
Sts_FuelSelectD	BOOL	Main Fuel D Selected
Sts_FuelPrimaryA	BOOL	Main Fuel A Selected as Primary
Sts_FuelPrimaryB	BOOL	Main Fuel B Selected as Primary
Sts_FuelPrimaryC	BOOL	Main Fuel C Selected as Primary
Sts_FuelPrimaryD	BOOL	Main Fuel D Selected as Primary
Sts_FuelLitA	BOOL	Main Fuel A Lit
Sts_FuelLitB	BOOL	Main Fuel B Lit
Sts_FuelLitC	BOOL	Main Fuel C Lit

Data Elements from Master to Interface

Element Name	Data Type	Description
Sts_FuelLitD	BOOL	Main Fuel D Lit
Sts_PilotActiveA	BOOL	Pilot A Component Active
Sts_PilotActiveB	BOOL	Pilot B Component Active
Sts_PilotActiveC	BOOL	Pilot C Component Active
Sts_PilotActiveD	BOOL	Pilot D Component Active
Sts_FuelActiveA	BOOL	Fuel A Component Active
Sts_FuelActiveB	BOOL	Fuel B Component Active
Sts_FuelActiveC	BOOL	Fuel C Component Active
Sts_FuelActiveD	BOOL	Fuel D Component Active
Sts_FuelHdrActive	BOOL	Fuel Header Component Active
Sts_PilotManVPA	BOOL	Pilot A Manual Valve Proving Active
Sts_PilotManVPB	BOOL	Pilot B Manual Valve Proving Active
Sts_PilotManVPC	BOOL	Pilot C Manual Valve Proving Active
Sts_PilotManVPD	BOOL	Pilot D Manual Valve Proving Active
Sts_FuelManVPA	BOOL	Fuel A Manual Valve Proving Active
Sts_FuelManVPB	BOOL	Fuel B Manual Valve Proving Active
Sts_FuelManVPC	BOOL	Fuel C Manual Valve Proving Active
Sts_FuelManVPD	BOOL	Fuel D Manual Valve Proving Active
Sts_FuelHdrManVP	BOOL	Fuel Header Manual Valve Proving Active
Sts_PilotManScavA	BOOL	Pilot A Manual Scavenging Active
Sts_PilotManScavB	BOOL	Pilot B Manual Scavenging Active
Sts_PilotManScavC	BOOL	Pilot C Manual Scavenging Active
Sts_PilotManScavD	BOOL	Pilot D Manual Scavenging Active
Sts_FuelManScavA	BOOL	Fuel A Manual Scavenging Active
Sts_FuelManScavB	BOOL	Fuel B Manual Scavenging Active
Sts_FuelManScavC	BOOL	Fuel C Manual Scavenging Active
Sts_FuelManScavD	BOOL	Fuel D Manual Scavenging Active
Cfg_HasPilotA	BOOL	Has Pilot A Component
Cfg_HasPilotB	BOOL	Has Pilot B Component
Cfg_HasPilotC	BOOL	Has Pilot C Component
Cfg_HasPilotD	BOOL	Has Pilot D Component
Cfg_HasFuelA	BOOL	Has Fuel A Component
Cfg_HasFuelB	BOOL	Has Fuel B Component
Cfg_HasFuelC	BOOL	Has Fuel C Component
Cfg_HasFuelD	BOOL	Has Fuel D Component
Cfg_HasFuelHdr	BOOL	Has Fuel Header Component
Cfg_HasVPPilotA	BOOL	Pilot A Has Valve Proving
Cfg_HasVPPilotB	BOOL	Pilot B Has Valve Proving
Cfg_HasVPPilotC	BOOL	Pilot C Has Valve Proving
Cfg_HasVPPilotD	BOOL	Pilot D Has Valve Proving
Cfg_HasVPFuelA	BOOL	Fuel A Has Valve Proving
Cfg_HasVPFuelB	BOOL	Fuel B Has Valve Proving
Cfg_HasVPFuelC	BOOL	Fuel C Has Valve Proving
Cfg_HasVPFuelD	BOOL	Fuel D Has Valve Proving
Cfg_HasVPFuelHdr	BOOL	Fuel Header Has Valve Proving
Cfg_HasScavPilotA	BOOL	Pilot A Has Fuel Scavenging
Cfg_HasScavPilotB	BOOL	Pilot B Has Fuel Scavenging
Cfg_HasScavPilotC	BOOL	Pilot C Has Fuel Scavenging
Cfg_HasScavPilotD	BOOL	Pilot D Has Fuel Scavenging
Cfg_HasScavFuelA	BOOL	Fuel A Has Fuel Scavenging

Data Elements from Master to Interface

Element Name	Data Type	Description
Cfg_HasScavFuelB	BOOL	Fuel B Has Fuel Scavenging
Cfg_HasScavFuelC	BOOL	Fuel C Has Fuel Scavenging
Cfg_HasScavFuelD	BOOL	Fuel D Has Fuel Scavenging

Interface AOI Configuration

The only configuration necessary for the interface AOI is determining if PCmd inputs will be used, or if the burner will respond to Inp_CallForHeat maintained input while in Program Command Source Mode. When Cfg_ProgCallForHeat is set, PCmd inputs are ignored and Inp_CallForHeat is used to start/stop the burner as manipulated by external logic. When Cfg_ProgCallForHeat is clear, external logic manipulates the PCmd inputs and Inp_CallForHeat is ignored.

Interlocks

The raS_Opr_BMS_Interface AOI is designed as a Standard Task command and status interface to the BMS Master, and as such, does not provide any additional Interlock capability.

Permissives

The raS_Opr_BMS_Interface AOI is designed as a Standard Task command and status interface to the BMS Master, and as such, does not provide any additional Permissive capability.

Operations

Operation Modes

The BMS Library objects provide only physical operation mode. There is no virtual/simulated mode offered.

Normal Operation Mode

The Interface has two sets of command inputs: Program and Operator. Only one of these command sets is active at any time. Each command set includes requests to switch from one command source to the other. Commands to change operating mode are only provided to the Operator. All command inputs are Boolean and are cleared by the Interface AOI once read by internal code. This has the effect of making all commands rising edge triggered. Note that all Program Command inputs are disabled if Cfg_ProgCallForHeat is set. In that case, the program's interface is a single maintained Boolean input: Inp_CallForHeat.

Test Enable Mode Operation

TEST ENABLE Mode is controlled by the Master AOI. While in TEST ENABLE Mode, the interface is only allowed to issue a Burner Reset command. All other interface commands are ignored.

While in TEST ENABLE Mode, the individual SINT device HMI interface elements (one SINT for each raM_Dvc_BMS_D2S that exists in the BMS system) will be active, allowing testing of devices individually from an HMI.

Required Files

Device Objects include Add-On Instructions (AOIs) and HMI faceplates. The revision number (e.g. 1.01) used in filenames can change as new revisions are created.

Controller Files

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

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

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

Device/Item	Add-On Instruction	ROUTINE Import
BMS Interface	raS_Opr_BMS_Interface_1.00_AOI.L5X	raS_Opr_BMS_Interface_1.00_StdROUTINE.L5X

FactoryTalk® View HMI Files

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

Device/Item	Type	FactoryTalk® View SE Faceplate
BMS Interface	Display	(raS-1_00-SE) raS_Opr_BMS_Interface-Faceplate.gfx
Graphic Symbols	Global Object	(raS-1-SE) Graphic Symbols - BMS.ggfx
Toolbox	Global Object	(raS-1-SE) Toolbox - BMS.ggfx

Add-On Instruction I/O Data

InOut Data

InOut	Function / Description	DataType
Ref_FromMaster	Status and configuration data from Master AOI to Interface AOI	raS_UDT_BMS_Master_ToInterface
Ref_ToMaster	Data from Interface AOI to Master AOI	SINT[8]

Input Data

Input	Function/Description	DataType
Inp_Enable	Enable BMS System Control	BOOL
Inp_CallForHeat	Burner Start Request (Cfg_CallForHeat set)	BOOL
Inp_FuelReqA	Request Use of Fuel A	BOOL
Inp_FuelReqB	Request Use of Fuel B	BOOL
Inp_FuelReqC	Request Use of Fuel C	BOOL
Inp_FuelReqD	Request Use of Fuel D	BOOL
Inp_FuelPri	Primary Fuel Select Value (0-4)	DINT
Inp_CCSFuelA	CCS Air/Fuel Ratio Control Includes Fuel A	BOOL
Inp_CCSFuelB	CCS Air/Fuel Ratio Control Includes Fuel B	BOOL
Inp_CCSFuelC	CCS Air/Fuel Ratio Control Includes Fuel C	BOOL
Inp_CCSFuelD	CCS Air/Fuel Ratio Control Includes Fuel D	BOOL
Inp_FAOK	CCS Air Supply is Sufficient for Combustion of Included Fuels	BOOL

Output Data

Output	Function/Descritpion	DataType
Out_BMSReset	Reset BMS Hardware Devices	BOOL
Sts_SafetyTrip	Burner Safety Trip	BOOL
Sts_Ready	Burner Ready to Start	BOOL
Sts_Active	Burner Active	BOOL
Sts_Normal	Burner System operating in NORMAL Mode	BOOL
Sts_Test	Burner System operating in TEST ENABLE Mode	BOOL
Sts_TestActive	Burner System in TEST ENABLE Mode, Device Testing is Active	BOOL
Sts_Prog	Burner System in Program Command Source (Logic in Control)	BOOL
Sts_Oper	Burner System in Operator/HMI Command Source (Operator in Control)	BOOL
Sts_VPRequired	Any Pilot/Fuel Requires Valve Proving Prior to Next Light-Off	BOOL
Sts_FlamePresent	Any Flame Sensed	BOOL
Sts_PurgeActive	Pre-Ignition Purge Active	BOOL
Sts_PurgeHold	Burner in Purge Hold	BOOL
Sts_Pilot	Any Pilot Active	BOOL
Sts_PilotHold	Burner in Pilot Hold	BOOL
Sts_BurnerLit	Any Main Fuel Lit	BOOL
Sts_Modulate	Release for Modulation	BOOL
Sts_ModulateHold	Modulation Hold for Fuel Change	BOOL
Sts_PostPurge	Post-Purge Active	BOOL
Sts_InvFuelRequest	Requested Fuel Selection Invalid	BOOL
Sts_FuelSelA	Main Fuel A Selected	BOOL
Sts_FuelPriA	Main Fuel A is Primary	BOOL
Sts_FuelLitA	Main Fuel A Lit	BOOL
Sts_FuelSelB	Main Fuel B Selected	BOOL
Sts_FuelPriB	Main Fuel B is Primary	BOOL
Sts_FuelLitB	Main Fuel B Lit	BOOL

Output	Function/Descritpion	DataType
Sts_FuelSelC	Main Fuel C Selected	BOOL
Sts_FuelPriC	Main Fuel C is Primary	BOOL
Sts_FuelLitC	Main Fuel C Lit	BOOL
Sts_FuelSelD	Main Fuel D Selected	BOOL
Sts_FuelPriD	Main Fuel D is Primary	BOOL
Sts_FuelLitD	Main Fuel D Lit	BOOL
Sts_PilotActiveA	Pilot A Component Active	BOOL
Sts_PilotManVPA	Pilot A Component Manual Valve Proving Active	BOOL
Sts_PilotManScavA	Pilot A Component Manual Fuel Scavenge Active	BOOL
Sts_PilotActiveB	Pilot B Component Active	BOOL
Sts_PilotManVPB	Pilot B Component Manual Valve Proving Active	BOOL
Sts_PilotManScavB	Pilot B Component Manual Fuel Scavenge Active	BOOL
Sts_PilotActiveC	Pilot C Component Active	BOOL
Sts_PilotManVPC	Pilot C Component Manual Valve Proving Active	BOOL
Sts_PilotManScavC	Pilot C Component Manual Fuel Scavenge Active	BOOL
Sts_PilotActiveD	Pilot D Component Active	BOOL
Sts_PilotManVPD	Pilot D Component Manual Valve Proving Active	BOOL
Sts_PilotManScavD	Pilot D Component Manual Fuel Scavenge Active	BOOL
Sts_FuelActiveA	Fuel A Component Active	BOOL
Sts_FuelManVPA	Fuel A Component Manual Valve Proving Active	BOOL
Sts_FuelManScavA	Fuel A Component Manual Fuel Scavenge Active	BOOL
Sts_FuelActiveB	Fuel B Component Active	BOOL
Sts_FuelManVPB	Fuel B Component Manual Valve Proving Active	BOOL
Sts_FuelManScavB	Fuel B Component Manual Fuel Scavenge Active	BOOL
Sts_FuelActiveC	Fuel C Component Active	BOOL
Sts_FuelManVPC	Fuel C Component Manual Valve Proving Active	BOOL
Sts_FuelManScavC	Fuel C Component Manual Fuel Scavenge Active	BOOL
Sts_FuelActiveD	Fuel D Component Active	BOOL
Sts_FuelManVPD	Fuel D Component Manual Valve Proving Active	BOOL
Sts_FuelManScavD	Fuel D Component Manual Fuel Scavenge Active	BOOL
Sts_FuelHdrActive	Fuel Header Component Active	BOOL
Sts_FuelHdrManVP	Fuel Header Component Manual Valve Proving Active	BOOL
Sts_InvPriRequest	Primary Fuel Request is Out of Range	BOOL
Sts_ManReqFail	Manual Activity Request Failure	BOOL

Parameters

The following tables list the parameters available for the raS_Opr_BMS_Interface AOI.

Command Parameter Name	Command Source Mode	Function
PCmd_Start	Program Only	Burner Start Request (Cfg_CallForHeat clear)
PCmd_Stop	Program Only	Burner Stop Request (Cfg_CallForHeat clear)
PCmd_Purge	Program Only	Request Burner Pre-Ignition Purge and Hold (Cfg_CallForHeat clear)
PCmd_Pilot	Program Only	Request Burner Hold at or Revert to Pilot (Cfg_CallForHeat clear)
PCmd_Oper	Program/Test	Request Operator Command Source Mode
PCmd_Prog	Operator Only	Request Program Command Source Mode
OCmd		HMI Commands (bit-mapped DINT)
OCmd.0	Operator Only	Operator Burner START Request
OCmd.1	Operator Only	Operator Burner STOP Request

Command Parameter Name	Command Source Mode	Function
OCmd.2	Operator Only	Operator Burner RESET Request
OCmd.3	Operator Only	Operator Burner PURGE HOLD Request
OCmd.4	Operator Only	Operator Burner PILOT HOLD Request
OCmd.5	Operator Only	Operator Burner TEST MODE Request
OCmd.6	Operator Only	Operator Burner NORMAL MODE Request
OCmd.7	Operator Only	Operator Request for PROGRAM Command Source
OCmd.8	Program Only	Operator Request for OPERATOR Command Source
OCmd.9		<reserved>
OCmd.10		<reserved>
OCmd.11	Operator Only	Operator Pilot A Valve Proving Start Request
OCmd.12	Operator Only	Operator Pilot B Valve Proving Start Request
OCmd.13	Operator Only	Operator Pilot C Valve Proving Start Request
OCmd.14	Operator Only	Operator Pilot D Valve Proving Start Request
OCmd.15	Operator Only	Operator Fuel A Valve Proving Start Request
OCmd.16	Operator Only	Operator Fuel B Valve Proving Start Request
OCmd.17	Operator Only	Operator Fuel C Valve Proving Start Request
OCmd.18	Operator Only	Operator Fuel D Valve Proving Start Request
OCmd.19	Operator Only	Operator Fuel Header Valve Proving Start Request
OCmd.20		<reserved>
OCmd.21	Operator Only	Operator Pilot A Scavenge Start Request
OCmd.22	Operator Only	Operator Pilot B Scavenge Start Request
OCmd.23	Operator Only	Operator Pilot C Scavenge Start Request
OCmd.24	Operator Only	Operator Pilot D Scavenge Start Request
OCmd.25	Operator Only	Operator Fuel A Scavenge Start Request
OCmd.26	Operator Only	Operator Fuel B Scavenge Start Request
OCmd.27	Operator Only	Operator Fuel C Scavenge Start Request
OCmd.28	Operator Only	Operator Fuel D Scavenge Start Request
OCmd.29 to .31		<reserved>

Configuration

The following tables list the configuration parameters available for the raS_Opr_BMS_Interface AOI.

Configuration Parameter Name	Function/Descritpion	DataType
Cfg_ProgCallForHeat	Set to Activate Inp_CallForHeat	BOOL

Data Types

The following BMS data types and interface tags are used to interface between BMS instructions. These data types are used within the raS_Opr_BMS_Interface instruction.

raS_UDT_BMS_Master_ToInterface

Member	Description	Data Type
SysSts		raS_UDT_BMS_Master_SysSts
Sts_SafetyTrip	Burner Safety Trip	BOOL
Sts_Ready	Burner Ready for Start Request	BOOL
Sts_PurgeHold	Burner Purge and Hold	BOOL
Sts_PilotHold	Burner Hold at Pilot	BOOL
Sts_AllFuelPoC	All Fuel Trains Proof of Closure	BOOL
Sts_PurgeActive	Burner Purge Sequence Active	BOOL
Sts_PilotActive	Any Pilot Sequence Active	BOOL
Sts_FuelLit	Any Fuel Active and Lit	BOOL
Sts_Modulate	Burner Release for Modulation	BOOL
Sts_FuelTransfer	Fuel Transfer In Progress	BOOL
Sts_FlameSensed	Any Flame Sensed at Burner	BOOL
Sts_VPRequired	Any Pilot/Fuel Requires Valve Proving	BOOL
Sts_FuelSelPending	Fuel Selection Change is Pending	BOOL
Sts_FuelSelInv	Requested Fuel Selection is Invalid	BOOL
Sts_ManReqFail	Manual Action Request Failed	BOOL
Ack_Request	Acknowledge Request from Interface	BOOL
Sts_FuelSelectA	Main Fuel A Selected	BOOL
Sts_FuelSelectB	Main Fuel B Selected	BOOL
Sts_FuelSelectC	Main Fuel C Selected	BOOL
Sts_FuelSelectD	Main Fuel D Selected	BOOL
Sts_FuelPrimaryA	Main Fuel A Selected as Primary	BOOL
Sts_FuelPrimaryB	Main Fuel B Selected as Primary	BOOL
Sts_FuelPrimaryC	Main Fuel C Selected as Primary	BOOL
Sts_FuelPrimaryD	Main Fuel D Selected as Primary	BOOL
Sts_FuelLitA	Main Fuel A Lit	BOOL
Sts_FuelLitB	Main Fuel B Lit	BOOL
Sts_FuelLitC	Main Fuel C Lit	BOOL
Sts_FuelLitD	Main Fuel D Lit	BOOL
Sts_PilotActiveA	Pilot A Component Active	BOOL
Sts_PilotActiveB	Pilot B Component Active	BOOL
Sts_PilotActiveC	Pilot C Component Active	BOOL
Sts_PilotActiveD	Pilot D Component Active	BOOL
Sts_FuelActiveA	Fuel A Component Active	BOOL
Sts_FuelActiveB	Fuel B Component Active	BOOL
Sts_FuelActiveC	Fuel C Component Active	BOOL
Sts_FuelActiveD	Fuel D Component Active	BOOL
Sts_FuelHdrActive	Fuel Header Component Active	BOOL
Sts_PilotVPA	Pilot A VP Active	BOOL
Sts_PilotVPB	Pilot B VP Active	BOOL
Sts_PilotVPC	Fuel C VP Active	BOOL
Sts_PilotVPD	Pilot D VP Active	BOOL
Sts_FuelVPA	Fuel A VP Active	BOOL
Sts_FuelVPB	Fuel B VP Active	BOOL
Sts_FuelVPC	Fuel C VP Active	BOOL

Member	Description	Data Type
Sts_FuelVPD	Fuel D VP Active	BOOL
Sts_FuelHdrVP	Fuel Header VP Active	BOOL
Sts_PilotManScavA	Pilot A Manual Scavenging Active	BOOL
Sts_PilotManScavB	Pilot B Manual Scavenging Active	BOOL
Sts_PilotManScavC	Pilot C Manual Scavenging Active	BOOL
Sts_PilotManScavD	Pilot D Manual Scavenging Active	BOOL
Sts_FuelManScavA	Fuel A Manual Scavenging Active	BOOL
Sts_PilotManScavB	Pilot B Manual Scavenging Active	BOOL
Sts_PilotManScavC	Pilot C Manual Scavenging Active	BOOL
Sts_PilotManScavD	Pilot D Manual Scavenging Active	BOOL
Sts_FuelManScavA	Fuel A Manual Scavenging Active	BOOL
Sts_FuelManScavB	Fuel B Manual Scavenging Active	BOOL
Sts_FuelManScavC	Fuel C Manual Scavenging Active	BOOL
Sts_FuelManScavD	Fuel D Manual Scavenging Active	BOOL
Cfg_HasBurnerFuelA	Has Burner Main Fuel A	BOOL
Cfg_HasBurnerFuelB	Has Burner Main Fuel B	BOOL
Cfg_HasBurnerFuelC	Has Burner Main Fuel C	BOOL
Cfg_HasBurnerFuelD	Has Burner Main Fuel D	BOOL
Cfg_HasPilotA	Has Pilot A Component	BOOL
Cfg_HasPilotB	Has Pilot B Component	BOOL
Cfg_HasPilotC	Has Pilot C Component	BOOL
Cfg_HasPilotD	Has Pilot D Component	BOOL
Cfg_HasFuelA	Has Fuel A Component	BOOL
Cfg_HasFuelB	Has Fuel B Component	BOOL
Cfg_HasFuelC	Has Fuel C Component	BOOL
Cfg_HasFuelD	Has Fuel D Component	BOOL
Cfg_HasFuelHdr	Has Fuel Header Component	BOOL
Cfg_HasVPPilotA	Pilot A Has Valve Proving	BOOL
Cfg_HasVPPilotB	Pilot B Has Valve Proving	BOOL
Cfg_HasVPPilotC	Pilot C Has Valve Proving	BOOL
Cfg_HasVPPilotD	Pilot D Has Valve Proving	BOOL
Cfg_HasVPFuelA	Fuel A Has Valve Proving	BOOL
Cfg_HasVPFuelB	Fuel B Has Valve Proving	BOOL
Cfg_HasVPFuelC	Fuel C Has Valve Proving	BOOL
Cfg_HasVPFuelD	Fuel D Has Valve Proving	BOOL
Cfg_HasVPFuelHdr	Fuel Header Has Valve Proving	BOOL
Cfg_HasScavPilotA	Pilot A Has Fuel Scavenging	BOOL
Cfg_HasScavPilotB	Pilot B Has Fuel Scavenging	BOOL
Cfg_HasScavPilotC	Pilot C Has Fuel Scavenging	BOOL
Cfg_HasScavPilotD	Pilot D Has Fuel Scavenging	BOOL
Cfg_HasScavFuelA	Fuel A Has Fuel Scavenging	BOOL
Cfg_HasScavFuelB	Fuel B Has Fuel Scavenging	BOOL
Cfg_HasScavFuelC	Fuel C Has Fuel Scavenging	BOOL
Cfg_HasScavFuelD	Fuel D Has Fuel Scavenging	BOOL

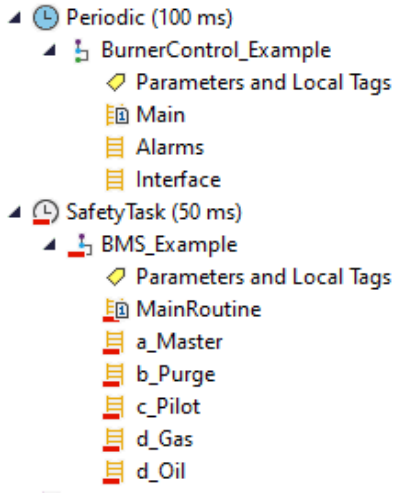
raS_UDT_BMS_Master_SysSts

BMS Master Burner System Status.

Input	Description	Data Type
Enable	Burner System Enable	BOOL
IntlksOK	Burner System Interlocks OK	BOOL
MFTR	Master Fuel Trip Relay Energized	BOOL
Normal	Burner System in NORMAL MODE	BOOL
TestMode	Burner System in TEST MODE	BOOL
TestActive	Burner System in TEST MODE with any Device Test Active	BOOL
NoFlame	Pre-Purge Not Completed. No Flame Allowed	BOOL
eState	Burner System Master State	SINT
HB	Communication Heartbeat	INT

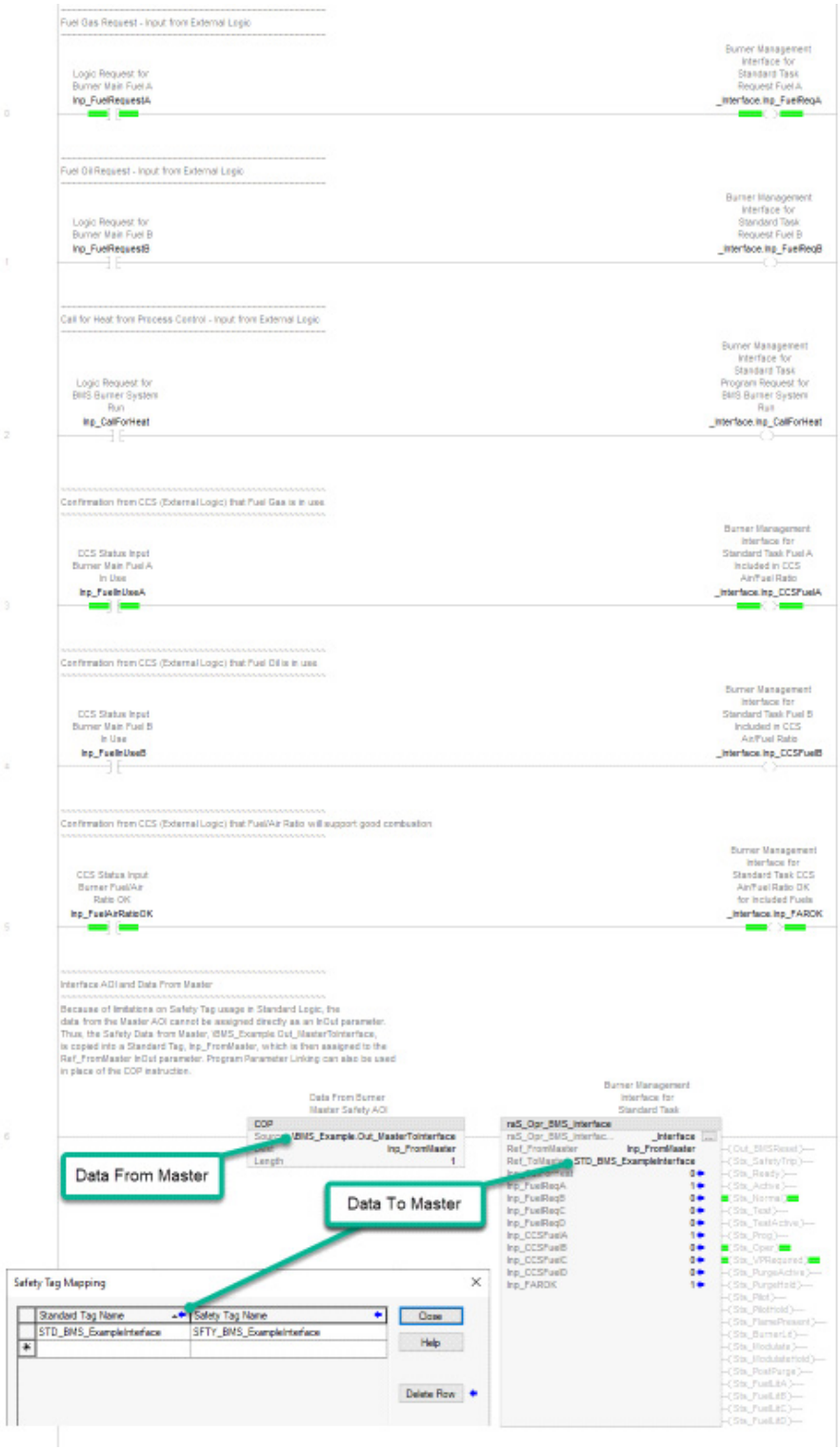
Programming Example

General arrangement for this example is shown in the following figure. All Safety-rated AOIs are contained within the BMS_Example program while the raS_Opr_BMS_Interface standard AOI is located in the BurnerControl_Example standard program.



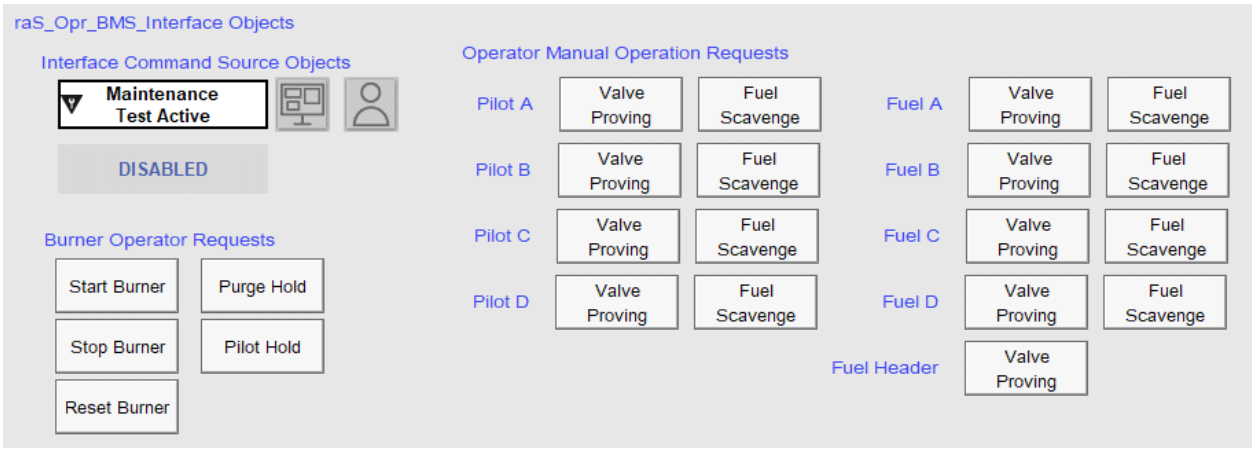
This example shows implementation for two fuels, Gas and Oil, as part of the complete BMS application. The rungs that follow are taken from the BurnerControl_Example - Interface routine:

- Rung 0: Logic driving Interface AOI's Inp_FuelReqA. When preceding logic is true, Interface will pass a request to burn Main Fuel A (Fuel Gas in this example) to the Master.
- Rung 1: Logic driving Interface AOI's Inp_FuelReqB. When preceding logic is true, Interface will pass a request to burn Main Fuel B (Fuel Oil in this example) to the Master.
- Rung 2: Logic driving Interface AOI's Inp_CallForHeat. When preceding logic is true, Interface will pass a request to light the burner to the Master. Note that this input must be configured for use, and the Interface must be in Program command source (Sts_Prog set) to respond to this input.
- Rung 3: Feedback from Combustion Control Logic confirming that Fuel A (Fuel Gas) is included in Fuel/Air regulation, and its actuator is under control of the CCS logic.
- Rung 4: Feedback from Combustion Control Logic confirming that Fuel B (Fuel Oil) is included in Fuel/Air regulation, and its actuator is under control of the CCS logic.
- Rung 5: Feedback from Combustion Control Logic confirming that calculations indicate the burners Fuel/Air Ratio will support good combustion.
- Rung 6: Copy Master-to-Interface data in \BMS_Example.Out_MasterToInterface tag into local Inp_FromMaster tag of same data type. Inp_FromMaster tag is placed in the Interface AOI's Ref_FromMaster InOut parameter. Data from the Interface AOI back to the Master is held in STD_BMS_ExampleInterface, a SINT array mapped to the safety tag version of the same structure for use by the Master.

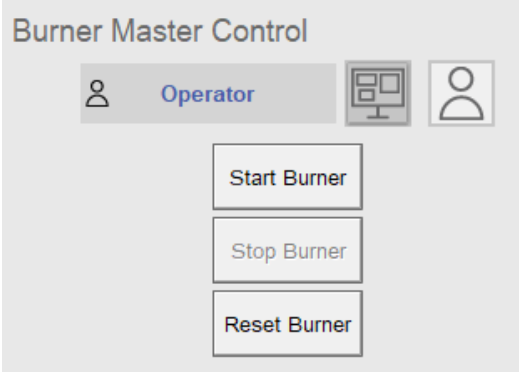


Global Object

Global Objects are provided for use with the Interface AOI and can be used for interface control and status on any display. The global objects include command source control/indication, operating mode control/indication, and burner command buttons. The global objects are shown in the following image.



An example of how this may be implemented follows.



BMS Master (raS_Opr_BMS_Master)

Overview

The BMS Master AOI coordinates the functions of all other BMS AOIs, receives commands (requests) from the Interface, validates those requests against its configuration, and converts those requests into coordinated actions that manipulate physical devices associated with the BMS system. Common interlock inputs such as process safety interlocks (e.g., high temperature limit) are monitored by the Master, along with interlock and trip status from all subordinate AOIs. A safety trip status received by the Master from any of these sources may or may not result in a safety trip for the entire burner.

Limitations

The following limitations must be considered when implementing this instruction within a BMS Library application.

AOI Class	Safety
Minimum Instances Required	1
Maximum Instances Allowed	1

System Interaction

The Master continuously communicates bidirectionally with all other AOIs by way of a single UDT with sub elements specific to each component AOI. One data point in each sub element is a heartbeat value, generated by the Master and echoed by the component. Each component gets a unique value that must be echoed back within two Safety Task execution cycles or a fault is generated, resulting in a safety trip if the burner is in operation.

Common System Data

The Master AOI assembles common system-scoped data in a UDT named raS_UDT_BMS_Master_SysSts (4 bytes). This UDT is included as an element in all sub elements transmitting data to the Burner's component AOIs. Elements in this common UDT are shown in the following table.

Common System Data from Master

Element Name	Data Type	Description
SysSts	raS_UDT_BMS_Master_SysSts	System-wide common status information.
SysSts.Enable	BOOL	Burner System Enable
SysSts.IntlksOK	BOOL	Burner System Interlocks OK
SysSts.MFTR	BOOL	Master Fuel Trip Relay Energized
SysSts.Normal	BOOL	Burner System in NORMAL Operating Mode
SysSts.TestMode	BOOL	Burner System in TEST Mode
SysSts.TestActive	BOOL	Burner in TEST Mode with a Device Test Active

Common System Data from Master

Element Name	Data Type	Description
SysSts.NoFlame	BOOL	Pre-Purge NOT Completed – No Flame Allowed
SysSts.eState	SINT	Burner System Master State Value
SysSts.HB	INT	Master Communication Heartbeat

Note that the heartbeat (SysSts.HB) to each component AOI is included as part of this data structure. Each component is assigned a unique range of values for this heartbeat to ensure that to confirm proper data connections between these AOIs. This ensures data in both directions (from Master to component and from component to Master) are both connected to the same component. The value ranges for each component AOI are shown in the table below.

Master Heartbeat Value Ranges

Component	Master Heartbeat Reference	Heartbeat Value Range
Interface	Ref_InterfaceData.To.User.SysSts.HB	100 to 119
Purge	Ref_InterfaceData.To.Purge.SysSts.HB	200 to 219
Pilot A	Ref_InterfaceData.To.Pilot[0].SysSts.HB	300 to 319
Pilot B	Ref_InterfaceData.To.Pilot[1].SysSts.HB	320 to 339
Pilot C	Ref_InterfaceData.To.Pilot[2].SysSts.HB	340 to 359
Pilot D	Ref_InterfaceData.To.Pilot[3].SysSts.HB	360 to 379
Fuel A	Ref_InterfaceData.To.Fuel[0].SysSts.HB	400 to 419
Fuel B	Ref_InterfaceData.To.Fuel[1].SysSts.HB	420 to 439
Fuel C	Ref_InterfaceData.To.Fuel[2].SysSts.HB	440 to 459
Fuel D	Ref_InterfaceData.To.Fuel[3].SysSts.HB	460 to 479
Fuel Header	Ref_InterfaceData.To.FuelHdr.SysSts.HB	500 to 519

Common Component Data

Each component AOI assembles data common to all in a UDT named raS_UDT_BMS_CommonStatus (8 bytes). This UDT is included as an element in all tags transmitting data from each component to the Master AOI. Elements in this common UDT are shown in the following table.

Common Data from Component AOIs to Master

Element Name	Data Type	Description
Enabled	BOOL	Component is Enabled
SafetyTrip	BOOL	The Component generated a Safety Trip
IntlkOK	BOOL	Component Interlocks OK
Ready	BOOL	Component Ready for Start Command
Active	BOOL	Component is Active
Normal	BOOL	Component in NORMAL Mode
TestMode	BOOL	Component in TEST ENABLE Mode
TestActive	BOOL	Component has a Device Test active.
eState	SINT	Component's Current State Value
HBEcho	INT	Master Communication Heartbeat Echo
BuildID	DINT	Component Build ID (hidden)

Master to Interface Data

Data Type: raS_UDT_BMS_Master_ToInterface (16 bytes).

Master Parameter Reference: Ref_InterfaceData.To.User (InOut)

This UDT is predominantly Boolean data, indicating status and configuration of all pilots and fuels. The UDT also includes system-wide status information provided by the Master to all other BMS AOIs. A list of the UDT elements is shown in the preceding Interface section.

Master to Purge Data

Data Type: raS_UDT_BMS_Master_ToPurge (8 bytes).

Master Parameter Reference: Ref_InterfaceData.To.Purge (InOut)

Data from Master to Purge includes the SysSts structure, start/stop/reset commands, and information gathered from pilot and fuel AOIs necessary for purge operation. A list of UDT elements is shown in the Purge section that follows.

Master to Pilot Data

Data Type: raS_UDT_BMS_Master_ToPilot (8 bytes) array, 4 elements.

Master Parameter Reference: Ref_InterfaceData.To.Pilot[4] (InOut)

Data from Master to Pilot includes the SysSts structure, operational commands, and information gathered from purge and fuel AOIs necessary for pilot operation. A list of UDT elements is shown in the Pilot section below.

Master to Fuel Data

Data Type: raS_UDT_BMS_Master_ToFuel (8 bytes) array, 4 elements.

Master Parameter Reference: Ref_InterfaceData.To.Fuel[4] (InOut)

Data from Master to Fuel includes the SysSts structure, operational commands, and information gathered from purge and pilot AOIs necessary for fuel operation. A list of UDT elements is shown in the Fuel section below.

Master to Fuel Header Data

Data Type: raS_UDT_BMS_Master_ToFuel (8 bytes).

Master Parameter Reference: Ref_InterfaceData.To.FuelHdr (InOut)

Data from Master to Fuel includes the SysSts structure, operational commands, and information gathered from purge and pilot AOIs necessary for fuel operation. A list of UDT elements is shown in the Fuel Header section below.

Master Data from Interface

Data Type: SINT[8] (8 bytes).

Master Parameter Reference: Ref_InterfaceData.From.Interface (InOut)

This array is primarily Boolean data, including all commands parsed by the Interface AOI, fuel selection requests, and air/fuel status in formation from burner combustion control (CCS). The array also includes the Interface AOI's heartbeat echo. The array can be expanded to include HMI command elements used by individual devices, but this data is not used by the Master AOI. General layout of the array elements is shown in the preceding Interface section.

Master Data from Purge

Data Type: raS_UDT_BMS_Purge_ToMaster (8 bytes).

Master Parameter Reference: Ref_InterfaceData.From.Purge (InOut)

Data from Purge to Master includes the common component data structure, purge operational status, and command acknowledge bits. A list of UDT elements is shown in the Purge section that follows.

Master Data from Pilots

Data Type: raS_UDT_BMS_Pilot_ToMaster (8 bytes) array, 4 elements.

Master Parameter Reference: Ref_InterfaceData.From.Pilot[4] (InOut)

Data from Pilot to Master includes the common component data structure, pilot operational status, pertinent configuration data, and command acknowledge bits. A list of UDT elements is shown in the Pilot section below.

Master Data from Fuels

Data Type: raS_UDT_BMS_Fuel_ToMaster (8 bytes) array, 4 elements.

Master Parameter Reference: Ref_InterfaceData.From.Fuel[4] (InOut)

Data from Fuel to Master includes the common component data structure, fuel operational status, pertinent configuration data, and command acknowledge bits. A list of UDT elements is shown in the Fuel section below.

Master Data from Fuel Header

Data Type: raS_UDT_BMS_Fuel_ToMaster (8 bytes).

Master Parameter Reference: Ref_InterfaceData.From.FuelHdr (InOut)

Data from Fuel Header to Master includes the common component data structure, fuel operational status, pertinent configuration data, and command acknowledge bits. A list of UDT elements is shown in the Fuel Header section below.

Master AOI Configuration

The Master AOI determines the overall burner configuration including firing mode, number of fuel trains, and number of pilot fuel trains supplying the burner. Other configuration settings allow or disallow hold after purge, hold at pilot, and response to single fuel trip. The Master AOI provides up to 8 user-assignable interlock inputs and up to 4 user-assignable permissive inputs that are enabled by configuration. Master configuration determines whether Pilot components are used as main fuel pilots (one or more fuel components) or as spark-ignited fuels (zero fuel components). Configuration parameter details are shown below.

Interlocks

The raS_Opr_BMSMaster AOI monitors 27 interlock conditions, 8 of which are user selectable. The following table identifies each interlock condition, when each is monitored, and when each will trigger a Safety Trip.

Master AOI Interlock List

Interlock Condition	Interlock Enable	Safety Trip Enable
Master AOI Disabled (Inp_Enabled = 0 OR EnableInFalse)	Always	Master Active
No Safety Task Signature	Always	Always
Configuration Error Exists (Sts_bCfgErr <> 0)	Master Enabled	Master Enabled
Operational Fault Exists (Sts_bFaults <> 0)	Master Enabled	Master Enabled
No Fuels Ready at Light-off	Master Enabled	Never
Process Limit A Not OK (Inp_ProcessLimitA = 0)	Master Enabled	Master Enabled
Process Limit B Not OK (Inp_ProcessLimitB = 0)	Master Enabled and Has Process Limit B	Master Enabled and Has Process Limit B
Local E-Stop Not OK (Inp_EStopLocal = 0)	Master Enabled	Master Enabled
Remote E-Stop Not OK (Inp_EStopRemote = 0)	Master Enabled	Master Enabled
MFTR Not Energized	Master Enabled	Estops, WDs, and Process Limits OK
VP Req'd at Pre-Purge Done	Master State (5) and Air At Light-Off	Master State (5)
User Interlock 01 Not OK (Inp_Intlk01 = 0)	Master Enabled and Has User Interlock 01	Master Active and Has User Interlock 01
User Interlock 02 Not OK (Inp_Intlk02 = 0)	Master Enabled and Has User Interlock 02	Master Active and Has User Interlock 02
User Interlock 03 Not OK (Inp_Intlk03 = 0)	Master Enabled and Has User Interlock 03	Master Active and Has User Interlock 03
User Interlock 04 Not OK (Inp_Intlk04 = 0)	Master Enabled and Has User Interlock 04	Master Active and Has User Interlock 04
User Interlock 05 Not OK (Inp_Intlk05 = 0)	Master Enabled and Has User Interlock 05	Master Enabled and Has User Interlock 05
User Interlock 06 Not OK (Inp_Intlk06 = 0)	Master Enabled and Has User Interlock 06	Master Enabled and Has User Interlock 06
User Interlock 07 Not OK (Inp_Intlk07 = 0)	Master Enabled and Has User Interlock 07	Master Enabled and Has User Interlock 07
User Interlock 08 Not OK (Inp_Intlk08 = 0)	Master Enabled and Has User Interlock 08	Master Enabled and Has User Interlock 08
MFTR Device Interlocked	Master Enabled	Always
Purge Component Safety Trip	Master Enabled	Never
Pilot A Component Safety Trip	Master Enabled, is not Spark-ignited Fuel, and not Hold on Fuel Trip	Never
Pilot B Component Safety Trip	Master Enabled, Has Pilot B, is not Spark-Ignited Fuel, and not Hold on Fuel Trip	Never
Pilot C Component Safety Trip	Master Enabled, Has Pilot C, is not Spark-Ignited Fuel, and not Hold on Fuel Trip	Never
Pilot D Component Safety Trip	Master Enabled, Has Pilot D, is not Spark-Ignited Fuel, and not Hold on Fuel Trip	Never
Fuel A Component Safety Trip	Master Enabled	Never
Fuel B Component Safety Trip	Master Enabled and Has Fuel B	Never

Master AOI Interlock List

Interlock Condition	Interlock Enable	Safety Trip Enable
Fuel C Component Safety Trip	Master Enabled and Has Fuel C	Never
Fuel D Component Safety Trip	Master Enabled and Has Fuel D	Never
Fuel Header Component Safety Trip	Master Enabled and Has Fuel Header	Never

Permissives

The raS_Opr_BMSMaster AOI monitors 8 permissive conditions, 4 of which are user selectable. The following table identifies each permissive condition and when each is monitored. Once the Master AOI transitions to an Active state, permissive conditions no longer impact operation.

Master AOI Permissive List

Permissive Lost Condition	Permissive Enable
Not NORMAL Operating Mode (Sts_Normal = 0)	Master Not Active
Fuel Selection Invalid (Sts_FuelRequestInv = 1)	Master Not Active
Purge AOI Not Ready	Master Not Active
Fuel Control Sequence Faulted	Master Not Active
User Permissive 01 Not OK (Inp_Perm01 = 0)	Master Not Active and Has User Permissive 01
User Permissive 02 Not OK (Inp_Perm02 = 0)	Master Not Active and Has User Permissive 02
User Permissive 03 Not OK (Inp_Perm03 = 0)	Master Not Active and Has User Permissive 03
User Permissive 04 Not OK (Inp_Perm04 = 0)	Master Not Active and Has User Permissive 04

Operations

Operation Modes

The BMS Library objects provide only physical operation mode. There is no virtual/simulated mode offered.

Normal Operation Mode

The Master AOI must be in NORMAL Operating mode to operate the burner. While in NORMAL Mode, the Master AOI receives requests from the Interface AOI, validates the requests, and if appropriate, makes appropriate operational changes based on the command received. If a request is made that is not appropriate for the current state or condition, that request is ignored. There is no direct interaction with the Master AOI except through the Interface.

NORMAL Mode is transmitted to all subordinate (Purge, Pilot, and Fuel) AOIs.

Test Enable Mode Operation

TEST ENABLE Mode is requested by setting Inp_TestEnable and once the Master AOI is in Test Mode, the burner is prevented from operating normally. Any commands to start or otherwise activate the Master AOI are ignored. Clearing Inp_TestEnable will revert the Master AOI to NORMAL Mode.

While in TEST ENABLE Mode, the Master Fuel Trip Relay (MFTR) is energized to allow Maintenance Testing of individual devices.

TEST ENABLE Mode is transmitted to all component (Purge, Pilot, and Fuel) AOIs.

Device Test Active

Device Test Active indicates a device within the BMS system is actively being tested. The Master AOI must be in TEST ENABLE Mode to activate Device Test Active and will remain in TEST ENABLE Mode while the device is being tested.

All component (Purge, Pilot, and Fuel) AOIs report their TestActive status to the Master and the Master in turn echoes this status back to all components as Device Test Active. If more than one component reports TestActive concurrently, the Master AOI generates an Operational Fault and a Safety Trip ensues, cancelling TEST ENABLE Mode and any active testing being performed.

Operation

The overall burner start-run-stop cycle is controlled by the Master, relying on each component Purge, Pilot, or Fuel AOI to perform its assigned duties. The general order of activities is as follows:

1. Continually monitor E-Stop, External Watchdog, and Process Limit inputs.
   a. Energize MFTR when all are OK and Burner is Enabled.
2. Continually monitor interlock and safety trip status of all components.
3. Monitor permissive inputs while Burner is idle.
4. Receive “Start” request from Interface.
5. If all permissive, interlock, and safety trip conditions are OK, request a Pre-Ignition Purge (Pre-Purge) from the Purge AOI.
6. Once the Pre-Ignition Purge is complete as indicated by signal from the Purge AOI, request a light-off of the selected fuel(s).
   a. If multiple fuels are allowed and are selected concurrently, one fuel at a time is lit. The order is A-B-C-D unless a fuel is selected as Primary.
   b. If a fuel is selected as Primary, that fuel is lit first with the remaining selected fuels lit in the preceding order.
   c. A fuel that is not ready to light (all component interlocks and permissives satisfied) will be skipped in the light-off sequence and the fuel next in line is lit.
7. When a light-off request is made to a fuel component, it responds with a request for whatever pilot(s) it requires.
   a. Each fuel component carries configuration that determines which pilots are required for its light-off.
8. The Master takes the request for pilot and sends a start command to the pilot(s) requested by the fuel.
   a. All requests for Pilots from all fuels are “OR-ed” together.
9. Once the Pilot(s) are lit and ready, the Master sends that status to the requested fuel and the fuel continues with its light-off sequence.
   a. One step in the fuel light-off sequence may require that the pilot flame is removed to confirm the fuel can burn on its own.

   b. Certain burners may operate with the pilot(s) burning continuously to support stable combustion.
   c. Either way, it is up to the fuel configuration to tell the master whether a pilot should be on or off, and when.
10. If multiple fuels have been requested, the master repeats this process from Step 6 for the remaining fuels in order until all requested fuels are lit.
11. Once the requested fuel(s) are lit and stable, the Master indicates to the Interface that the burner is lit and ready for modulation.
    a. This enables the boiler, oven, or furnace to modulate the burner output for temperature control, steam pressure control, or other process action.
12. When the need for the burner's heat energy has passed, the interface sends a Stop request to the Master.
13. This is passed to the fuels and purge concurrently.
14. If a fuel component requires scavenging (purging of fuel from internal piping), the Fuel AOI goes into scavenging mode and again makes a request for pilot as required.
    a. The same sequence as before is performed, with the Master activating pilot(s) based on fuel request and passing pilot lit status back to the fuel as appropriate.
    b. Fuel Scavenging is a timed function and once the timer expires, the Fuel AOI removes the request for pilot and shuts itself off.
15. The Master is alerted once all fuels have shut off and all fuel blocking valves are confirmed closed. This information is passed on to the Purge component.
16. The Purge AOI goes into a Post-Firing purge mode.
    a. Timed purge
    b. The air actuator hold output is energized, holding air flow at its last value.
    c. Fuel modulating actuators are allowed to return to their idle position.
17. The Purge AOI indicates Post-Purge complete to the master
18. The Master indicates the burner is off to the Interface and air modulating devices can go to their idle position.
19. The Master returns to its idle state. The burner is idle.

Multi-Fuel Trip Management

Applications using burners that allow concurrent firing may require the burner to stay lit if one or more fuels experience an isolated safety trip. This function is accommodated as long as the trip is isolated to the fuel(s), and there is at least one active fuel that remains lit. When such an event occurs during modulation, the Master transitions to a Modulation Hold state to allow the Air/Fuel Ratio to stabilize following the abrupt drop in fuel flow. This action can be configured to hold fuel actuators at their current position or revert to light-off position and hold there. The operator must intervene to return the burner to modulation and is given three choices:

1. Reset the tripped fuel(s) and attempt a re-start.
2. De-Select the tripped fuel(s) and return to modulation with the remaining lit fuel(s).
3. Stop the burner.

Required Files

Objects include Add-On Instructions (AOIs) and HMI faceplates. The revision number (e.g. 1.01) used in filenames can change as new revisions are created.

Controller Files

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

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

A single routine import file is provided which includes both the BMS Master and BMS Purge instructions and related logic.

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

Device/Item	Add-On Instruction	ROUTINE Import
BMS Interface	raS_Opr_BMS_Master_1.00_AOI.L5X	raS_Opr_BMS_MasterPurge_1.00_SftyROUTINE.L5X

FactoryTalk® View HMI Files

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

Device/Item	Type	FactoryTalk® View SE Faceplate
BMSMaster Faceplate	Display	(raS-1_00-SE) raS_Opr_BMS_Master-Faceplate.gfx
Graphic Symbols	Global Object	(raS-1-SE) Graphic Symbols - BMS.ggfx
Toolbox	Global Object	(raS-1-SE) Toolbox - BMS.ggfx

Add-On Instruction I/O Data

The following tables identify raS_Opr_BMS_Master AOI parameters.

InOut Parameters

raS_Opr_BMS_Master AOI InOut Reference Parameters

Parameter Name	Data Type	Function
Ref_InterfaceData	raS_UDT_BMS_Master_Data	Data to and from all Burner Components
Ref_MFTR	raS_Dvc_BMS_D2S	Master Fuel Trip Relay Device Tag

Configuration Parameters

raS_Opr_BMS_Master AOI Configuration Parameters

Parameter Name	Data Type	Function
Cfg_HasProcessLimit_B	BOOL	Process Limit B Exists and Is In Use
Cfg_HasPurgeHold	BOOL	Purge Hold State Allowed
Cfg_VPDuringPurge	BOOL	Perform Pre-Light-Off Valve Proving During Pre-Purge
Cfg_WDChannels	DINT	Number of External Watchdog Channels (0, 1, or 2)
Cfg_FuelQty	DINT	Qty of Fuel Component AOIs (0-4). If 0, Pilot Components are treated as spark-ignited main fuels.
Cfg_HasHeader	BOOL	Fuel Supply Has Common Header
Cfg_PilotQty	DINT	Qty of Pilot Component AOIs (1-4)
Cfg_FiringMode	DINT	0=Alternate or Single Fuel 1=Transfer Only 2=Concurrent
Cfg_AnyFlameOK	BOOL	Any Flame OK for Light Off (Single Burner Only)
Cfg_HoldOnFuelTrip	BOOL	0=Trip Burner on Fuel Trip 1=Modulation Hold on FuelTrip
Cfg_HoldAtLightOff	BOOL	Burner to Light-Off on Modulation Hold
Cfg_PilotHoldA	BOOL	Light Pilot A in Pilot Hold State
Cfg_PilotHoldB	BOOL	Light Pilot B in Pilot Hold State
Cfg_PilotHoldC	BOOL	Light Pilot C in Pilot Hold State
Cfg_PilotHoldD	BOOL	Light Pilot D in Pilot Hold State
Cfg_HasPerm01	BOOL	Inp_Perm01 is Configured for Use
Cfg_HasPerm02	BOOL	Inp_Perm02 is Configured for Use
Cfg_HasPerm03	BOOL	Inp_Perm03 is Configured for Use
Cfg_HasPerm04	BOOL	Inp_Perm04 is Configured for Use
Cfg_HasIntlk01	BOOL	Inp_Intlk01 is Configured for Use
Cfg_HasIntlk02	BOOL	Inp_Intlk02 is Configured for Use
Cfg_HasIntlk03	BOOL	Inp_Intlk03 is Configured for Use
Cfg_HasIntlk04	BOOL	Inp_Intlk04 is Configured for Use
Cfg_HasIntlk05	BOOL	Inp_Intlk05 is Configured for Use
Cfg_HasIntlk06	BOOL	Inp_Intlk06 is Configured for Use
Cfg_HasIntlk07	BOOL	Inp_Intlk07 is Configured for Use
Cfg_HasIntlk08	BOOL	Inp_Intlk08 is Configured for Use

Input Parameters

raS_Opr_BMS_Master AOI Input Parameters

Parameter Name	Data Type	Function
Inp_Enable	BOOL	Burner Enable
Inp_TestEnable	BOOL	Enable Maintenance Test Mode
Inp_ProcessLimitA	BOOL	Required Process Limit Interlock
Inp_ProcessLimitB	BOOL	Process Limit Interlock B
Inp_EStopLocal	BOOL	Emergency Stop Push Button at Burner Location
Inp_EStopRemote	BOOL	Emergency Stop Push Button Remote from Burner Location
Inp_Perm01	BOOL	User Permissive Input 01

raS_Opr_BMS_Master AOI Input Parameters

Parameter Name	Data Type	Function
Inp_Perm02	BOOL	User Permissive Input 02
Inp_Perm03	BOOL	User Permissive Input 03
Inp_Perm04	BOOL	User Permissive Input 04
Inp_Intlk01	BOOL	User Interlock Input 01
Inp_Intlk02	BOOL	User Interlock Input 02
Inp_Intlk03	BOOL	User Interlock Input 03
Inp_Intlk04	BOOL	User Interlock Input 04
Inp_Intlk05	BOOL	User Interlock Input 05
Inp_Intlk06	BOOL	User Interlock Input 06
Inp_Intlk07	BOOL	User Interlock Input 07
Inp_Intlk08	BOOL	User Interlock Input 08
Inp_WD1	BOOL	External Watchdog Relay 1 Energized
Inp_WD2	BOOL	External Watchdog Relay 2 Energized

Control Output Parameters

raS_Opr_BMS_Master AOI Control Output Parameters

Parameter Name	Data Type	Function
Out_WD1	BOOL	Watchdog 1 Pulsed Output
Out_WD2	BOOL	Watchdog 2 Pulsed Output
Out_FuelAirA	BOOL	CCS Must Include Fuel A in Air/Fuel Ratio
Out_FuelAirB	BOOL	CCS Must Include Fuel B in Air/Fuel Ratio
Out_FuelAirC	BOOL	CCS Must Include Fuel C in Air/Fuel Ratio
Out_FuelAirD	BOOL	CCS Must Include Fuel D in Air/Fuel Ratio
Out_Modulate	BOOL	Burner System Release for Modulation
Val_eState	DINT	Burner Master Control State Value
Val_WDNextTest	REAL	Time Until Next Watchdog Relay Test (minutes)

Status and Information Output Parameters

raS_Opr_BMS_Master AOI Status and Information Output Parameters

Parameter Name	Data Type	Function
Sts_InitDN	BOOL	First Scan Initialization Complete
Sts_Enabled	BOOL	Burner Enabled
Sts_ER	BOOL	Burner Configuration Error
Sts_bCfgErr	DINT	Configuration Error Status (bit-mapped)
Sts_Faulted	BOOL	Burner Fault
Sts_bFaults	DINT	Fault Status Word (bit-mapped)
Sts_FuelSelectInv	BOOL	Fuel Selection is Invalid
Sts_FuelFail	BOOL	Fuel Control Sequence Encountered a Failure
Sts_bFuelFail	DINT	Fuel Failure Status (bit-mapped)
Sts_bWD	DINT	External Watchdog Control and Fault Status (bit-mapped)
Sts_PermsOK	BOOL	Burner Start Permissives OK
Sts_bPerms	DINT	Permissives Lost Status Word (bit-mapped)
Sts_IntlksOK	BOOL	All Burner System Interlocks are OK
Sts_bIntlks	DINT	Interlocks Lost Status Word (bit-mapped)
Sts_SafetyTrip	BOOL	Burner System Safety Trip
Sts_bSafetyTrip	DINT	Safety Trip First Out Capture (bit-mapped)
Sts_Ready	BOOL	Burner System Idle and Ready to Start

raS_Opr_BMS_Master AOI Status and Information Output Parameters

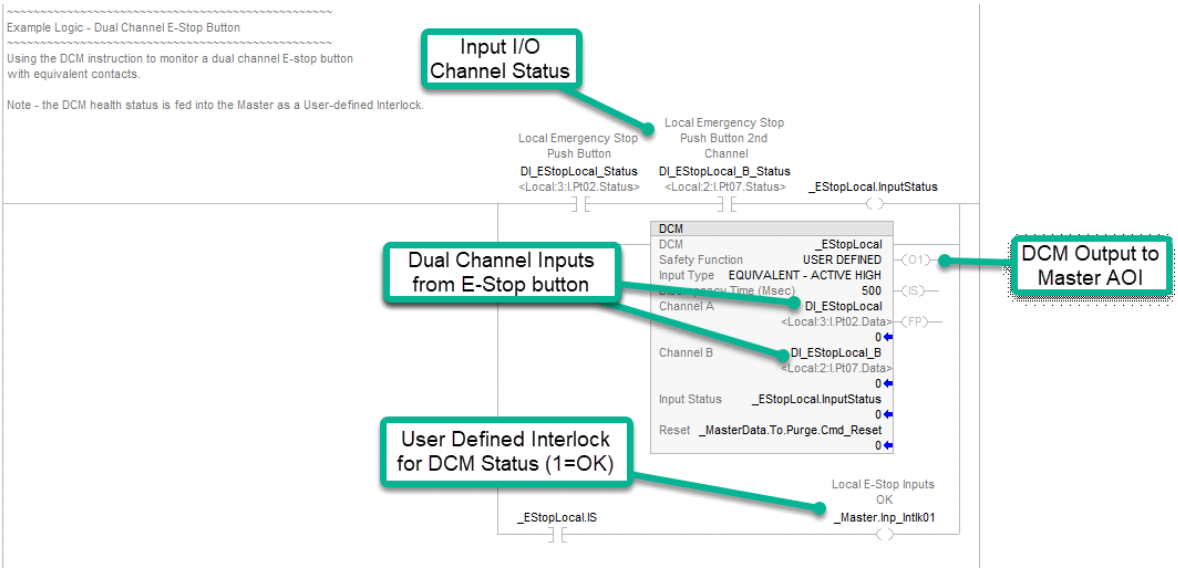
Parameter Name	Data Type	Function
Sts_Active	BOOL	Burner System Active
Sts_Normal	BOOL	Burner System NORMAL Mode
Sts_Test	BOOL	Burner System TEST ENABLE Mode
Sts_TestActive	BOOL	Burner System Device Testing Active
Sts_VPRequired	BOOL	At Least 1 Fuel or Pilot Requires Valve Proving Prior to Burner Light-Off
Sts_FuelOff	BOOL	Burner System All Fuel Confirmed Off
Sts_FuelOn	BOOL	Burner System Any Fuel Source Active
Sts_FlamePresent	BOOL	Burner System Any Flame Sensed
Sts_PurgeHoldReq	BOOL	Burner Hold after Pre-Purge Complete Requested
Sts_PurgeHold	BOOL	Burner Holding after Pre-Purge Complete
Sts_PilotHoldReq	BOOL	Burner Hold at Pilot Requested
Sts_PilotHold	BOOL	Burner Holding at Pilot
Sts_FuelHdrActive	BOOL	Burner Main Fuel Supply Header Active
Sts_FuelSelA	BOOL	Fuel A Selected
Sts_FuelPriA	BOOL	Fuel A is Primary
Sts_FuelActiveA	BOOL	Fuel A Active
Sts_FuelSelB	BOOL	Fuel B Selected
Sts_FuelPriB	BOOL	Fuel B is Primary
Sts_FuelActiveB	BOOL	Fuel B Active
Sts_FuelSelC	BOOL	Fuel C Selected
Sts_FuelPriC	BOOL	Fuel C is Primary
Sts_FuelActiveC	BOOL	Fuel C Active
Sts_FuelSelD	BOOL	Fuel D Selected
Sts_FuelPriD	BOOL	Fuel D is Primary
Sts_FuelActiveD	BOOL	Fuel D Active
Inf_Label	BOOL	Burner System Label (@Description used by HMI)
Inf_Tag	BOOL	Tag ID (@Description used by HMI)
Inf_BuildID	DINT	BMS Master AOI (raS_Opr_BMS_Master) Build ID

Programming Example

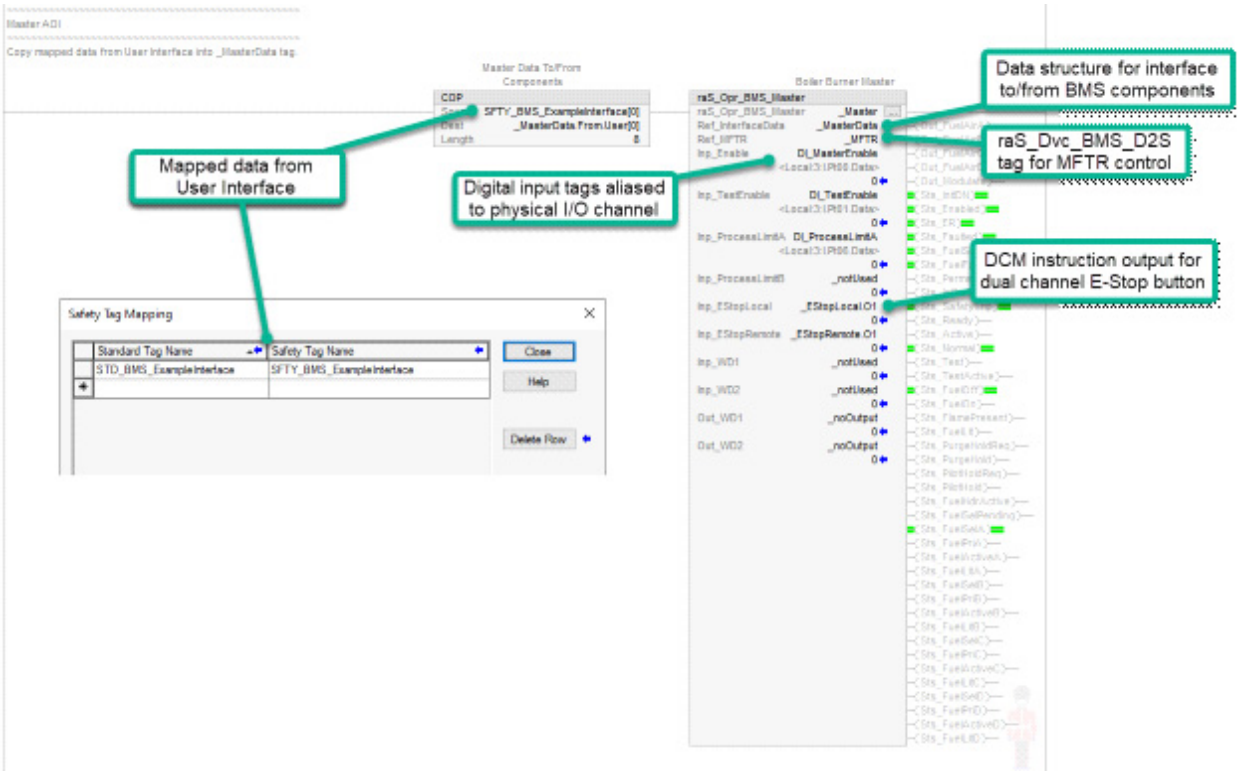
General arrangement for this example is shown in the . The raS_Opr_BMS_Master, being a Safety AOI, is located in the BMS_Example program, a_Master routine. The rungs that follow are taken from that routine:

- Rung 1: Dual Channel Local E-Stop interface using DCM instruction. Output from DMC is tied to required E-Stop input on the Master AOI, DCM status is tied to User Interlock to differentiate I/O faults from an E-stop event.
- Rung 2: Same as Rung 0, but for Remote E-Stop inputs.
- Rung 3: raS_Opr_BMS_Master AOI and supporting data copy. Transfer data received in mapped safety tag into *_MasterData* structure used for all data transfer between the Master and other BMS AOIs.

Dual Channel E-Stop Example



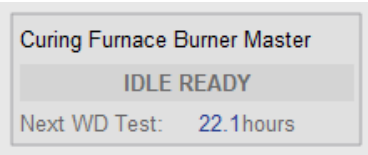
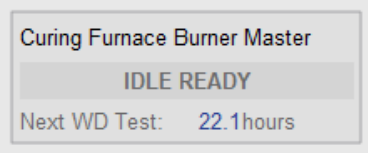
raS_Opr_BMS_Master AOI Example



Graphic Symbols

Graphic Symbols are used as launch buttons within HMI applications to open up faceplate displays. See See Basic Launch Button Attributes on page 22 section for details on configuration and indicators.

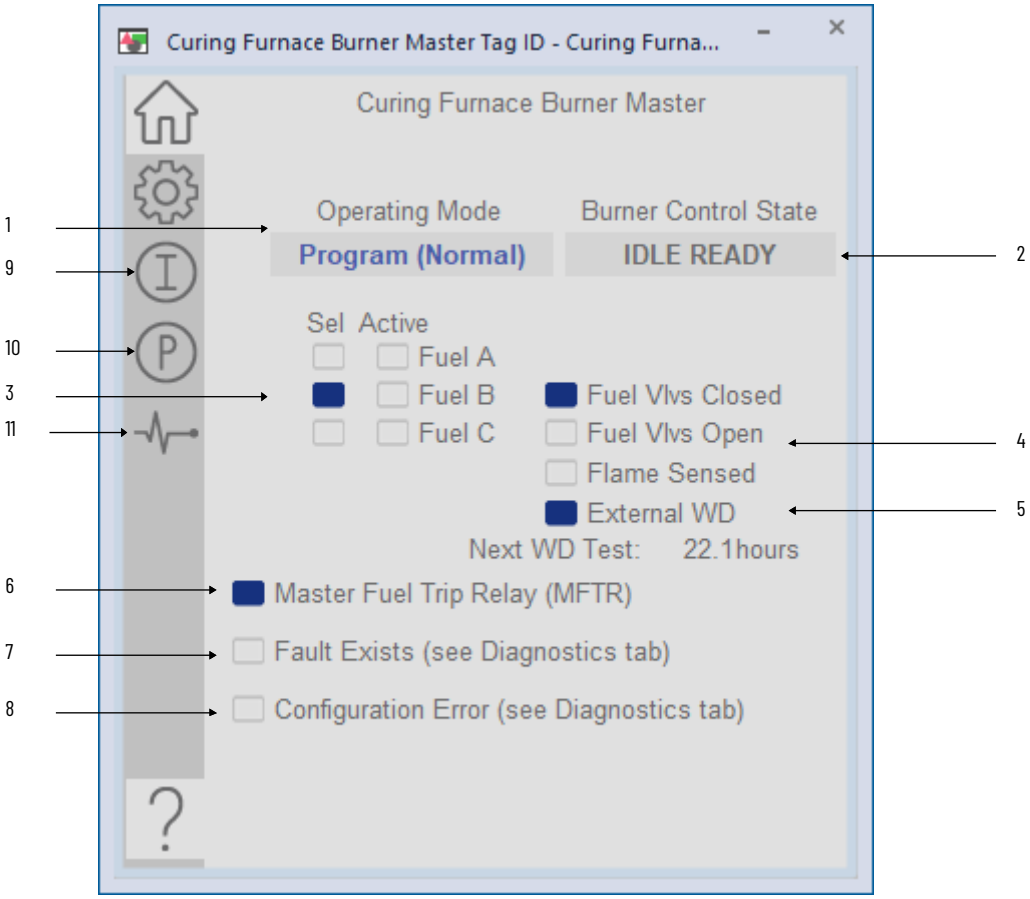
FactoryTalk® View SE Graphic Symbols

Graphic Symbol Name	Graphic Symbol	Description	Global Object Parameter Values
GO_MasterStatus		Master Status with faceplate navigation. This launch button graphic object allows the user to navigate to the object faceplate. The text on the button face is set to the tag's .Inf_Label.@Description field. Also shown are the operating state and time to next external watchdog relay test.	#102: AOI Instance (e.g. {::[PAC]Program::SafetyProgram._InstanceName}) #103: HMI Control Tag #120: Faceplate X Position, pixels from left edge of display (e.g. /X100) #121: Faceplate Y Position, pixels from top edge of display (e.g. Y100)
GO_MasterStatusNoNav		Master Status without navigation. . The text on the button face is set to the tag's .Inf_Label.@Description field. Also shown are the operating state and time to next external watchdog relay test.	#102: AOI Instance (e.g. {::[PAC]Program::SafetyProgram._InstanceName})

Faceplates

There are basic faceplate attributes that are common across all instructions. See Basic Faceplate Attributes on page 21.

Home

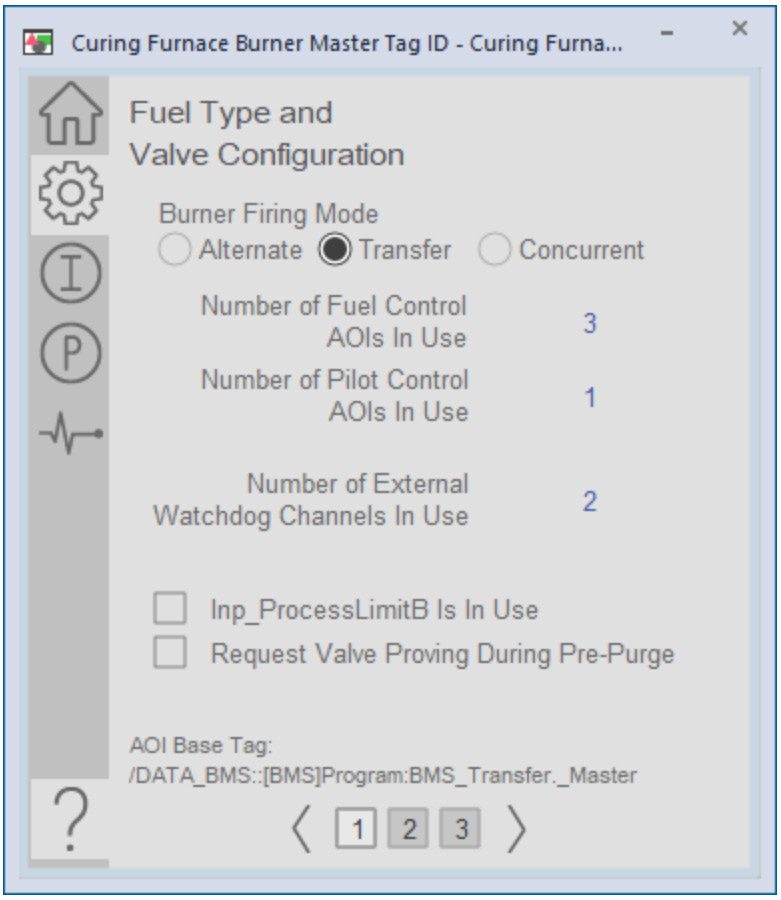


Item	Description
1	Burner Master Operating Mode indicator.
2	Burner Master Operating State indicator.
3	Fuel Selected and Active status indicators.
4	Burner Fuel Valve and Flame Status summary.
5	Burner Master External Watchdog Relay status and time until next relay test.
6	Master Fuel Trip Relay status indicator.
7	Burner Master Fault Status Summary indicator.
8	Burner Master Configuration status summary indicator.
9	Burner Master Interlock status summary indicator, showing all Interlock conditions OK.
10	Burner Master Permissive status summary indicator, showing all Permissive conditions OK.
11	Burner Master Diagnostic status summary indicator, showing all Diagnostic conditions OK.

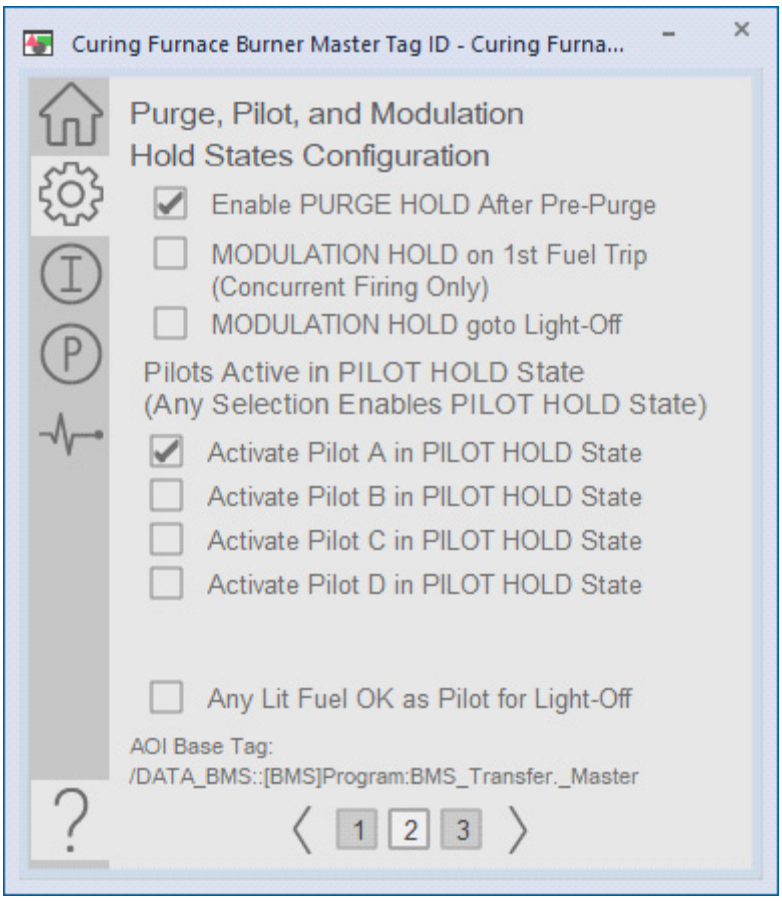
Configuration

There are three pages within the configuration tab.

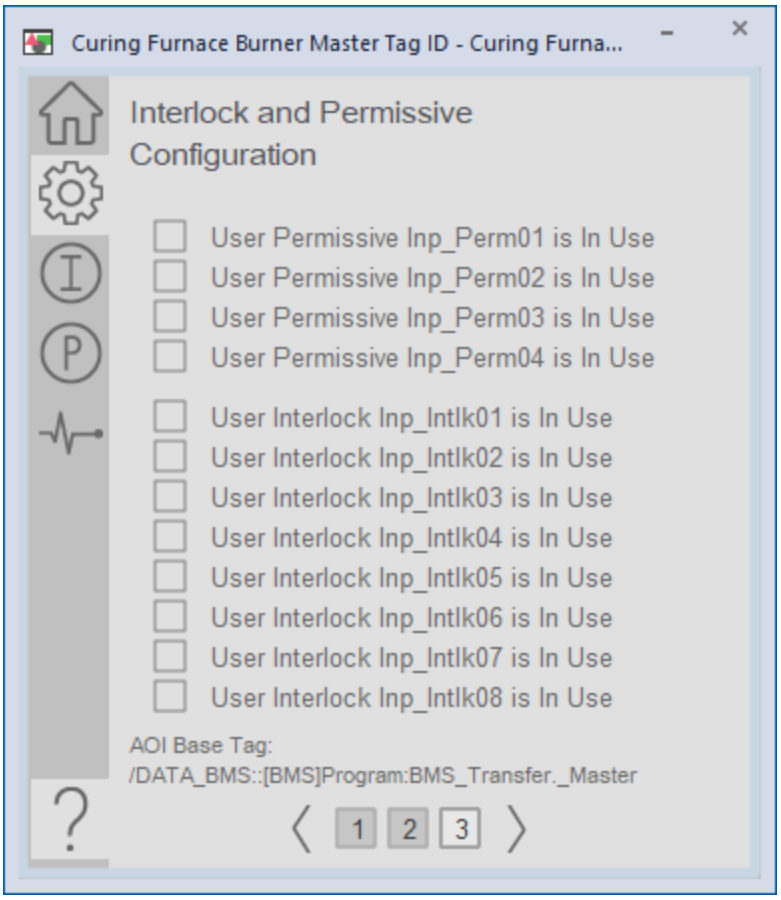
Configuration Page 1 - Fuel Type and Valve Configuration



Configuration Page 2 - Purge, Pilot, and Modulation Hold States Configuration



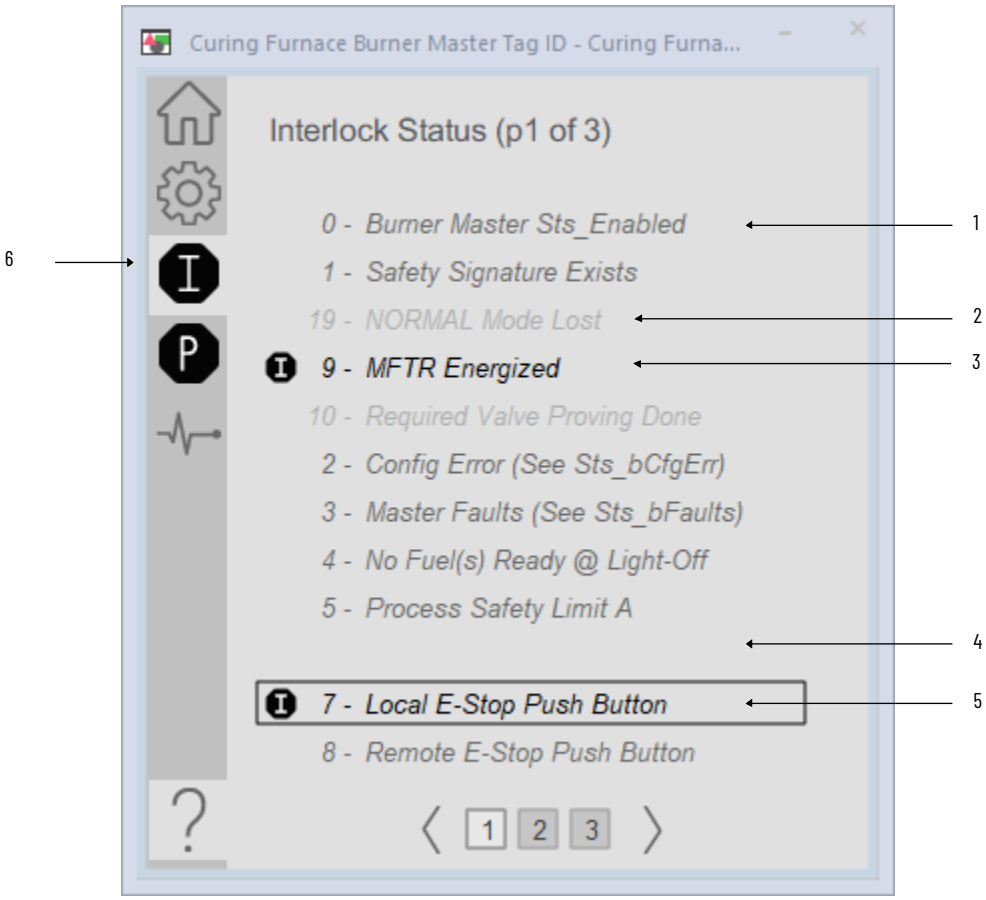
Configuration Page 3 - Interlock and Permissive Configuration



Interlock Status

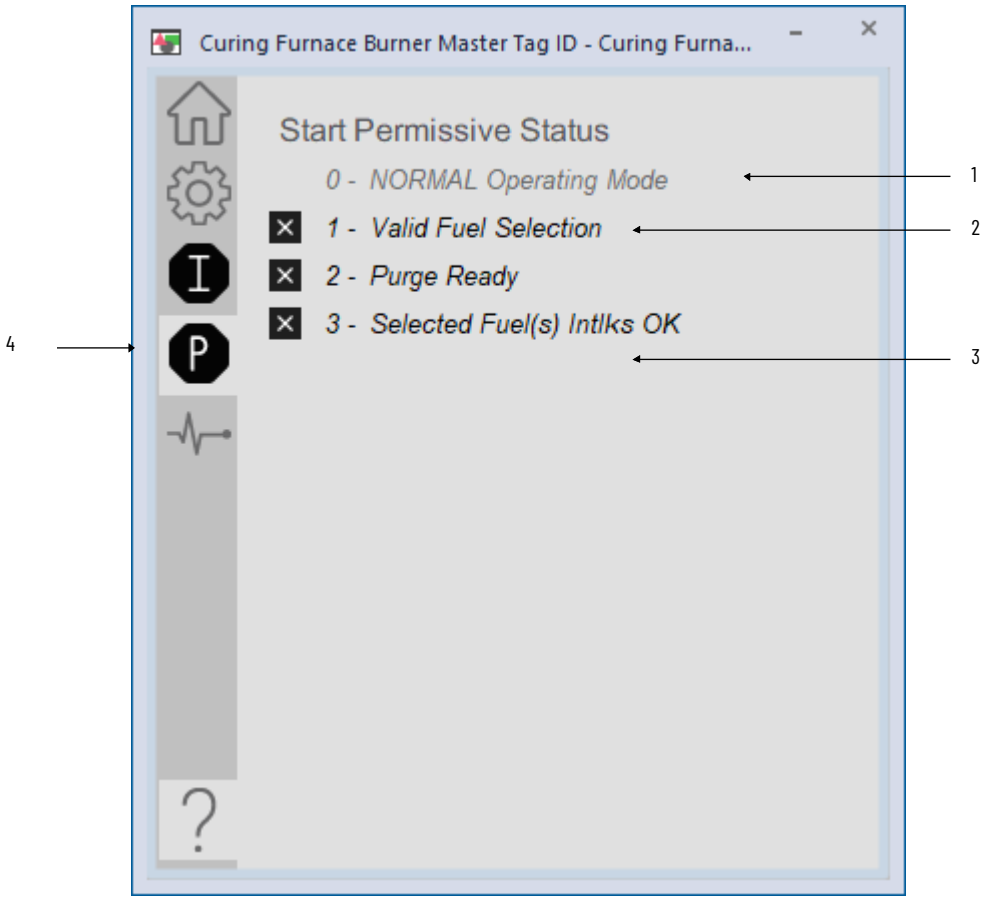
There are 3 pages within the Interlock Status tab.

Interlock Status Page 1



Item	Description
1	Interlock currently monitored, but OK to operate.
2	Interlock configured for use, but currently not monitored.
3	Interlock currently monitored, but NOT OK to operate.
4	Interlock not configured for use (Process Safety Limit B in this case).
5	Interlock currently monitored, NOT OK, and has generated a Safety Trip.
6	Summary indicator – at least one Interlock is NOT OK.

Permissive Status

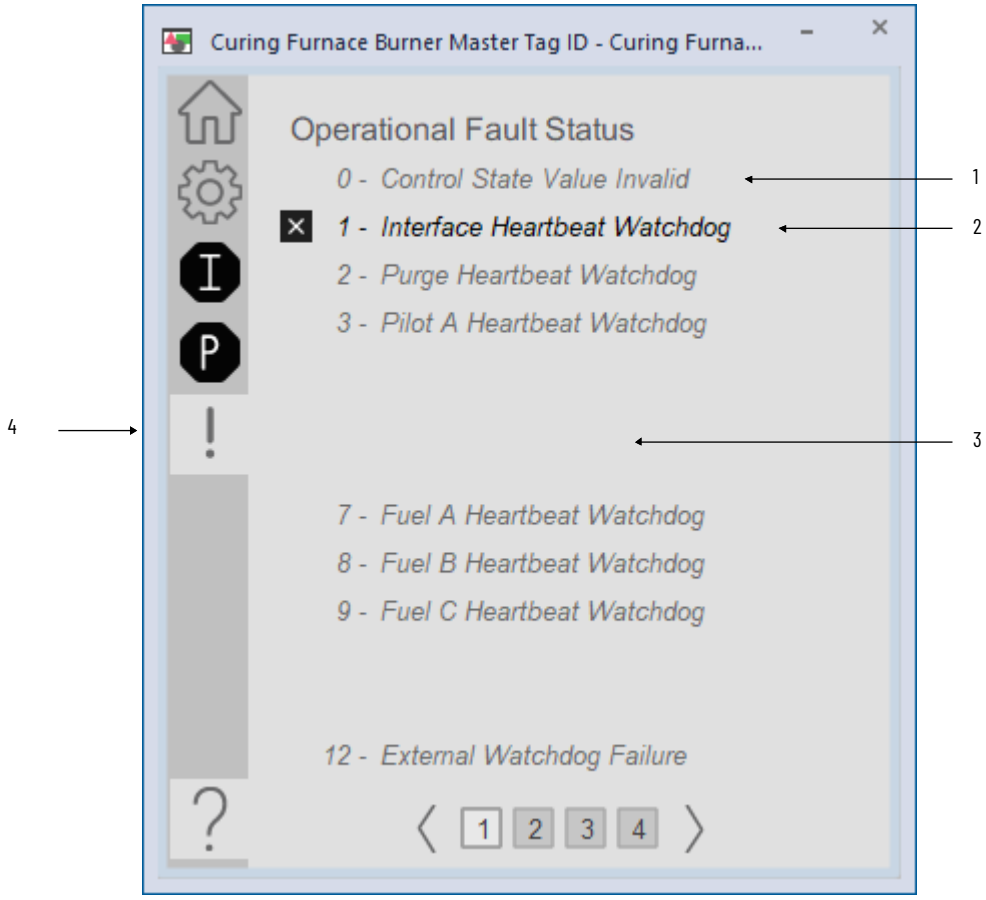


Item	Description
1	Permissive currently monitored, but OK to Start Operation.
2	Permissive currently monitored, but NOT OK to Start Operation.
3	Permissive not configured for use (User Permissive Input 1 in this case).
4	Summary indicator – at least one Permissive condition is NOT OK.

Diagnostics

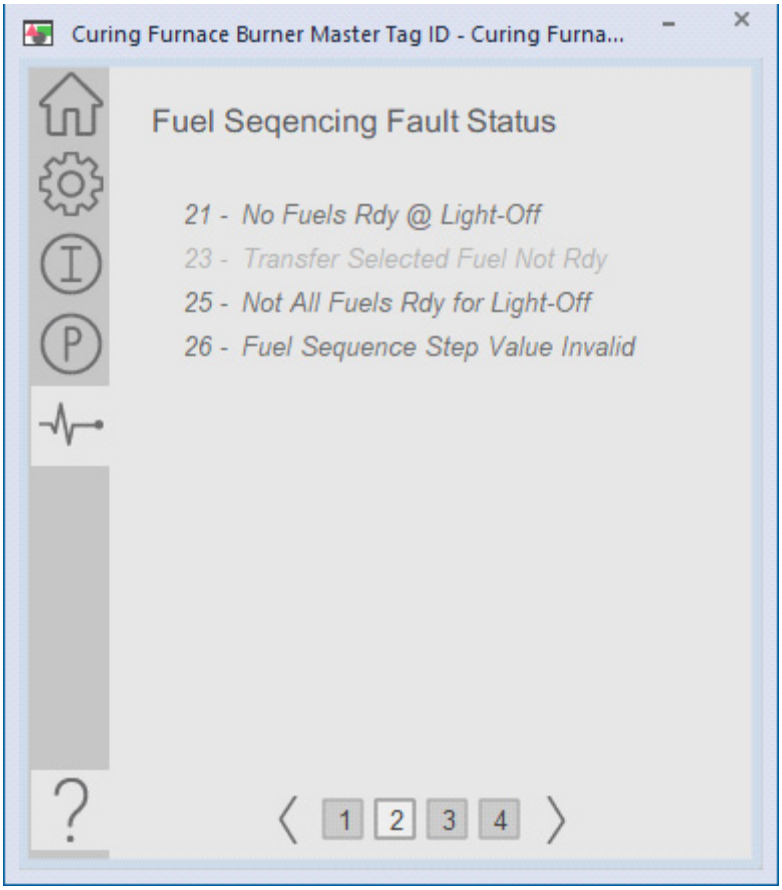
There are 4 pages within the diagnostics tab.

Diagnostics Page 1 - Operational Fault Status

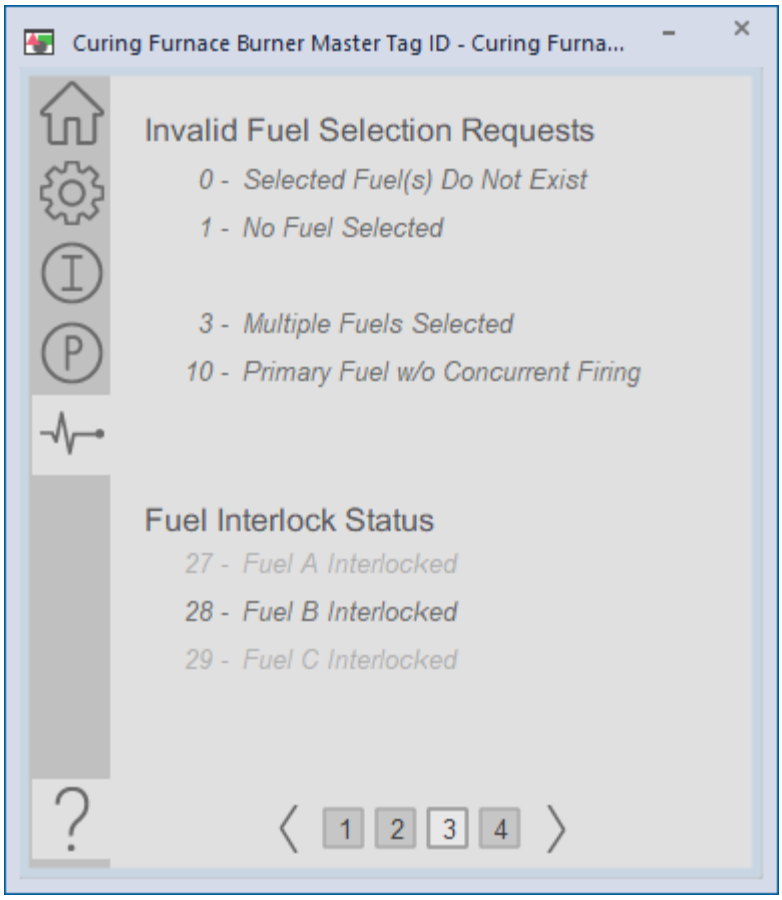


Item	Description
1	Operational Fault currently monitored, and OK to Operate.
2	Operational Fault currently monitored, and NOT OK to Operate.
3	Operational Fault not configured for use (Pilot B Heartbeat Watchdog in this case).
4	Summary indicator – at least one Diagnostic condition is NOT OK.

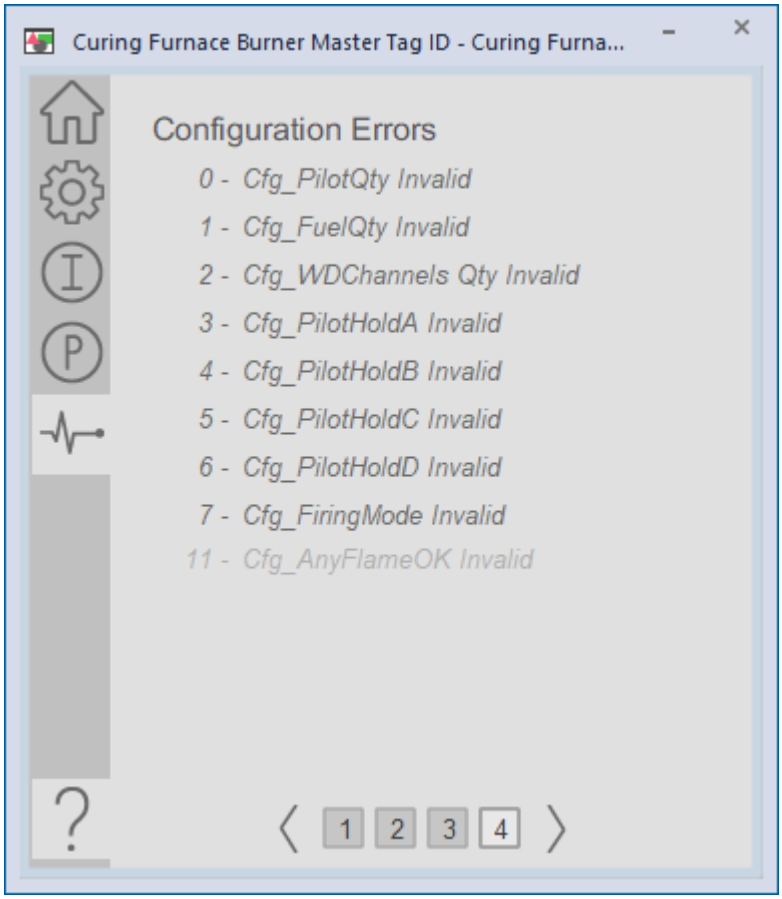
Diagnostics Page 2 - Fuel Sequencing Fault Status



Diagnostics page 3 – Invalid Fuel Selection and Fuel Interlock Status

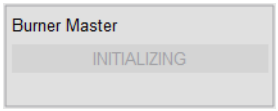
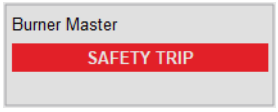
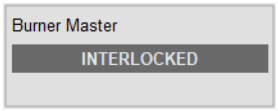
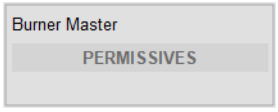


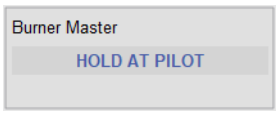
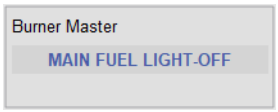
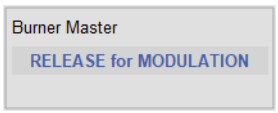
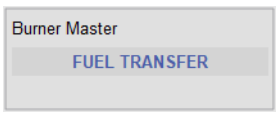
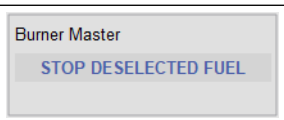
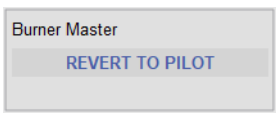
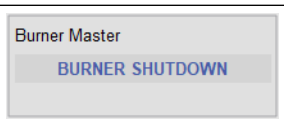
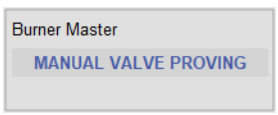
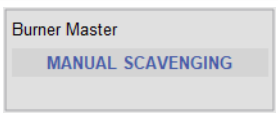
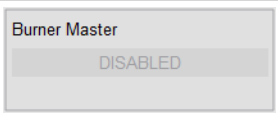
Diagnostics page 4 – Configuration Errors



State Value Enumerations

The following table identifies the states available to the Burner Master when enabled (Sts_Enabled is True). If Sts_Enabled is False, the Burner Master is allowed to only be in States 0, 1, and 2, and will indicate DISABLED.

Val_eState	State Name	Description	HMI Graphic
0	Initializing	AOI performing 1st scan initialization. (Enabled: Sts_Enabled = 1)	
1	Safety Trip	Master Safety Trip present. (Enabled: Sts_Enabled = 1)	
2	Interlocked	Master Interlock not met. (Enabled: Sts_Enabled = 1)	
3	Permissives	Master waiting for all Permissives OK.	
4	Idle / Ready	Master Idle and Ready for Start Command.	

Val_eState	State Name	Description	HMI Graphic
5	Pre-Ignition Purge	Purge AOI performing Pre-Ignition Purge	
6	Purge Hold	Burner in Purge and Hold, awaiting Start to continue.	
7	Hold at Pilot	Burner in Pilot Hold, awaiting Start to continue.	
8	Main Fuel Light-Off	Light selected main fuel(s).	
9	Release for Modulation	Burner Release for Modulation.	
11	Fuel Transfer	Transfer Fuel while lit.	
12	Stop Deselected Fuel	Extinguish De-Selected Fuel(s).	
13	Revert to Pilot	Extinguish all Main Fuels and revert to Pilot Hold.	
14	Burner Shutdown	Extinguish all Fuels and stop burner.	
15	Manual Valve Proving	Manual request for Valve Proving on selected Pilot/Fuel.	
16	Manual Fuel Scavenge	Manual request for Fuel Scavenge on selected Pilot/Fuel.	
0, 1, or 2		Master Disabled (Sts_Enabled = 0)	

BMS Purge (raS_Opr_BMS_Purge)

Overview

The Purge component AOI monitors all burner air flow and required fan status but does not control any devices directly. A burner may have but one constant speed fan with a single damper, or the system may have multiple fans with multiple dampers controlling air flow. The Purge does not control air flow directly but monitors interlock switches to ensure the required fan(s) are running and the burner has at least a minimum required air flow to support combustion.

Limitations

The following limitations must be considered when implementing this instruction within a BMS Library application.

AOI Class	Safety
Minimum Instances Required	1
Maximum Instances Allowed	1

System Interaction

The Purge AOI exchanges data with the BMS Master AOI using a pair of UDT InOut references, one for data in each direction. The content of these UDTs is detailed in the following sections.

Purge to Master Data

Data Type: raS_UDT_BMS_Purge_ToMaster (12 bytes).

Purge Parameter Reference: Ref_ToMaster (InOut)

Data from Purge to Master includes the common component data structure, purge operational status, and command acknowledge bits. A list of UDT elements is shown in the following table.

Purge to Master Data Elements

Element Name	Data Type	Description
Ref_ToMaster	raS_UDT_BMS_Purge_ToMaster	Purge Component Data to Master
Ref_ToMaster.Sts	raS_UDT_BMS_CommonStatus	Common Component Status to Master
Ref_ToMaster.Sts.Enabled	BOOL	Purge Component Enabled
Ref_ToMaster.Sts.SafetyTrip	BOOL	Purge Component Generated Safety Trip
Ref_ToMaster.Sts.IntlkOK	BOOL	Purge Component Interlocks OK
Ref_ToMaster.Sts.Ready	BOOL	Purge Component Ready for Start
Ref_ToMaster.Sts.Active	BOOL	Purge Component Operating Sequence is Active
Ref_ToMaster.Sts.Normal	BOOL	Purge Component in NORMAL MODE
Ref_ToMaster.Sts.TestMode	BOOL	Purge Component in TEST ENABLE MODE

Purge to Master Data Elements

Element Name	Data Type	Description
Ref_ToMaster.Sts.TestActive	BOOL	Purge Component in TEST ENABLE MODE with Device Test Active
Ref_ToMaster.Sts.eState	SINT	Purge Component Enumerated State Value
Ref_ToMaster.Sts.HBEcho	INT	Master Communication Heartbeat Echo
Ref_ToMaster.Sts.BuildID (hidden parameter)	DINT	raS_Opr_BMS_Purge AOI Build ID
Ref_ToMaster.Sts_Purging	BOOL	Purge Component Pre-Ignition Purge In Progress
Ref_ToMaster.Sts_PurgeDN	BOOL	Purge Component Pre-Ignition Purge Complete
Ref_ToMaster.Sts_PostPurge	BOOL	Purge Component Post-Firing Purge In Progress
Ref_ToMaster.Sts_PostPurgeDN	BOOL	Purge Component Post-Firing Purge Complete
Ref_ToMaster.Sts_AtLightOff	BOOL	Purge Component Air Actuator(s) at Light-Off Position
Ref_ToMaster.Ack_Start	BOOL	Purge Component Acknowledge Start Request
Ref_ToMaster.Ack_Stop	BOOL	Purge Component Acknowledge Stop Request
Ref_ToMaster.Ack_Reset	BOOL	Purge Component Acknowledge Reset Request

Master to Purge Data

Data Type: raS_UDT_BMS_Master_ToPurge (8 bytes).

Purge Parameter Reference: Ref_FromMaster (InOut)

Data from Master to Purge includes the SysSts structure, start/stop/reset commands, and information gathered from pilot and fuel AOIs necessary for purge operation. A list of UDT elements is shown in the following table.

Master to Purge Data Elements

Element Name	Data Type	Description
Ref_FromMaster	raS_UDT_BMS_Master_toPurge	Data from Master to Purge Component.
Ref_FromMaster.SysSts	raS_UDT_BMS_Master_SysSts	System-wide common status information.
Ref_FromMaster.SysSts.Enable	BOOL	Burner System Enable
Ref_FromMaster.SysSts.IntlksOK	BOOL	Burner System Interlocks OK
Ref_FromMaster.SysSts.MFTR	BOOL	Master Fuel Trip Relay Energized
Ref_FromMaster.SysSts.Normal	BOOL	Burner System in NORMAL Mode
Ref_FromMaster.SysSts.TestMode	BOOL	Burner System in TEST ENABLE Mode
Ref_FromMaster.SysSts.TestActive	BOOL	Burner in TEST ENABLE Mode with a Device Test Active
Ref_FromMaster.SysSts.NoFlame	BOOL	Pre-Purge NOT Completed – No Flame Allowed
Ref_FromMaster.SysSts.eState	SINT	Burner System Master State Value
Ref_FromMaster.SysSts.HB	INT	Master Communication Heartbeat
Ref_FromMaster.Cmd_Start	BOOL	Purge Sequence Start Request
Ref_FromMaster.Cmd_Stop	BOOL	Purge Sequence Stop Request
Ref_FromMaster.Cmd_Reset	BOOL	Purge Reset Request
Ref_FromMaster.Inp_AllFuelPoC	BOOL	Summation of All Fuel & Pilot PoC Status
Ref_FromMaster.Inp_AnyFuelActive	BOOL	Any Fuel or Pilot Active
Ref_FromMaster.Inp_AtLightOff	BOOL	All Fuel Actuators At Light-Off
Ref_FromMaster.Inp_FlamePresent	BOOL	Any Flame Sensed
Ref_FromMaster.Inp_ActuatorHold	BOOL	Air Actuator(s) Hold Current Position
Ref_FromMaster.Inp_ActuatorRel	BOOL	Air Actuator(s) Released for Modulation

Purge AOI Configuration

The Purge AOI determines the times associated with Pre-Ignition and Post-Firing Purges, fans required and fault delays for fan starts, and which air flow sensors are required. The Purge AOI provides up to 8 user-assignable interlock

inputs and up to 4 user-assignable permissive inputs that are enabled by configuration. Configuration parameter details are shown below.

Interlocks

The raS_Opr_BMS_Purge AOI monitors 28 interlock conditions, 8 of which are user selectable. The following table identifies each interlock condition, when each is monitored, and when each will trigger a Safety Trip.

Purge AOI Interlock List

Interlock Condition	Interlock Enable	Safety Trip Enable
Purge Component Disabled	Always	Purge Component Active
Master Comm Heartbeat Watchdog Timeout	Always	Purge Component Active
System Interlock Exists	Always	Never
Normal Mode Lost (Sts_Normal = 0)	Purge Component Active and System Interlocks OK	Same as Interlock Enable
Configuration Error Exists (Sts_bCfgErr <> 0)	Purge Component Enabled	Purge Component Active
Invalid State Value	Purge Component Enabled	Purge Component Active
Fuel Proof of Closure Lost	Purge States 5 through 8 inclusive (Active but Pre-Purge Not Complete)	Same as Interlock Enable
Premature Flame Sense	Purge States 5 through 8 inclusive (Active but Pre-Purge Not Complete)	Same as Interlock Enable
Loss of Combustion Air Flow	All Fans On	Same as Interlock Enable
Loss of Exhaust Air Flow	All Fans On and Has Exhaust Switch	Same as Interlock Enable
Fuel Actuators Not At Light-Off Position	Purge State 5	Same as Interlock Enable
Loss of Purge Air Flow	Timed Pre-Purge State 7	Same as Interlock Enable
Purge Credit Expired	Purge Component Enabled	Same as Interlock Enable
Loss of Combustion Fan (Inp_CombAirFan = 0)	Purge Active and Fans Required	Same as Interlock Enable
Loss of Primary/Supply Fan (Inp_SupplyFan = 0)	Purge Active, Fans Required, and Has Supply Fan	Same as Interlock Enable
Loss of Exhaust Fan (Inp_ExhaustFan = 0)	Purge Active, Fans Required, and Has Exhaust Fan	Same as Interlock Enable
Any Fan Fail to Start Fault	Purge Component Enabled	Same as Interlock Enable
Combustion Air Switch Failed to Open Fault	Purge Component Enabled	Same as Interlock Enable
Exhaust Air Switch Failed to Open Fault	Purge Component Enabled and Has Exhaust Switch	Same as Interlock Enable
Purge Air Flow Switch Failed to Open	Purge Component Enabled	Same as Interlock Enable
User Interlock 01 Not OK (Inp_Intlk01 = 0)	Purge Component Enabled and Has User Interlock 01	Purge Component Active and Has User Interlock 01
User Interlock 02 Not OK (Inp_Intlk02 = 0)	Purge Component Enabled and Has User Interlock 02	Purge Component Active and Has User Interlock 02
User Interlock 03 Not OK (Inp_Intlk03 = 0)	Purge Component Enabled and Has User Interlock 03	Purge Component Active and Has User Interlock 03
User Interlock 04 Not OK (Inp_Intlk04 = 0)	Purge Component Enabled and Has User Interlock 04	Purge Component Active and Has User Interlock 04
User Interlock 05 Not OK (Inp_Intlk05 = 0)	Purge Component Enabled and Has User Interlock 05	Purge Component Active and Has User Interlock 05
User Interlock 06 Not OK (Inp_Intlk06 = 0)	Purge Component Enabled and Has User Interlock 06	Purge Component Active and Has User Interlock 06
User Interlock 07 Not OK (Inp_Intlk07 = 0)	Purge Component Enabled and Has User Interlock 07	Purge Component Active and Has User Interlock 07
User Interlock 08 Not OK (Inp_Intlk08 = 0)	Purge Component Enabled and Has User Interlock 08	Purge Component Active and Has User Interlock 08

Permissives

The raS_Opr_BMSPurge AOI monitors 9 permissive conditions, 4 of which are user selectable. The following table identifies each permissive condition and when each is monitored. Once the Purge AOI transitions to an Active state, permissive conditions no longer impact operation.

Purge AOI Permissive List

Permissive Lost Condition	Permissive Enable
Not NORMAL Operating Mode (Sts_Normal = 0)	Purge Component Not Active
MFTR Energized	Purge Component Not Active
Fuel Proof of Closure Lost	Purge Component Not Active
Fuel and Air Actuators Not at Light-Off	Purge Component Not Active

Purge AOI Permissive List

Permissive Lost Condition	Permissive Enable
Flame Present	Purge Component Not Active
User Permissive 01 Not OK (Inp_Perm01 = 0)	Purge Component Not Active and Has User Permissive 01
User Permissive 02 Not OK (Inp_Perm02 = 0)	Purge Component Not Active and Has User Permissive 02
User Permissive 03 Not OK (Inp_Perm03 = 0)	Purge Component Not Active and Has User Permissive 03
User Permissive 04 Not OK (Inp_Perm04 = 0)	Purge Component Not Active and Has User Permissive 04

Operations

Operation Modes

The BMS Library objects provide only physical operation mode. There is no virtual/simulated mode offered.

Normal Operation Mode

The Master must be in NORMAL Mode to operate the burner, this mode status is transmitted to the Purge Component. While in NORMAL Mode, the Purge Component receives requests from the Master, making appropriate operational changes based on the command received. There is no direct interaction with the Purge Component except through the Master's Data Interface.

NORMAL Mode status is echoed back to the Master indicating whether the Purge Component is in NORMAL Mode or not.

Test Enable Mode Operation

The Purge Component is in TEST ENABLE Mode any time the Master is in TEST ENABLE Mode. Any commands to start or otherwise activate the Purge Component are ignored.

TEST ENABLE Mode status is echoed back to the Master indicating whether the Purge Component is in TEST ENABLE Mode or not.

Device Test Active

Device Test Active indicates a device under control of the Purge Component is actively being tested. Since the Purge Component has no devices under its control, the Purge Component will not indicate Device Test Active.

Operation

The Purge component controls two purge activities: Pre-Ignition Purge (Pre-Purge) and Post-Firing Purge (Post-Purge), both intended to remove unburned fuel vapors from the combustion chamber, minimizing the probability of an uncontrolled explosive atmosphere within the burner chamber.

Pre-Purge is performed at the beginning of a burner light-off cycle, triggered by request from the Master AOI.

1. Required fan motors are started if not already running.
2. Once fan motor interlocks are satisfied, the Purge component makes a request for Pre-Purge air flow.
3. The burner's combustion control system (CCS) puts all air actuators in the proper positions to achieve the necessary air flow for Pre-Purge.
4. Purge air flow and/or damper switch interlocks confirm Pre-Purge air flow has been achieved.
5. Pre-Purge timing begins.
   a. Pre-Purge time duration is configurable and must provide sufficient time to ensure a required number of complete air changes in the entire burner chamber, including all recirculating and exhaust ducting.
   b. If any of the pre-purge confirmation interlocks is lost, the Purge component generates a safety trip.
6. Once the timer expires (without air flow loss) the Purge component indicates Pre-Purge complete to the Master.
7. The request for Pre-Purge air flow is removed, allowing the air flow modulators to return to their light-off positions.

Upon completion of the Pre-Purge sequence, the Purge component returns to a monitoring condition, ensuring that fans stay running and minimum combustion air flow is maintained. If the switch indicating air flow is lost, or if any required fan motor drops out, a safety trip is generated, and the burner is shut down.

A post-purge is performed once the burner has been shut down by request from Interface (normal shut-down) or by interlock loss (safety trip). The post-purge commences once fuel valves are confirmed closed and flame confirmed extinguished the Purge Component performs a Post-Firing Purge (Post-Purge).

1. The Purge component indicates Post-Purge active to the Master.
   a. Air modulation hold is requested, maintaining current air flow.
   b. Fans are monitored, and if lost, Post-purge time may be adjusted depending on the burner application and Purge component configuration.
2. Post-Purge is timed.
3. On timer expiration, the air modulation hold is removed, allowing air actuator to return to their idle position.
4. Post-Purge is indicated complete to the Master.
5. Purge component returns to idle state.

Required Files

Objects include Add-On Instructions (AOIs) and HMI faceplates. The revision number (e.g. 1.01) used in filenames can change as new revisions are created.

Controller Files

Add-On Instructions are reusable code objects that contain encapsulated logic that can streamline implementing your system. This lets you create your own

instruction set for programming logic as a supplement to the instruction set provided natively in the ControlLogix® firmware. An Add-On Instruction is defined once in each controller project, and can be instantiated multiple times in your application code as needed.

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

A single routine import file is provided which includes both the BMS Master and BMS Purge instructions and related logic.

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

Device/Item	Add-On Instruction	ROUTINE Import
BMS Interface	raS_Opr_BMS_Master_1.00_AOI.L5X	raS_Opr_BMS_MasterPurge_1.00_SftyROUTINE.L5X

FactoryTalk® View HMI Files

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

Device/Item	Type	FactoryTalk® View SE Faceplate
BMSMaster Faceplate	Display	(raS-1_00-SE) raS_Opr_BMS_Purge-Faceplate.gfx
Graphic Symbols	Global Object	(raS-1-SE) Graphic Symbols - BMS.ggfx
Toolbox	Global Object	(raS-1-SE) Toolbox - BMS.ggfx

Add-On Instruction I/O Data

The following tables identify raS_Opr_BMS_Purge AOI parameters.

InOut Parameters

raS_Opr_BMS_Purge AOI InOut Reference Parameters

Parameter Name	Data Type	Function
Ref_FromMaster	raS_UDT_BMS_Master_ToPurge	Command Data to Purge Control
Ref_ToMaster	raS_UDT_BMS_Purge_ToMaster	Status Data From Purge Control

Configuration Parameters

raS_Opr_BMS_Purge AOI Configuration Parameters

Parameter Name	Data Type	Function
Cfg_HasSupplyFan	BOOL	Primary/Supply Fan Exists and is Used
Cfg_HasExhaustFan	BOOL	Exhaust Fan Exists and is Used
Cfg_HasExhaustSw	BOOL	Minimum Exhaust Air Flow Confirmation Switch Exists
Cfg_DeBounceTime	REAL	Air Flow Switch De-Bounce TIme (seconds)
Cfg_FlowSwReleaseTime	REAL	Fan(s) Off Flow Switch Release Delay (seconds)
Cfg_FanStartDelay	REAL	Delay After Fan Start for Air Flow (seconds)
Cfg_FanFailToStartDelay	REAL	Air Fan Fail to Start Fault Delay (seconds)
Cfg_HasT1400Bypass	BOOL	Bypass Pre-Purge if Temp>1400F
Cfg_PrePurgeTime	REAL	Pre-Ignition Purge Time (minutes)
Cfg_PurgeCreditTime	REAL	Pre-Purge Expiration Time (minutes)
Cfg_PostPurgeTime	REAL	Post-Firing Purge Time (seconds)
Cfg_NatDraftPost	BOOL	Increase Time for Fans-Off Post-Purge
Cfg_NatDraftTime	REAL	Natural Draft Post-Purge Time (minutes)
Cfg_HasPerm01	BOOL	User Permissive Input 1 In Use
Cfg_HasPerm02	BOOL	User Permissive Input 2 In Use
Cfg_HasPerm03	BOOL	User Permissive Input 3 In Use
Cfg_HasPerm04	BOOL	User Permissive Input 4 In Use
Cfg_HasIntlk01	BOOL	User Interlock Input 1 In Use
Cfg_HasIntlk02	BOOL	User Interlock Input 2 In Use
Cfg_HasIntlk03	BOOL	User Interlock Input 3 In Use
Cfg_HasIntlk04	BOOL	User Interlock Input 4 In Use
Cfg_HasIntlk05	BOOL	User Interlock Input 5 In Use
Cfg_HasIntlk06	BOOL	User Interlock Input 6 In Use
Cfg_HasIntlk07	BOOL	User Interlock Input 7 In Use
Cfg_HasIntlk08	BOOL	User Interlock Input 8 In Use

Input Parameters

raS_Opr_BMS_Purge AOI Input Parameters

Parameter Name	Data Type	Function
Inp_Enable	BOOL	Enable Purge Control AOI
Inp_CombAirOK	BOOL	Minimum Combustion Air Flow Confirmation
Inp_PurgeAirOK	BOOL	Purge Air Flow Confirmation
Inp_ExhAirOK	BOOL	Minimum Exhaust Air Flow Confirmation
Inp_Above1400F	BOOL	Combustion Chamber Temperature Exceeds 1400F (760C)
Inp_SupplyFan	BOOL	Primary/Supply Fan Running
Inp_CombAirFan	BOOL	Secondary/Combustion Air Fan Running
Inp_ExhaustFan	BOOL	Exhaust Fan Running
Inp_AtLightOff	BOOL	Air Actuator(s) at Light-Off Position
Inp_Perm01	BOOL	User Permissive Input 1

raS_Opr_BMS_Purge AOI Input Parameters

Parameter Name	Data Type	Function
Inp_Perm02	BOOL	User Permissive Input 2
Inp_Perm03	BOOL	User Permissive Input 3
Inp_Perm04	BOOL	User Permissive Input 4
Inp_Intlk01	BOOL	User Interlock Input 1
Inp_Intlk02	BOOL	User Interlock Input 2
Inp_Intlk03	BOOL	User Interlock Input 3
Inp_Intlk04	BOOL	User Interlock Input 4
Inp_Intlk05	BOOL	User Interlock Input 5
Inp_Intlk06	BOOL	User Interlock Input 6
Inp_Intlk07	BOOL	User Interlock Input 7
Inp_Intlk08	BOOL	User Interlock Input 8

Control Output Parameters

raS_Opr_BMS_Purge AOI Control Output Parameters

Parameter Name	Data Type	Function
Out_RunFans	BOOL	Run All Air Fans
Out_LightOff	BOOL	Air Actuators to Light-Off Position
Out_Purge	BOOL	Air Actuators to Purge Position
Out_Hold	BOOL	Air Actuators Hold at Current Position
Out_Release	BOOL	Air Actuators Release for Modulation
Val_eState	DINT	Purge Control State Engine Enumeration Value
Val_TimeRemaining	REAL	Current Activity Time Remaining (seconds)

Status and Information Output Parameters

raS_Opr_BMS_Purge AOI Status and Information Output Parameters

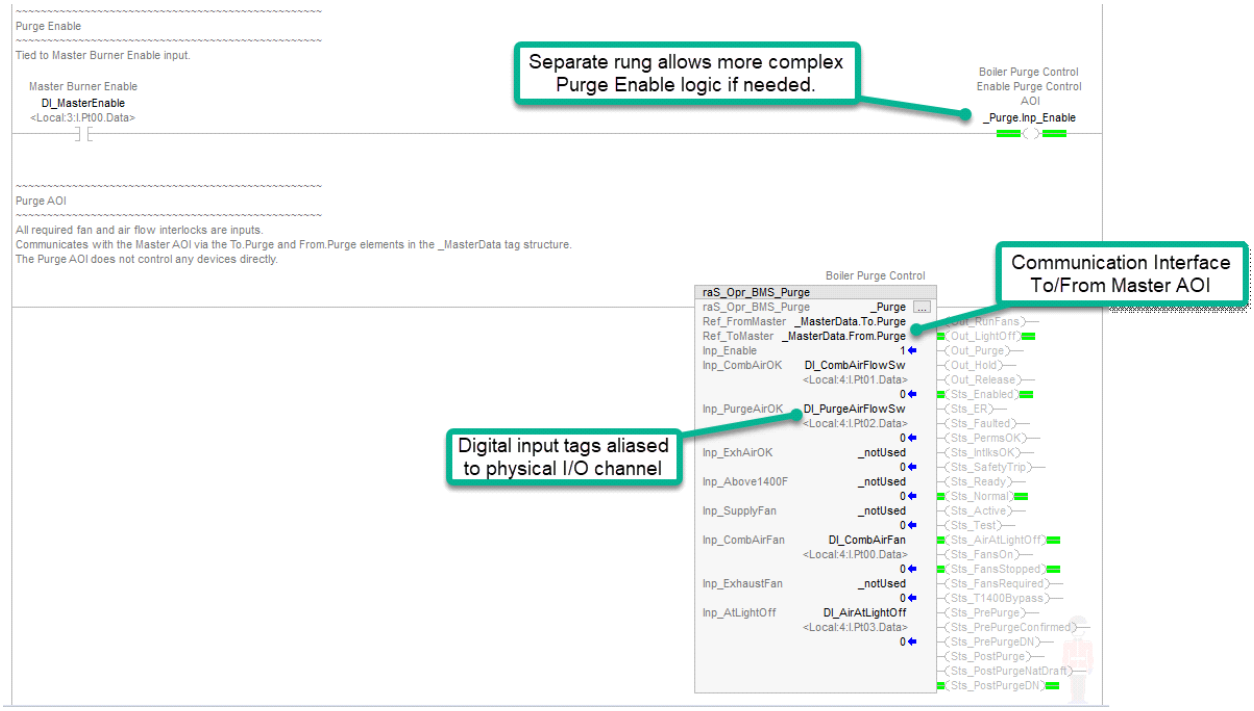
Parameter Name	Data Type	Function
Sts_Enabled	BOOL	Purge Control Enabled
Sts_ER	BOOL	Configuration Error Operation is Disabled
Sts_bCfgErr	DINT	Configuration Error Status (bit-mapped)
Sts_Faulted	BOOL	Operational Fault Exists
Sts_bFaults	DINT	Fault Status Word (bit-mapped)
Sts_PermsOK	BOOL	Purge Sequence Permissives OK
Sts_bPerms	DINT	Permissives Lost Status Word (bit-mapped)
Sts_IntlksOK	BOOL	All Interlocks OK
Sts_bIntlks	DINT	Interlocks Lost Status Word (bit-mapped)
Sts_SafetyTrip	BOOL	Interlock Safety Trip Reset Required
Sts_bSafetyTrip	DINT	Safety Trip First Out Capture (bit-mapped)
Sts_Ready	BOOL	Ready To Operate
Sts_Normal	BOOL	Normal Operating Mode
Sts_Active	BOOL	Purge Control Active
Sts_Test	BOOL	Maintenance Test Mode Enabled
Sts_AirAtLightOff	BOOL	Air Actuator(s) at Light-Off Position
Sts_FansOn	BOOL	All Fans are Running
Sts_FansStopped	BOOL	Fans are Not Running and Rotation Stop Delay Timer Done
Sts_FansRequired	BOOL	All Fans Required for Burner Operation
Sts_T1400Bypass	BOOL	Combustion Chamber Exceeds 1400F (760C) Pre-Purge Bypassed
Sts_PrePurge	BOOL	Burner Pre-Ignition Purge In Progress
Sts_PrePurgeConfirmed	BOOL	Pre-Purge Air Flow Confirmed
Sts_PrePurgeDN	BOOL	Burner Pre-Ignition Purge Complete Enable MFTR
Sts_PostPurge	BOOL	Burner Post-Firing Purge In Progress
Sts_PostPurgeNatDraft	BOOL	Burner Post-Purge in Natural Draft Timing Mode
Sts_PostPurgeDN	BOOL	Burner Post-Firing Purge Complete
Rdy_Reset	BOOL	Fault or Safety Trip Reset Required
Inf_Label	BOOL	Purge Control Label (@Description used by HMI)
Inf_Tag	BOOL	Tag ID (@Description used by HMI)
Inf_BuildID	DINT	raS_Opr_BMS_Purge AOI Build ID

Programming Example

General arrangement for this example is shown in the Figure 1 - Example Task, Program, and Routine Arrangement. The raS_Opr_BMS_Purge, being a Safety AOI, is located in the BMS_Example program, b_Purge routine. The rungs that follow are taken from that routine:

- Rung 1: Purge Enable logic, tied to same input as Master AOI.
- Rung 2: raS_Opr_BMS_Purge AOI.

Purge Logic



Graphic Symbols

Graphic Symbols are used as launch buttons within HMI applications to open up faceplate displays. See Basic Launch Button Attributes section for details on configuration and indicators.

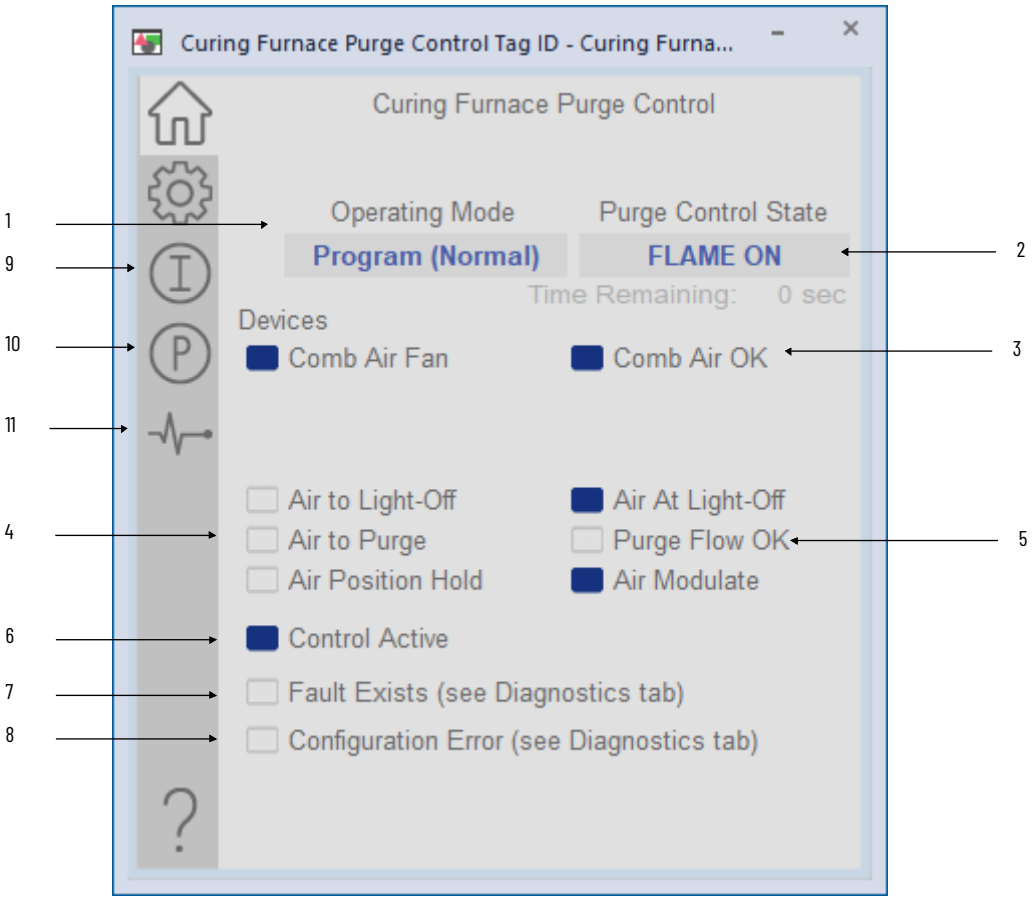
FactoryTalk® View SE Graphic Symbols

Graphic Symbol Name	Graphic Symbol	Description	Global Object Parameter Values
GO_PurgeStatus		Purge Status with faceplate navigation. This launch button graphic object allows the user to navigate to the object faceplate. The text on the button face is set to the tag's .Inf_Label.@Description field. Also shown are the operating state and purge time remaining in current operation/state.	#102: AOI Instance (e.g. {::[PAC]Program::SafetyProgram._Instance Name}) #103: HMI Control Tag #120: Faceplate X Position, pixels from left edge of display (e.g. /X100) #121: Faceplate Y Position, pixels from top edge of display (e.g. Y100)
GO_PurgeStatusNoNav		Purge Status without navigation. . The text on the object face is set to the tag's .Inf_Label.@Description field. Also shown are the operating state and purge time remaining in current operation/state	#102: AOI Instance (e.g. {::[PAC]Program::SafetyProgram._Instance Name})

Faceplates

There are basic faceplate attributes that are common across all instructions. See Basic Faceplate Attributes on page 21.

Home

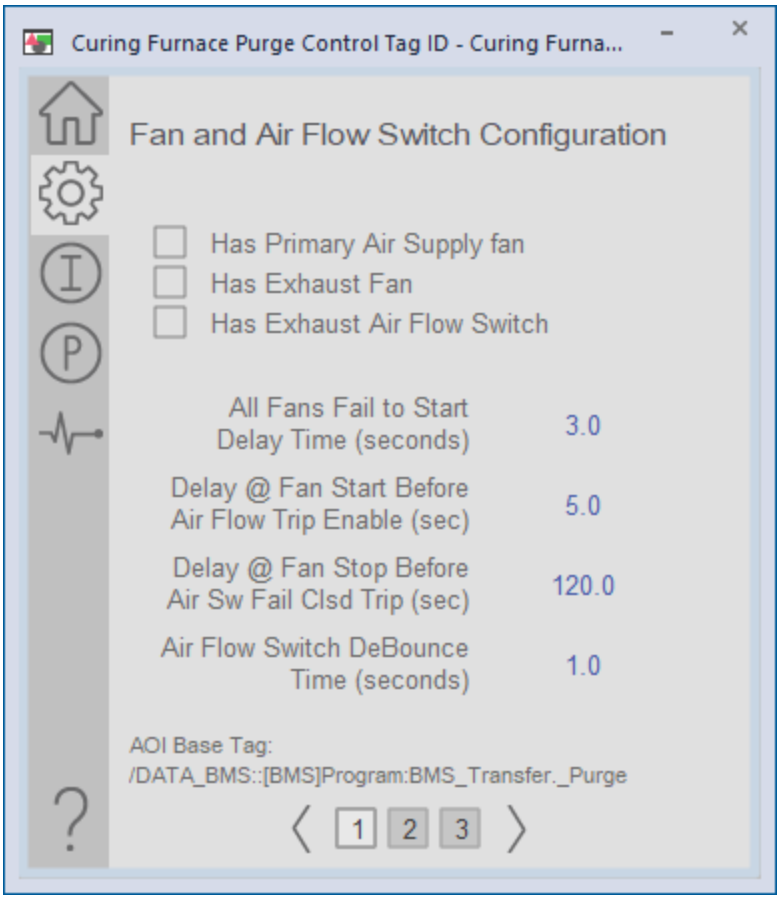


Item	Description
1	Purge Operating Mode indicator.
2	Purge Operating State indicator.
3	Air Fan(s) status and Air Flow status indicators
4	Air Actuator Request indicator
5	Status indicators
6	Purge sequence active status indicator
7	Purge fault status summary indicator
8	Purge configuration status summary indicator
9	Purge Interlock status summary indicator, showing all Interlock conditions OK.
10	Purge Permissive status summary indicator, showing all Permissive conditions OK.
11	Purge Master Diagnostic status summary indicator, showing all Diagnostic conditions OK.

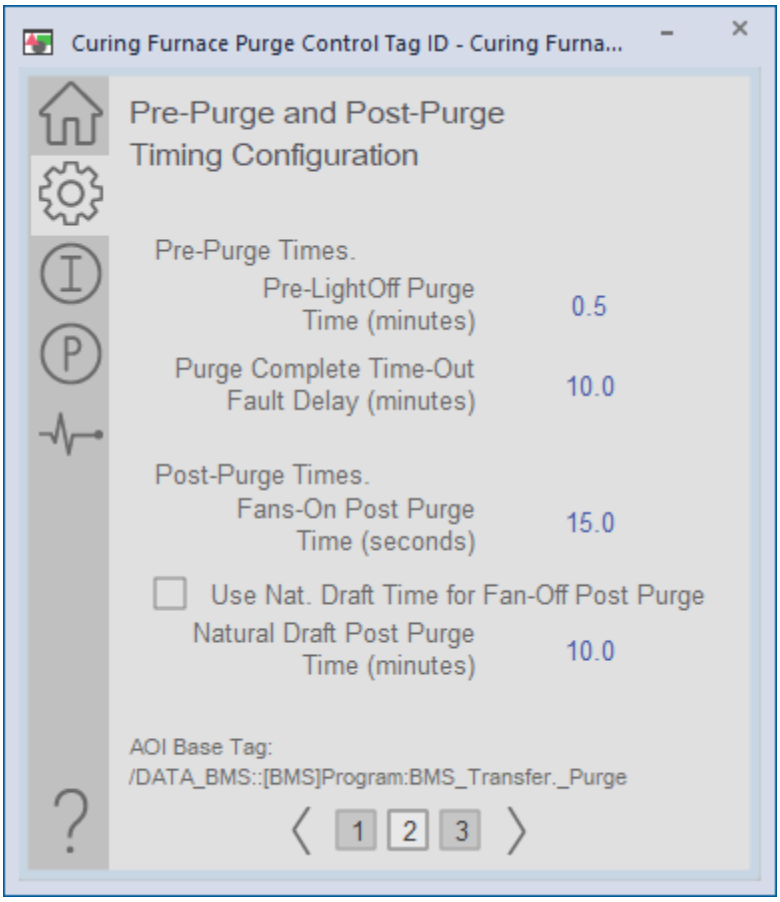
Configuration

There are three pages within the configuration tab.

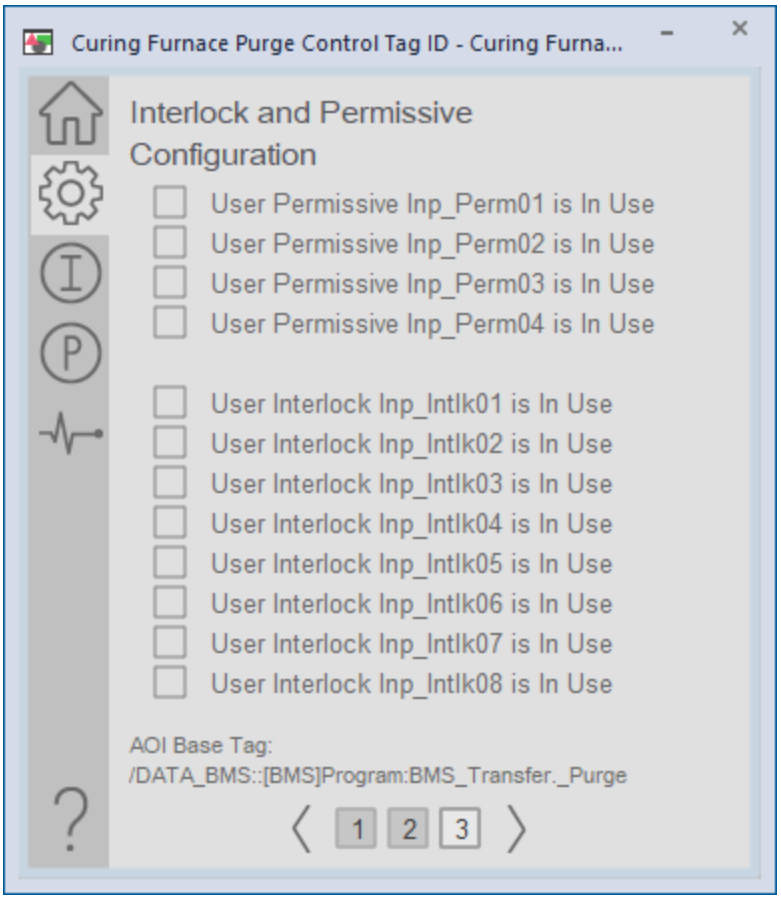
Configuration Page 1 - Fan and Air Flow Switch Configuration



Configuration Page 2 - Pre-purge and Post-purge Timing Configuration



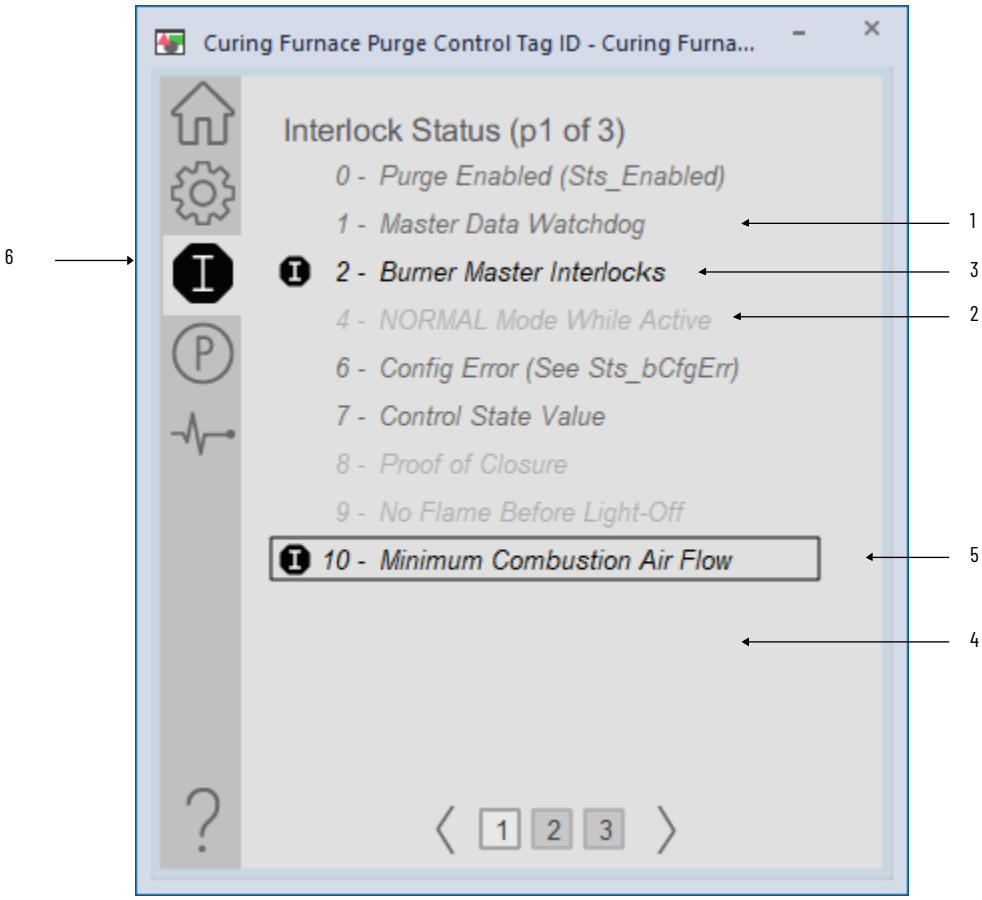
Configuration Page 3 - Interlock and Permissive Configuration



Interlock Status

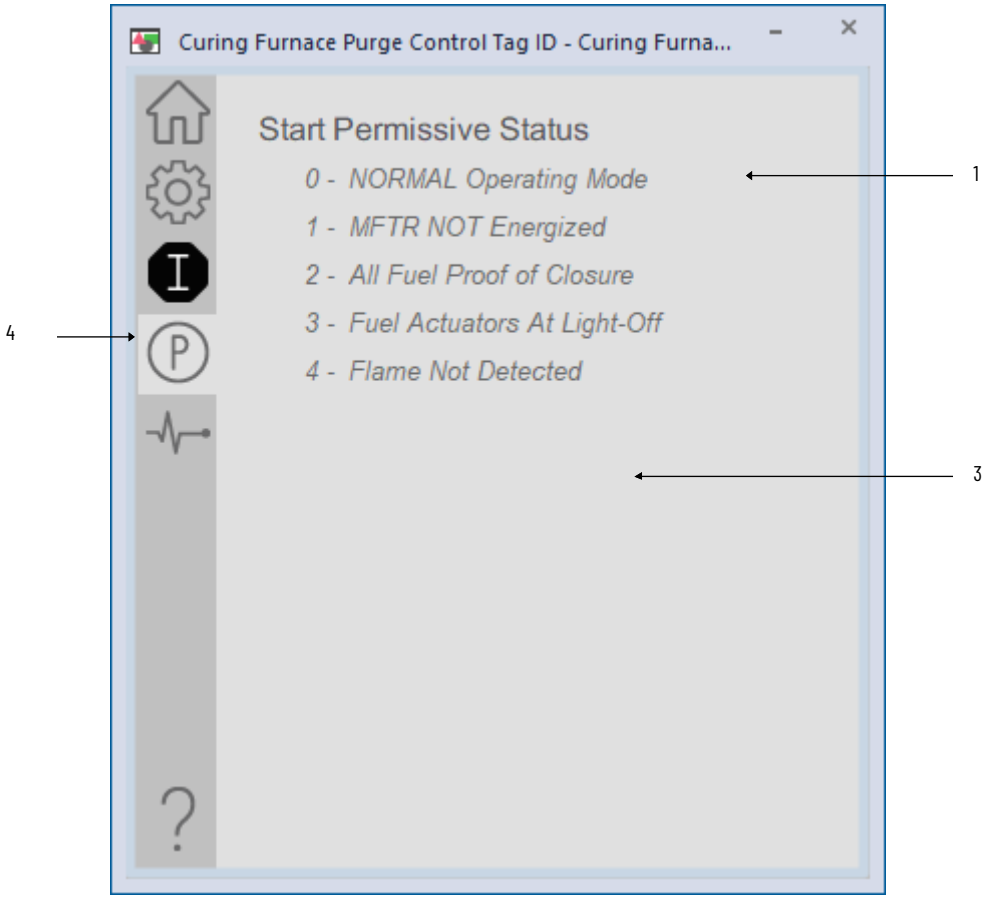
There are 3 pages within the Interlock Status tab - each with 12 interlocks.

Interlock Status Page 1 - Interlocks 0-11



Item	Description
1	Interlock currently monitored, but OK to operate.
2	Interlock configured for use, but currently not monitored.
3	Interlock currently monitored, but NOT OK to operate.
4	Interlock not configured for use (Process Safety Limit B in this case).
5	Interlock currently monitored, NOT OK, and has generated a Safety Trip.
6	Summary indicator – at least one Interlock is NOT OK.

Permissive Status

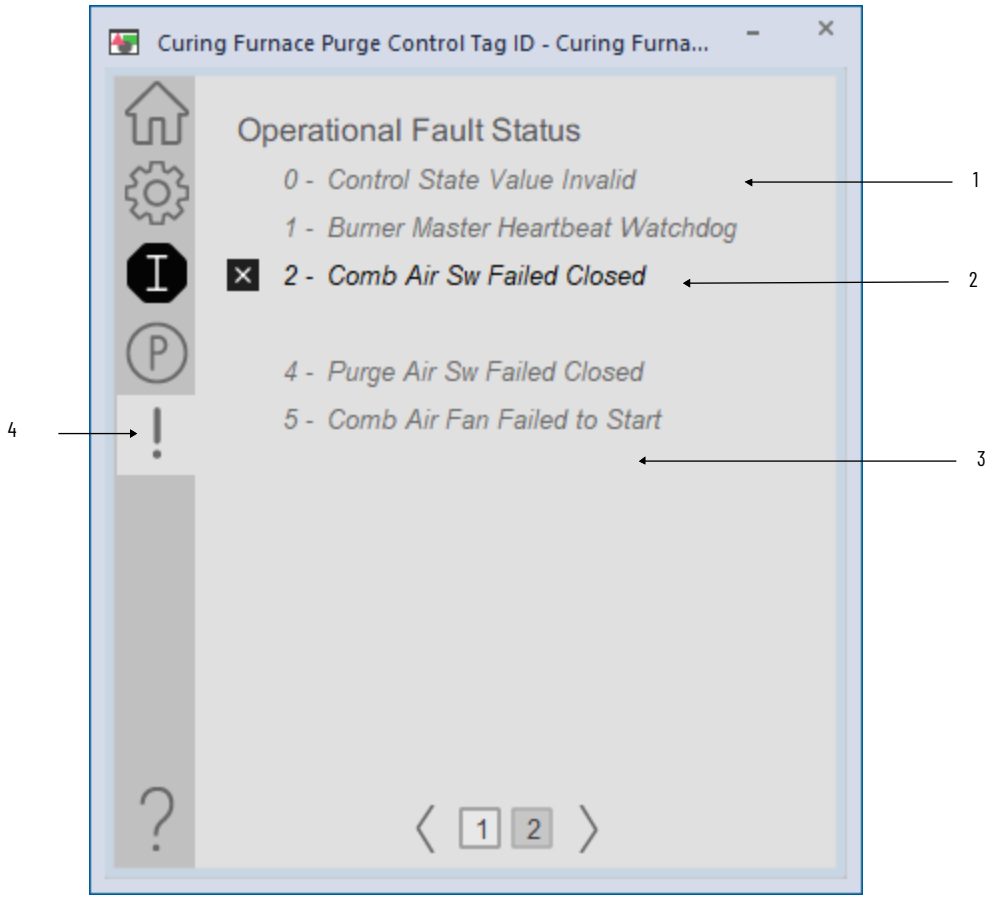


Item	Description
1	Permissive currently monitored, but OK to Start Operation.
3	Permissive not configured for use (User Permissive Input 1 in this case).
4	Summary indicator – at least one Permissive condition is NOT OK.

Diagnostics

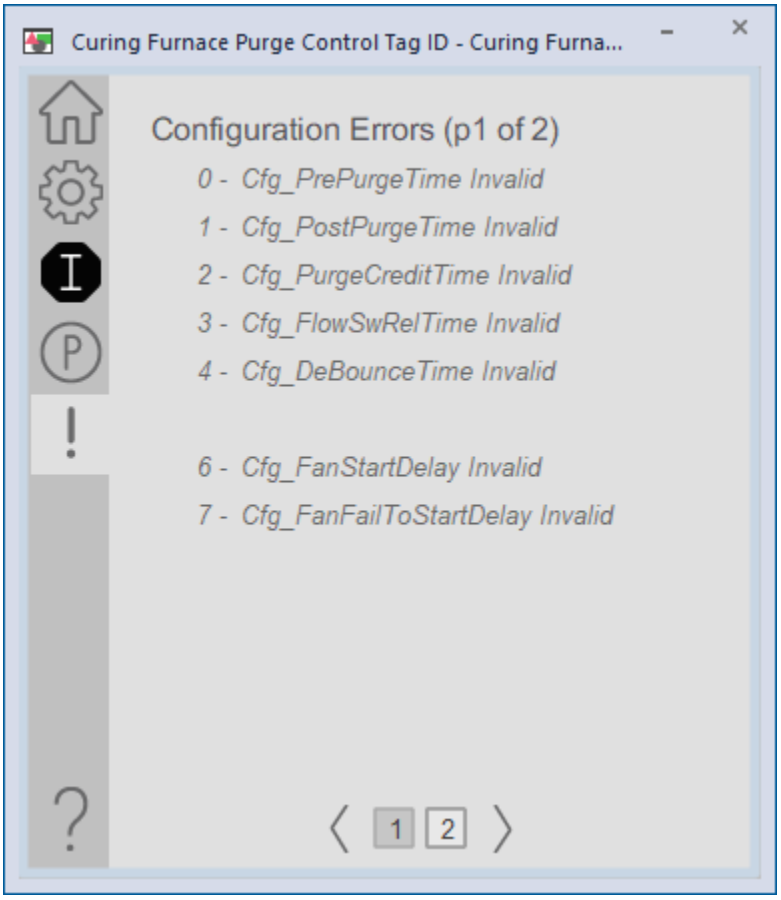
There are 2 pages within the diagnostics tab.

Diagnostics Page 1 - Operational Fault Status



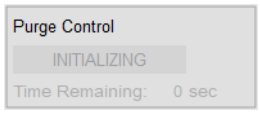
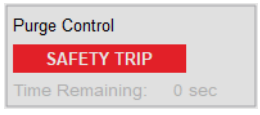
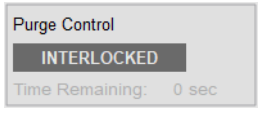
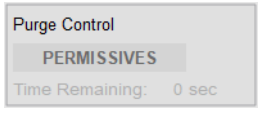
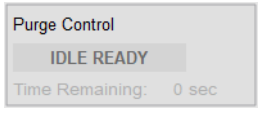
Item	Description
1	Operational Fault currently monitored, and OK to Operate.
2	Operational Fault currently monitored, and NOT OK to Operate.
3	Operational Fault not configured for use (Pilot B Heartbeat Watchdog in this case).
4	Summary indicator – at least one Diagnostic condition is NOT OK.

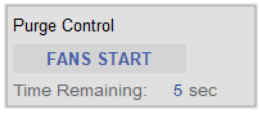
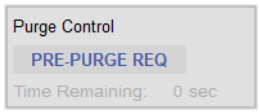
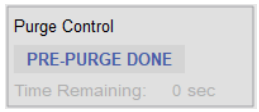
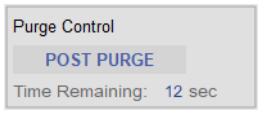
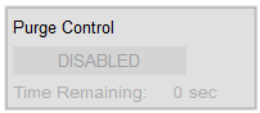
Diagnostics Page 2 - Configuration Errors



State Enumerations

The following table identifies the states available to the Purge when enabled (Sts_Enabled is True). If Sts_Enabled is False, the Purge is allowed to only be in States 0, 1, and 2, and will indicate DISABLED.

Val_eState	State Name	Description	HMI Graphic
0	Initializing	AOI performing 1st scan initialization.	
1	Safety Trip	Purge Safety Trip present.	
2	Interlocked	Purge Interlock not met.	
3	Not Ready	Purge waiting for all Permissives OK.	
4	Idle / Ready	Purge Idle and Ready for Start Command.	

5	Fan(s) Start	Request Fan(s) Start and confirm air flow.	
6	Pre-Purge Request	Request Purge Air Flow.	
7	Purge Confirmed	Purge Air Flow Confirmed, timed pre-purge active.	
8	Purge Complete	Timed Pre-Purge complete, returning air to light-off conditions.	
9	Purge Hold	Air flow at light-off, awaiting fuel light-off confirmation.	
10	Flame Present	Flame presence confirmed.	
11	Post Purge	Post-firing timed purge active.	
0, 1, or 2		Purge Disabled (Sts_Enabled = 0)	

BMS Pilot (raS_Opr_BMS_Pilot)

Overview

The Pilot component AOI manages all interlock signals and devices associated with one pilot fuel train and ignition. A burner system requires at least one Pilot component and supports as many as 4 independent pilot fuel trains. The primary function of the Pilot component is safe activation and light-off of a pilot flame used for ignition of a main fuel. Alternatively, some burners have spark ignited main fuel(s). The Pilot component is employed for these burners as well, supporting up to 4 spark ignited main fuels. A single burner system will not be allowed to employ the Pilot component(s) as both spark-ignited fuels and pilots.

Limitations

The following limitations must be considered when implementing this instruction within a BMS Library application.

AOI Class	Safety
Minimum Instances Required	1
Maximum Instances Allowed	4

System Interaction

Each Pilot component exchanges data with the BMS Master using a pair of UDT tags, one for data in each direction. The content of these UDTs is detailed in the following sections.

Pilot to Master Data

Data Type: raS_UDT_BMS_Pilot_ToMaster (12 bytes) array of 4 elements. Each Pilot component uses one element of this array, and the element used identifies to the Master which Pilot Instance (i.e. Pilot A, B, C, or D) the data is coming from. It is imperative that each Pilot component uses the same array index value for data to and from the Master.

Pilot Parameter Reference: Ref_ToMaster (InOut)

Data from each Pilot to Master includes the common component data structure, pilot operational status, pertinent configuration data, and command acknowledge bits. A list of UDT elements is shown in the following table.

Pilot to Master Data Elements

Element Name	Data Type	Description
Ref_ToMaster	raS_UDT_BMS_Purge_ToMaster	Pilot Component Status to Master
Ref_ToMaster.Sts	raS_UDT_BMS_CommonStatus	Pilot Component Common Status to Master
Ref_ToMaster.Sts.Enabled	BOOL	Pilot Component Enabled
Ref_ToMaster.Sts.SafetyTrip	BOOL	Pilot Component Generated Safety Trip
Ref_ToMaster.Sts.IntlkOK	BOOL	Pilot Component Interlocks OK
Ref_ToMaster.Sts.Ready	BOOL	Pilot Component Ready for Start
Ref_ToMaster.Sts.Active	BOOL	Pilot Component Operating Sequence is Active
Ref_ToMaster.Sts.Normal	BOOL	Pilot Component in NORMAL MODE
Ref_ToMaster.Sts.TestMode	BOOL	Pilot Component in TEST MODE
Ref_ToMaster.Sts.TestActive	BOOL	Pilot AOI Component in TEST MODE with Device Test Active
Ref_ToMaster.Sts.eState	SINT	Pilot Component Enumerated State Value
Ref_ToMaster.Sts.HBEcho	INT	Pilot Component Communication Heartbeat Echo
Ref_ToMaster.Sts.BuildID (hidden parameter)	DINT	raS_Opr_BMS_Purge AOI Build ID
Ref_ToMaster.Sts_FuelPoC	BOOL	All Fuel Blocking Valves Closed
Ref_ToMaster.Sts_AtLightOff	BOOL	Fuel Actuator at Light Off Position
Ref_ToMaster.Sts_FuelEnabled	BOOL	Blocking Valves Energized
Ref_ToMaster.Sts_FlamePresent	BOOL	Flame Sensed by Pilot Flame Sensor
Ref_ToMaster.Sts_Release	BOOL	Pilot Fuel Lit and Flame Stable
Ref_ToMaster.Sts_VPActive	BOOL	Valve Proving Sequence Active
Ref_ToMaster.Sts_VPReqd	BOOL	Valve Proving Sequence Required Prior to Next Burner Light-Off
Ref_ToMaster.Sts_Scavenging	BOOL	Fuel Scavenging Active
Ref_ToMaster.Cfg_Exists	BOOL	Pilot Exists
Ref_ToMaster.Cfg_HasVP	BOOL	Pilot Has and Requires Valve Proving
Ref_ToMaster.Cfg_HasScavenge	BOOL	Pilot Has and Requires Post-Firing Scavenging
Ref_ToMaster.Cfg_IsMainFuel	BOOL	Pilot Configured as Spark-Ignited Main Fuel
Ref_ToMaster.Ack_Start	BOOL	Acknowledge Start Request
Ref_ToMaster.Ack_Stop	BOOL	Acknowledge Stop Request
Ref_ToMaster.Ack_Reset	BOOL	Acknowledge Reset Request
Ref_ToMaster.Ack_VPStart	BOOL	Acknowledge Valve Proving Sequence Start Request
Ref_ToMaster.Ack_Scavenge	BOOL	Acknowledge Manual Fuel Scavenge Request

Master to Pilot Data

Data Type: raS_UDT_Master_ToPilot (8 bytes) array of 4 elements. Each Pilot component uses one element of this array, and the element used identifies to the Master which Pilot Instance (i.e. Pilot A, B, C, or D) is receiving the data. It is imperative that each Pilot component uses the same array index value for data to and from the Master.

Pilot Parameter Reference: Ref_FromMaster (InOut)

Data from Master to Purge includes the SysSts structure, start/stop/reset commands, and information gathered from pilot and fuel AOIs necessary for purge operation. A list of UDT elements is shown in the following table.

Master to Pilot Data Elements

Element Name	Data Type	Description
Ref_FromMaster	raS_UDT_BMS_Master_ToPilot	Data from Master to Pilot component.
Ref_FromMaster.SysSts	raS_UDT_BMS_Master_SysSts	System-wide common status information.
Ref_FromMaster.SysSts.Enable	BOOL	Burner System Enable
Ref_FromMaster.SysSts.IntlksOK	BOOL	Burner System Interlocks OK
Ref_FromMaster.SysSts.MFTR	BOOL	Master Fuel Trip Relay Energized
Ref_FromMaster.SysSts.Normal	BOOL	Burner System in NORMAL Operating Mode
Ref_FromMaster.SysSts.TestMode	BOOL	Burner System in TEST Mode
Ref_FromMaster.SysSts.TestActive	BOOL	Burner in TEST Mode with a Device Test Active
Ref_FromMaster.SysSts.NoFlame	BOOL	Pre-Purge NOT Completed – No Flame Allowed
Ref_FromMaster.SysSts.eState	SINT	Burner System Master State Value
Ref_FromMaster.SysSts.HB	INT	Master Communication Heartbeat
Ref_FromMaster.Cmd_Start	BOOL	Pilot Sequence Start Request
Ref_FromMaster.Cmd_Stop	BOOL	Pilot Sequence Stop Request
Ref_FromMaster.Cmd_Reset	BOOL	Pilot Reset Request
Ref_FromMaster.Cmd_VPStart	BOOL	Manual Fuel Valve Proving Sequence Start Request
Ref_FromMaster.Cmd_Scavenge	BOOL	Manual Fuel Scavenge Request
Ref_FromMaster.Inp_FlameOK	BOOL	Flame Present is OK for Light-Off
Ref_FromMaster.Inp_ActuatorHold	BOOL	Fuel Actuator Hold Current Position
Ref_FromMaster.Inp_ActuatorRel	BOOL	Fuel Actuator Released for Modulation
Ref_FromMaster.Inp_HeaderNotRdy	BOOL	Fuel Header is Not Ready to supply main fuel to burner.
Ref_FromMaster.Sts_IsMainFuel	BOOL	Pilot Component is Spark-Ignited Main Fuel.

Pilot AOI Configuration

The Pilot AOI configuration includes fuel type, required interlock devices, fuel train valve configuration, number of spark ignitor devices, and number/type of flame sensors. Parameters for Valve proving and fuel scavenging configuration are also provided. The Pilot Component provides up to 2 user-assignable interlock inputs and up to 4 user-assignable permissive inputs that are enabled by configuration. Configuration parameter details are shown below.

The Master determines whether Pilot components are Main Fuels or Pilots based on setting of Cfg_FuelQty (Master configuration parameter). If the Master's Cfg_FuelQty is non-zero, the burner has separate Fuel components and Pilot components are used as pilots for those fuels. If the Master's Cfg_FuelQty is zero, the burner does not have any separate Fuel components and Pilot components are used as Spark-Ignited Main Fuels for the burner. This information is conveyed from the Master to the Pilot components in the Ref_FromMaster.Sts_IsMainFuel data element.

Interlocks

The raS_Opr_BMS_Pilot AOI monitors 29 interlock conditions, 2 of which are user selectable. The following table identifies each interlock condition, when each is monitored, and when each will trigger a Safety Trip.

Pilot AOI Interlock List

Interlock Condition	Interlock Enable	Safety Trip Enable
Pilot Component Disabled	Always	Pilot Component Active
Master Comm Heartbeat Watchdog Timeout	Always	Pilot Component Active
System Interlock Exists	Always	Never
Normal Mode Lost (Sts_Normal = 0)	Pilot Component Active	Same as Interlock Enable
MFTR Lost	Pilot Component Active	Same as Interlock Enable
Configuration Error Exists (Sts_bCfgErr <> 0)	Pilot Component Enabled	Pilot Component Active
Invalid State Value	Pilot Component Enabled	Pilot Component Active
Flame Sensor Fault	Pilot Component Enabled	Same as Interlock Enable
Fuel Actuator Not at Light-Off (Inp_AtLightOffPosn = 0)	Light-Off Pos'n Required and Is Main Fuel	Same as Interlock Enable
Light-Off Flame Failure Fault	Pilot Component Enabled	Same as Interlock Enable
Flame-Out Fault	Pilot Component Enabled	Same as Interlock Enable
Flame Sensor Failed On Fault	Pilot Component Enabled	Same as Interlock Enable
Fuel Low Pressure Trip (Inp_PSL = 0)	Pilot Component Enabled	Same as Interlock Enable
Fuel High Pressure Trip (Inp_PSH = 0)	Pilot Component Enabled	Same as Interlock Enable
Fuel Low Temperature Trip (Inp_TSL = 0)	Pilot Component Enabled and Has Low Temperature Switch	Same as Interlock Enable
Fuel High Temperature Trip (Inp_TSH = 0)	Pilot Component Enabled and Has High Temperature Switch	Same as Interlock Enable
Atomizing Pressure Loss (Inp_AtomizerPS = 0)	Pilot Component Enabled and Has Atomizing	Same as Interlock Enable
Atomizing Flow Loss (Inp_AtomizerFS = 0)	Pilot Component Enabled and Has Atomizing	Same as Interlock Enable
Valve Proving Sequence Failure	Pilot Component Enabled and Has Valve Proving	Same as Interlock Enable
Blocking Valve A Failure	Pilot Component Enabled	Pilot Component Active or Valve Proving
Blocking Valve B Failure	Pilot Component Enabled and Has Blocking Valve B	Pilot Component Active or Valve Proving
Fuel Auxiliary Valve Failure	Pilot Component Enabled and Has Fuel Auxililary Valve	Pilot Component Active or Valve Proving
Atomizing Valve Failure	Pilot Component Enabled and Has Atomizing	Pilot Component Active
Scavenging Valve Failure	Pilot Component Enabled and Has Fuel Scavenging	Pilot Component Active
Spark Ignitor A Failure	Pilot Component Enabled	Pilot Component Active
Spark Ignitor B Failure	Pilot Component Enabled and Has Ignitor B	Pilot Component Active
Spark Ignitor C Failure	Pilot Component Enabled and Has Ignitor C	Pilot Component Active
User Interlock 01 Not OK (Inp_Intlk01 = 0)	Pilot Component Enabled and Has User Interlock 01	Same as Interlock Enable
User Interlock 02 Not OK (Inp_Intlk02 = 0)	Pilot Component Enabled and Has User Interlock 02	Same as Interlock Enable

Permissives

The raS_Opr_BMS_Pilot AOI monitors 8 permissive conditions, 4 of which are user selectable. The following table identifies each permissive condition and when each is monitored. Once the Pilot AOI transitions to an Active state, permissive conditions no longer impact operation.

Pilot AOI Permissive List

Permissive Lost Condition	Permissive Enable
Not NORMAL Operating Mode (Sts_Normal = 0)	Pilot Component Not Active
MFTR NOT Energized	Pilot Component Not Active
Fuel Actuator Not at Light-Off	Pilot Component Not Active and Is Main Fuel
Fuel Header Not Ready	Pilot Component Not Active and Is Main Fuel
User Permissive 01 Not OK (Inp_Perm01 = 0)	Pilot Component Not Active and Has User Permissive 01
User Permissive 02 Not OK (Inp_Perm02 = 0)	Pilot Component Not Active and Has User Permissive 02
User Permissive 03 Not OK (Inp_Perm03 = 0)	Pilot Component Not Active and Has User Permissive 03
User Permissive 04 Not OK (Inp_Perm04 = 0)	Pilot Component Not Active and Has User Permissive 04

Operations

Operation Modes

The BMS Library objects provide only physical operation mode. There is no virtual/simulated mode offered.

Normal Operation Mode

The Master must be in NORMAL Mode to operate the burner, this mode status is transmitted to the Pilot component. While in NORMAL Mode, the Pilot component receives requests from the Master, making appropriate operational changes based on the command received. There is no direct interaction with the Pilot component except through the Master's Data Interface.

NORMAL Mode status is echoed back to the Master indicating whether the Pilot Component is in NORMAL Mode or not.

Test Enable Mode Operation

The Pilot component is in TEST ENABLE Mode any time the Master is in TEST ENABLE Mode. Any commands to start or otherwise activate the Pilot component are ignored.

TEST ENABLE Mode status is echoed back to the Master indicating whether the Pilot component is in TEST ENABLE Mode or not.

Device Test Active

Device Test Active indicates a device under control of the Pilot AOI is actively being tested. Any time a device under control of this Pilot component is actively being tested, testing of all other devices is immediately disabled and the Pilot component indicates Device Test Active. This status is sent to the Master.

Operation

The Pilot AOI is intended to support gaseous fuels (e.g. natural gas or LP gas) or light fuel oil. It is not intended to support heavy fuel oil.

The Pilot AOI controls up to 8 physical devices: 1 to 3 spark ignitors and 1 to 5 valves. At least one spark ignitor and one valve is required.

The Pilot AOI supports 1 to 3 flame sensors and must use at least one.

Gas pilot fuel trains typically use 1, 2, or 3 valves: 1 or 2 fuel blocking valves, and an optional vent between 2 blocking valves. The Pilot AOI dedicates device references to the three valve functions. A Valve Proving sequence is also available for gas fuels to confirm the blocking valves seal when closed.

Light oil gas trains may also have 1 or 2 blocking valves, but instead of a vent, the third valve is used for recirculation. Selecting light oil as the pilot fuel alters the valve sequencing to accommodate. Oil may require fuel atomization so one device is dedicated to that function and included in the control sequencing as appropriate. The light oil fuel train may require scavenging so one device is dedicated to that function and included in the control sequencing as appropriate.

Typical Pilot Fuel Control Sequence (same sequence for Pilot Duty and Main Fuel Duty):

1. Continually monitor all interlocks directly associated with the pilot fuel and ignition function.
2. A start request is received from the Master, kicking off the pilot sequence.
3. If the fuel requires atomizing, the atomizing media valve is energized and flow confirmed.
   a. Atomizing media flow is confirmed by flow or pressure switch
   b. If atomizing is not required, this step is skipped.
4. Spark ignitor is energized.
5. Pilot fuel blocking valves are energized to admit pilot fuel to burner.
   a. If vent is used, the vent valve is closed before blocking valves open.
   b. If recirc is used, the blocking valves are energized first, then recirc is closed.
   c. The order of valve sequencing is determined by Cfg_FuelAuxVlvFirst
      -Set = Energize Fuel Auxiliary Valve (vent) before energizing blocking valves.
      -Clear = Energize blocking valves first, the energize Fuel Auxiliary Valve (recirc).
6. After a configurable delay not to exceed what is allowed by code, the spark ignitor is turned off and the flame sensor is checked for flame presence.
   a. Failure to sense flame by the end of this step triggers a Light-Off Failure safety trip.
7. Another configurable delay can be used to ensure flame stability before indicating to the Master that the pilot is ready for main fuel light-off.
8. Once flame presence is confirmed and Light-Off is complete, any future flame loss triggers a Flame-Out safety trip unless blocking valves are closed first.
9. Pilot ready for main fuel ignition signal is sent to the Master.
   a. Pilot stays lit and active until a stop request is made.
10. Stop request from Master
    a. Blocking valves are closed, vent or recirc is opened.
    b. Pilot ready signal to Master is removed.
11. If scavenging is required, the spark ignitor is energized and the scavenge valve is opened.
    a. Timed activity
    b. Once the scavenging timer is done, turn off spark and close scavenging valve.
    c. Note that fuel scavenging is intended to safely burn any fuel remaining in burner feed piping, but a continuous and steady flame cannot be guaranteed. Thus, flame confirmation is not reliable for

this activity, and flame loss will not generate any type of trip during fuel scavenging.

d. If scavenging not required, this step is skipped

12. Pilot AOI returns to idle.

Blocking Valve Proving Sequence

For gaseous fuels, a Valve Proving sequence can be selected to run either before pilot light-off or after pilot shutdown. This sequence is not intended to supersede annual manual valve leak testing as required by code but does provide a tacet confirmation that the blocking valves have not failed and are closing on request. A Valve Proving test failure generates a safety trip, removing that fuel from service until the trip is reset.

The Valve Proving test relies on two digital inputs tied to pressure switches monitoring the cavity between the fuel blocking valves. One switch monitors a high pressure threshold, and the other monitors a low pressure threshold. Both thresholds are tested in sequence to confirm both blocking valves can seal and do not have significant leakage failures.

Typical Valve Proving Sequence

1. Confirm both blocking valves are deenergized to start.
2. Energize the auxiliary vent valve to close it.
   a. Skip this step if the valve train does not include a vent.
3. Open the downstream blocking valve to release any latent pressure between the blocking valves.
   a. Both pressure switches must indicate deenergized inputs (pressure is below the threshold for both switches).
   b. Failure of either or both pressure switches to open constitutes a sequence failure. Generate a safety trip and abort this sequence.
4. After a few seconds, close the downstream blocking valve and monitor the low pressure switch.
   a. If at any time during this step the low pressure switch is activated, the upstream blocking valve has failed and is leaking fuel into the test chamber. Generate a safety trip and abort this sequence.
5. After the low pressure test time, energize the upstream blocking valve to pressurize the test chamber.
   a. Both pressure switches must indicate energized inputs (pressure is above the threshold for both switches).
   b. Failure of either or both pressure switches to close constitutes a sequence failure. This may indicate a severely leaking vent valve. Generate a safety trip and abort this sequence.
6. After a few seconds, close the upstream blocking valve and monitor the high pressure switch.
   a. If at any time during this step the high pressure switch is deactivated, the downstream blocking valve may have failed and is leaking fuel out of the test chamber. Generate a safety trip and abort this sequence.
   b. Note that a leaking vent valve may also cause this test to fail. Additionally, a severely leaking vent valve may mask a leakage failure of the upstream blocking valve.
7. After the high pressure test time, the test is complete.

a. Deenergize the auxiliary vent valve if it exists.

Required Files

Objects include Add-On Instructions (AOIs) and HMI faceplates. The revision number (e.g. 1.01) used in filenames can change as new revisions are created.

Controller Files

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

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

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

Device/Item	Add-On Instruction	ROUTINE Import
BMS Interface	raS_Opr_BMS_Pilot_1.00_AOI.L5X	raS_Opr_BMS_Pilot_1.00_SftyROUTINE.L5X

FactoryTalk® View HMI Files

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

Device/Item	Type	FactoryTalk® View SE Faceplate
BMSMaster Faceplate	Display	(raS-1_00-SE) raS_Opr_BMS_Pilot-Faceplate.gfx
Graphic Symbols	Global Object	(raS-1-SE) Graphic Symbols - BMS.ggfx
Toolbox	Global Object	(raS-1-SE) Toolbox - BMS.ggfx

Add-On Instruction I/O Data

The following tables identify raS_Opr_BMS_Pilot AOI parameters.

InOut Parameters

raS_Opr_BMS_Pilot AOI InOut Reference Parameters

Parameter Name	Data Type	Function
Ref_FromMaster	raS_UDT_BMS_Master_ToPilot	Command Data to Pilot Control
Ref_ToMaster	raS_UDT_BMS_Pilot_ToMaster	Status Data From Pilot Control
Ref_IgnitorA	raS_Dvc_BMS_D2S	Spark Ignitor A Device Tag
Ref_IgnitorB	raS_Dvc_BMS_D2S	Spark Ignitor B Device Tag
Ref_IgnitorC	raS_Dvc_BMS_D2S	Spark Ignitor C Device Tag
Ref_FuelBlkVlvA	raS_Dvc_BMS_D2S	Fuel Train Downstream or Single Blocking Valve Device Tag
Ref_FuelBlkVlvB	raS_Dvc_BMS_D2S	Fuel Train Upstream Blocking Valve Device Tag
Ref_FuelAuxVlv	raS_Dvc_BMS_D2S	Fuel Train Auxiliary Valve Device Tag
Ref_Atomizer	raS_Dvc_BMS_D2S	Atomizing Media Valve Device Tag
Ref_ScavengeVlv	raS_Dvc_BMS_D2S	Fuel Scavenging Valve Device Tag

Configuration Parameters

raS_Opr_BMS_Pilot AOI Configuration Parameters

Parameter Name	Data Type	Function
Cfg_FuelType	BOOL	0=Gas Fuel 1=Light Oil
Cfg_FullStrokeTime	REAL	Main Fuel Actuator Full Stroke Time at Max Rate (seconds)
Cfg_HasIgnitorB	BOOL	Ignitor B Exists and is Required
Cfg_HasIgnitorC	BOOL	Ignitor C Exists and is Required
Cfg_HasFlameB	BOOL	Flame Sensor B Exists and is Required
Cfg_HasFlameC	BOOL	Flame Sensor C Exists and is Required
Cfg_FlameAQNC	BOOL	Inp_FlameAQ Is NC Contact (1=NoFlame)
Cfg_FlameBQNC	BOOL	Inp_FlameBQ is NC Contact (1=NoFldame)
Cfg_FlameCQNC	BOOL	Inp_FlameCQ is NC Contact (1=NoFlame)
Cfg_FlameIndep	BOOL	Pilot Flame Sensor(s) Independent from Main Flame Sensor
Cfg_FlameOffDelay	REAL	Flame Sensor Off Debounce Delay (seconds)
Cfg_FlameOnDelay	REAL	Flame Sensor On Debounce Delay (seconds)
Cfg_IgnitionTime	REAL	Ignitor Active Time (seconds)
Cfg_TFITime	REAL	Proof of Ignition Delay Time (seconds)
Cfg_StabilizeTime	REAL	Flame Stabilization Time (seconds)
Cfg_HasTSL	BOOL	Heated Fuel Oil - Low Temperature Switch Exists and is Required
Cfg_HasTSH	BOOL	Heated Fuel Oil - High Temperature Switch Exists and is Required
Cfg_SingleFuelBlkVlv	BOOL	Single Fuel Blocking Valve. (Ref_FuelBlkVlvA Only)
Cfg_HasProofOfClosure	BOOL	Pilot Blocking Valve(s) Have Closed Limit Switches
Cfg_HasFuelAuxVlv	BOOL	Fuel Train Auxiliary (Vent or Recirc) Valve Exists and Is In Use.
Cfg_FuelAuxVlvFirst	BOOL	Energized Fuel Auxiliary Valve Before Blocking Valve(s)
Cfg_HasAtomizing	BOOL	Pre-LightOff is Atomizing Media
Cfg_AtomizerDelay	REAL	Delay Before Activating Interlock on Pre-LightOff Feedback (seconds)
Cfg_HasFuelScavenge	BOOL	Fuel Requires Post-Firing Scavenging
Cfg_FuelScavengeTime	REAL	Post-Firing Active Time (seconds)
Cfg_HasVP	BOOL	Gas Fuel Train with Valve Proving
Cfg_VPPostFiring	BOOL	Perform Valve Proving Post-Firing
Cfg_VPPrepTime	REAL	Valve Proving Evacuate and Pressurize Time (seconds)
Cfg_VPLoTestTime	REAL	Valve Proving Low Pressure Test Time (seconds)
Cfg_VPHiTestTime	REAL	Valve Proving High Pressure Test Time (seconds)
Cfg_HasPerm01	BOOL	User Permissive Input 1 In Use
Cfg_HasPerm02	BOOL	User Permissive Input 2 In Use
Cfg_HasPerm03	BOOL	User Permissive Input 3 In Use

raS_Opr_BMS_Pilot AOI Configuration Parameters

Parameter Name	Data Type	Function
Cfg_HasPerm04	BOOL	User Permissive Input 4 In Use
Cfg_HasIntlk01	BOOL	User Interlock Input 1 In Use
Cfg_HasIntlk02	BOOL	User Interlock Input 2 In Use

Input Parameters

raS_Opr_BMS_Pilot AOI Input Parameters

Parameter Name	Data Type	Function
Inp_Enable	BOOL	Enable Pilot Control AOI
Inp_FlameA	BOOL	Flame Sensor A Input Normally Open (1=Flame present)
Inp_FlameAQ	BOOL	Flame Sensor A OK or Normally Closed
Inp_FlameStrengthA	REAL	Flame Sensor A Strength Value
Inp_FlameB	BOOL	Flame Sensor B Input Normally Open (1=Flame present)
Inp_FlameBQ	BOOL	Flame Sensor B OK or Normally Closed
Inp_FlameStrengthB	REAL	Flame Sensor B Strength Value
Inp_FlameC	BOOL	Flame Sensor C Input Normally Open (1=Flame present)
Inp_FlameCQ	BOOL	Flame Sensor C OK or Normally Closed
Inp_FlameStrengthC	REAL	Flame Sensor C Strength Value
Inp_PSL	BOOL	Fuel Low Pressure Switch
Inp_PSH	BOOL	Fuel High Pressure Switch
Inp_TSL	BOOL	Heated Fuel Oil Low Temperature Switch
Inp_TSH	BOOL	Heated Fuel Oil High Temperature Switch
Inp_VPSL	BOOL	Valve Proving Low Pressure Threshold Switch
Inp_VPSH	BOOL	Valve Proving High Pressure Threshold Switch
Inp_AtLightOffPosn	BOOL	Fuel Actuator at Light-Off Position
Inp_AtomizerPS	BOOL	Atomizing Media Supply Pressure Switch
Inp_AtomizerFS	BOOL	Atomizing Media Flow Confirmation
Inp_BlkVlvAPoC	BOOL	Fuel Blocking Valve A Proof-of-Closure Switch
Inp_BlkVlvBPoC	BOOL	Fuel Blocking Valve B Proof-of-Closure Switch
Inp_Perm01	BOOL	User Permissive Input 1
Inp_Perm02	BOOL	User Permissive Input 2
Inp_Perm03	BOOL	User Permissive Input 3
Inp_Perm04	BOOL	User Permissive Input 4
Inp_Intlk01	BOOL	User Interlock Input 1
Inp_Intlk02	BOOL	User Interlock Input 2

Control Output Parameters

raS_Opr_BMS_Pilot AOI Control Output Parameters

Parameter Name	Data Type	Function
Out_LightOff	BOOL	Spark-Ignited Main Fuel Actuator to Light-Off Position
Out_Hold	BOOL	Spark-Ignited Main Fuel Actuator Hold at Current Position
Out_Release	BOOL	Spark-Ignited Main Fuel Actuator Released for Modulation
Val_eState	DINT	Pilot Control State Engine Enumeration Value
Val_TimeRemaining	REAL	Current Activity Time Remaining (seconds)
Val_FlameA	REAL	Flame Sensor A Strength Value
Val_FlameB	REAL	Flame Sensor B Strength Value
Val_FlameC	REAL	Flame Sensor C Strength Value

Status and Information Output Parameters

raS_Opr_BMS_Pilot AOI Status and Information Output Parameters

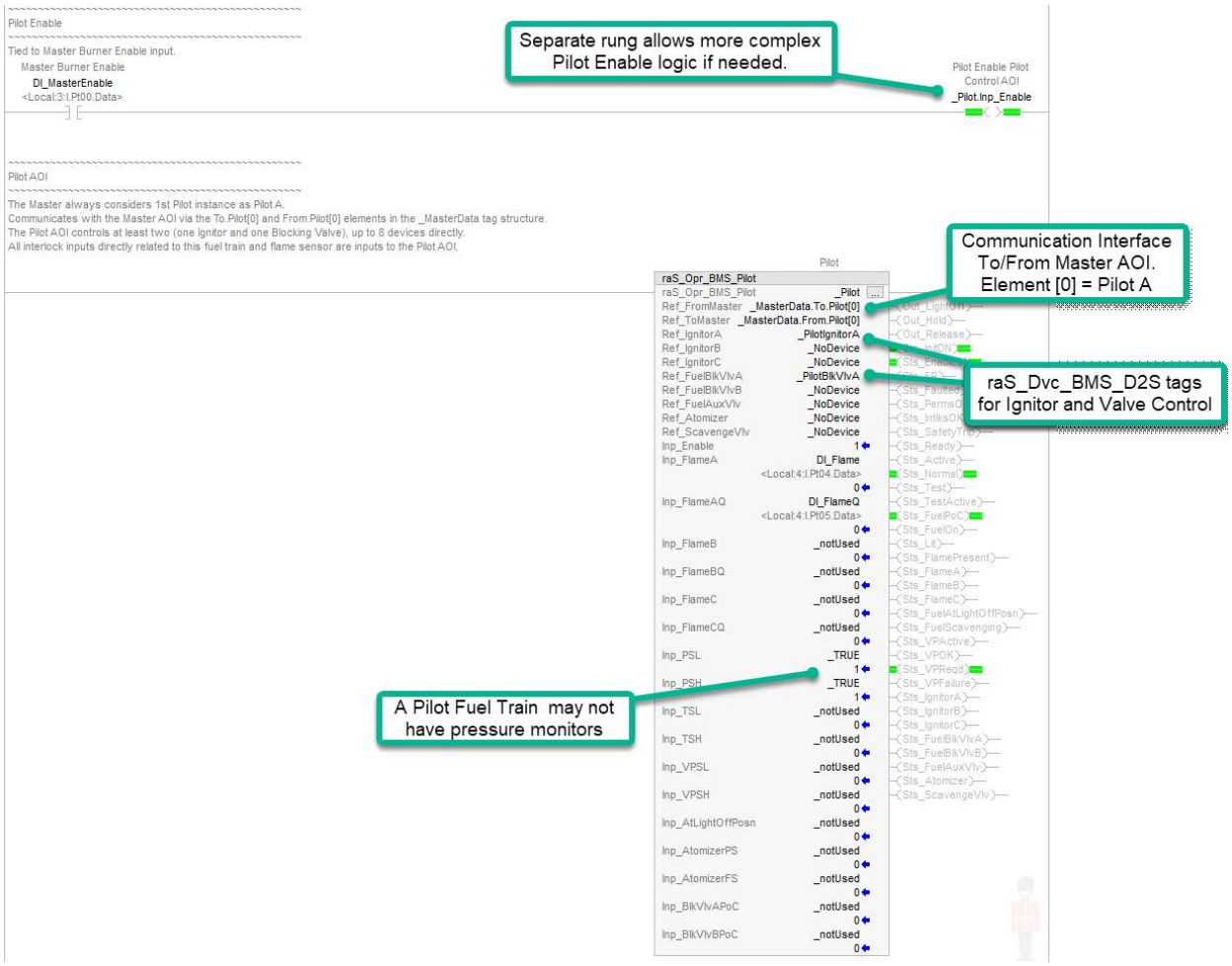
Parameter Name	Data Type	Function
Sts_InitDN	BOOL	First Scan Initialization Complete
Sts_Enabled	BOOL	Pilot Control Enabled
Sts_ER	BOOL	Configuration Error Operation is Disabled
Sts_bCfgErr	DINT	Configuration Error Status (bit-mapped)
Sts_Faulted	BOOL	Device Fault Exists
Sts_bFaults	DINT	Fault Status Word (bit-mapped)
Sts_PermsOK	BOOL	Fuel Sequence Permissives OK
Sts_bPerms	DINT	Permissives Lost Status Word (bit-mapped)
Sts_IntlksOK	BOOL	All Interlocks OK
Sts_bIntlks	DINT	Interlocks Lost Status Word (bit-mapped)
Sts_SafetyTrip	BOOL	Interlock Safety Trip Reset Required
Sts_bSafetyTrip	DINT	Safety Trip First Out Capture (bit-mapped)
Sts_Ready	BOOL	Ready To Operate
Sts_Active	BOOL	Pilot Control is Active
Sts_Normal	BOOL	Normal Command Mode
Sts_Test	BOOL	Maintenance Test Mode Enabled
Sts_TestActive	BOOL	Fuel Device Maintenance Testing is Active
Sts_FuelPoC	BOOL	Fuel Blocking Valves Confirmed Closed
Sts_FuelOn	BOOL	Fuel to Burner Enabled
Sts_Lit	BOOL	Pilot Fuel Lit and Flame Stable
Sts_FlamePresent	BOOL	Flame Presence Sensed
Sts_FlameA	BOOL	Flame Sensed at Detector A
Sts_FlameB	BOOL	Flame Sensed at Detector B
Sts_FlameC	BOOL	Flame Sensed at Detector C
Sts_FuelAtLightOffPosn	BOOL	Spark-Ignited Main Fuel Actuator At Light-Off Position
Sts_VPActive	BOOL	Valve Proving Sequence Active
Sts_VPOK	BOOL	Valve Proving Test Successful
Sts_VPReqd	BOOL	Valve Proving Test Required Prior to Next Burner Light-Off
Sts_VPFailure	BOOL	Valve Proving Test Unsuccessful
Sts_IgnitorA	BOOL	Ignitor A Energized
Sts_IgnitorB	BOOL	Ignitor B Energized
Sts_IgnitorC	BOOL	Ignitor C Energized
Sts_FuelBlkVlvA	BOOL	Upstream Fuel Blocking Valve Energized
Sts_FuelBlkVlvB	BOOL	Downstream Blocking Valve Energized
Sts_FuelAuxVlv	BOOL	Auxiliary Fuel Valve (Vent or Recirc) Energized
Sts_Atomizer	BOOL	Atomizing Media Valve Energized
Sts_ScavengeVlv	BOOL	Fuel Scavenging Valve Energized
Sts_FuelGas	BOOL	Gasseous Fuel Selected
Sts_FuelOilLt	BOOL	Light Fuel Oil (#2, #4) Selected
Rdy_Reset	BOOL	Fault or Safety Trip Reset Required
Inf_Label	BOOL	Pilot Control (@Description Used by HMI)
Inf_Tag	BOOL	Tag ID (@Description Used by HMI)
Inf_BuildID	DINT	raS_Opr_BMS_Pilot AOI Build ID

Programming Example

General arrangement for this example is shown in the Figure 1 - Example Task, Program, and Routine Arrangement. The raS_Opr_BMS_Pilot, being a Safety AOI, is located in the BMS_Example program, c_Pilot routine. The rungs that follow are taken from that routine:

- Rung 1: Pilot Enable logic, tied to same input as Master AOI, shown here.
- Rung 2: raS_Opr_BMS_Pilot AOI, shown here.
- Rung 3: Rung for Ignitor control, example in raS_Dvc_BMS_D2S AOI section.
- Rung 4: Rung for Valve control, example in raS_Dvc_BMS_D2S AOI section.

Pilot Logic



Graphic Symbols

Graphic Symbols are used as launch buttons within HMI applications to open up faceplate displays. See Basic Launch Button Attributes section for details on configuration and indicators.

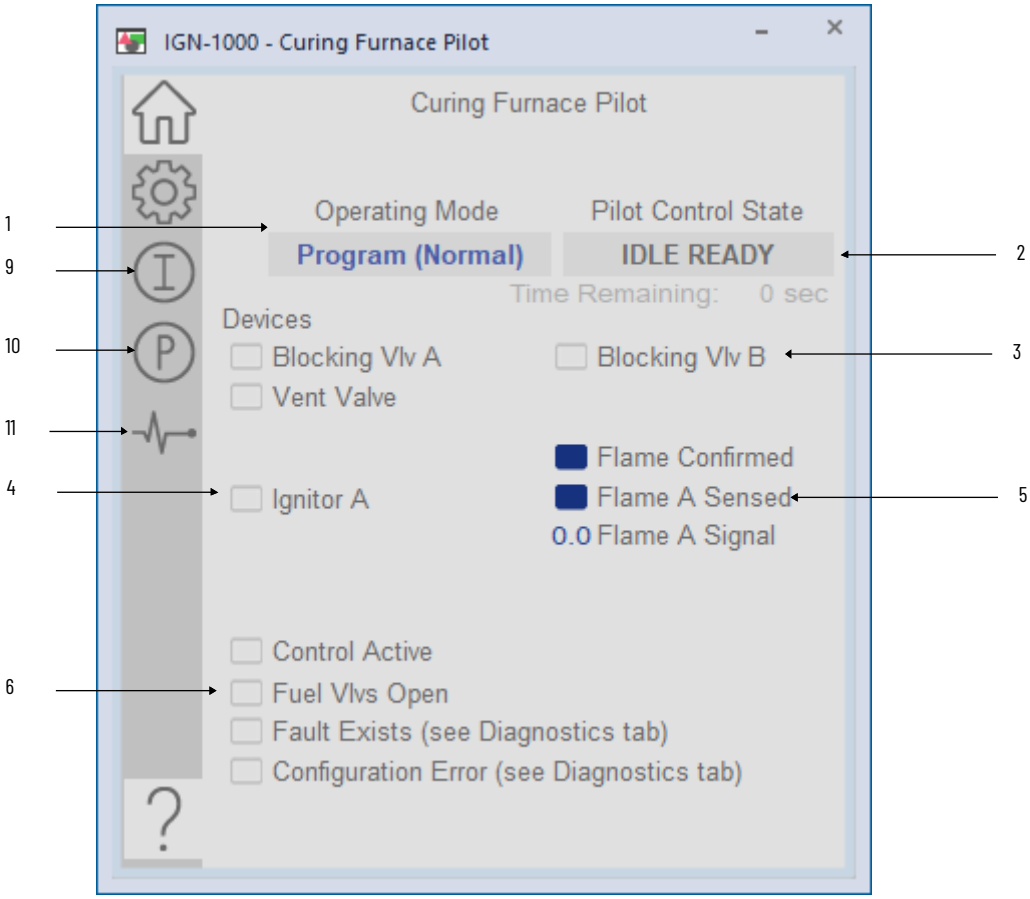
FactoryTalk® View SE Graphic Symbols

Graphic Symbol Name	Graphic Symbol	Description	Global Object Parameter Values
GO_PilotStatus		Pilot Status with faceplate navigation. This launch button graphic object allows the user to navigate to the object faceplate. The text on the button face is set to the tag's .Inf_Label.@Description field. Also shown are the operating state and Pilot time remaining in current operation/state.	#102: AOI Instance (e.g. {::[PAC]Program::SafetyProgram._Instance Name}) #103: HMI Control Tag #120: Faceplate X Position, pixels from left edge of display (e.g. /X100) #121: Faceplate Y Position, pixels from top edge of display (e.g. Y100)
GO_PilotStatusNoNav		Pilot Status without navigation. . The text on the object face is set to the tag's .Inf_Label.@Description field. Also shown are the operating state and Pilot time remaining in current operation/state	#102: AOI Instance (e.g. {::[PAC]Program::SafetyProgram._Instance Name})

Faceplates

There are basic faceplate attributes that are common across all instructions. See Basic Faceplate Attributes on page 21.

Home

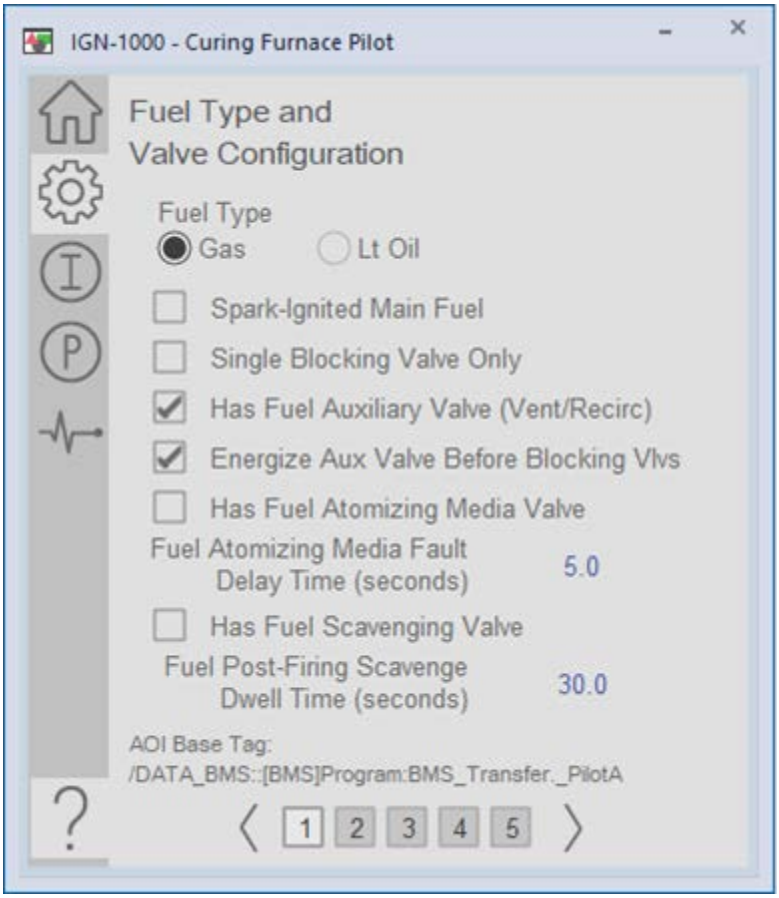


Item	Description
1	Pilot Operating Mode indicator.
2	Pilot Operating State indicator.
3	Devices status
4	Ignitor(s) status
5	Flame status
6	Status indicators
9	Pilot Interlock status summary indicator, showing all Interlock conditions OK.
10	Pilot Permissive status summary indicator, showing all Permissive conditions OK.
11	Pilot Master Diagnostic status summary indicator, showing all Diagnostic conditions OK.

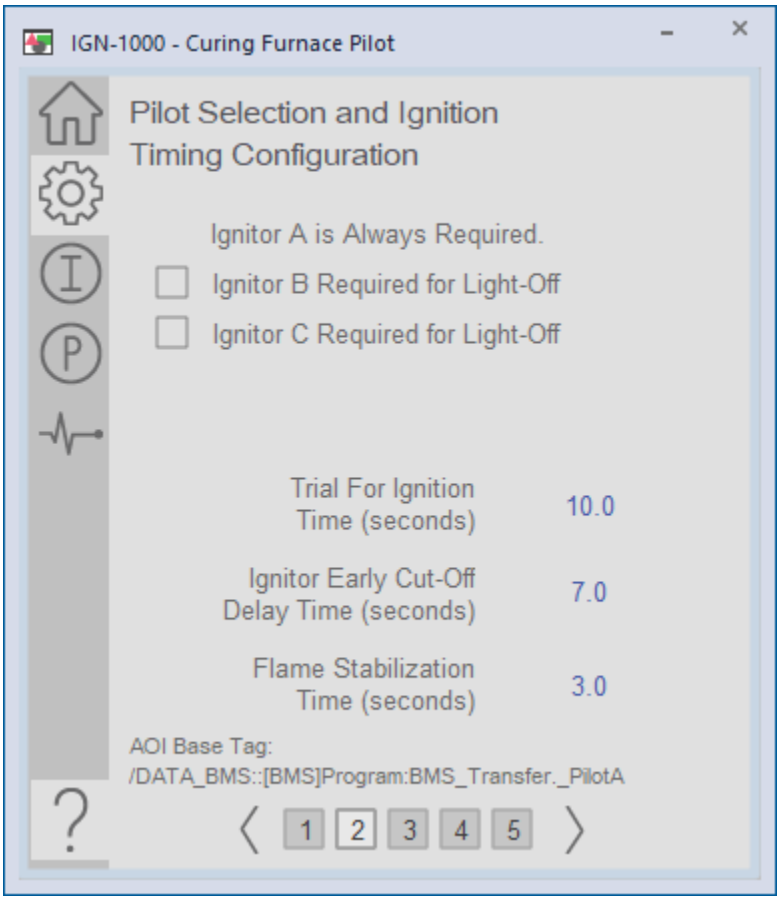
Configuration

There are five pages within the configuration tab.

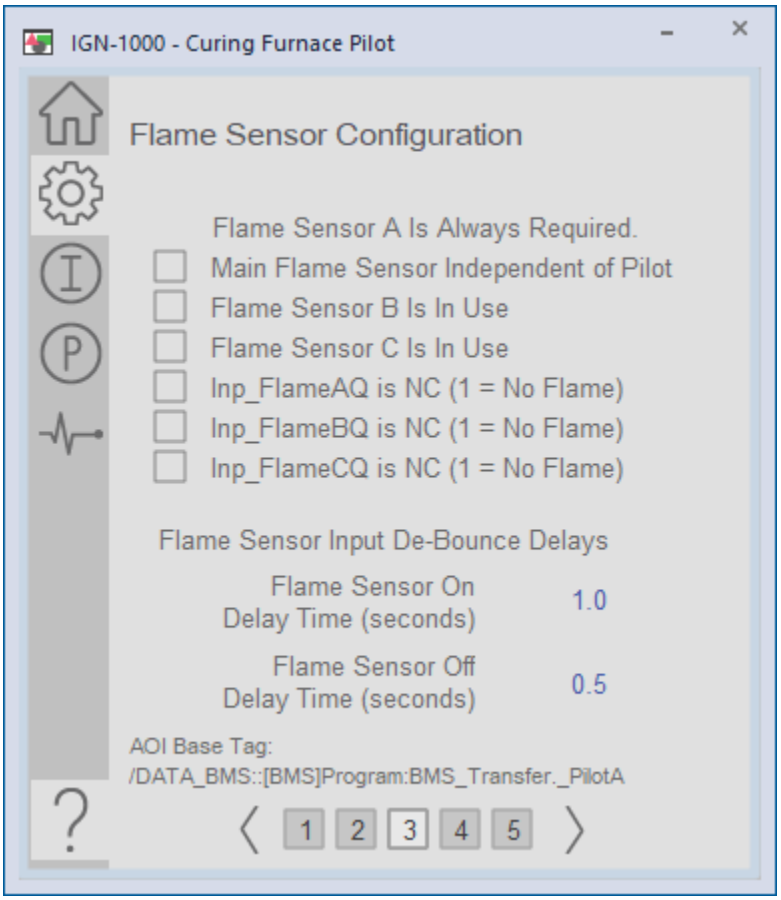
Configuration Page 1 - Fuel Type and Valve Configuration



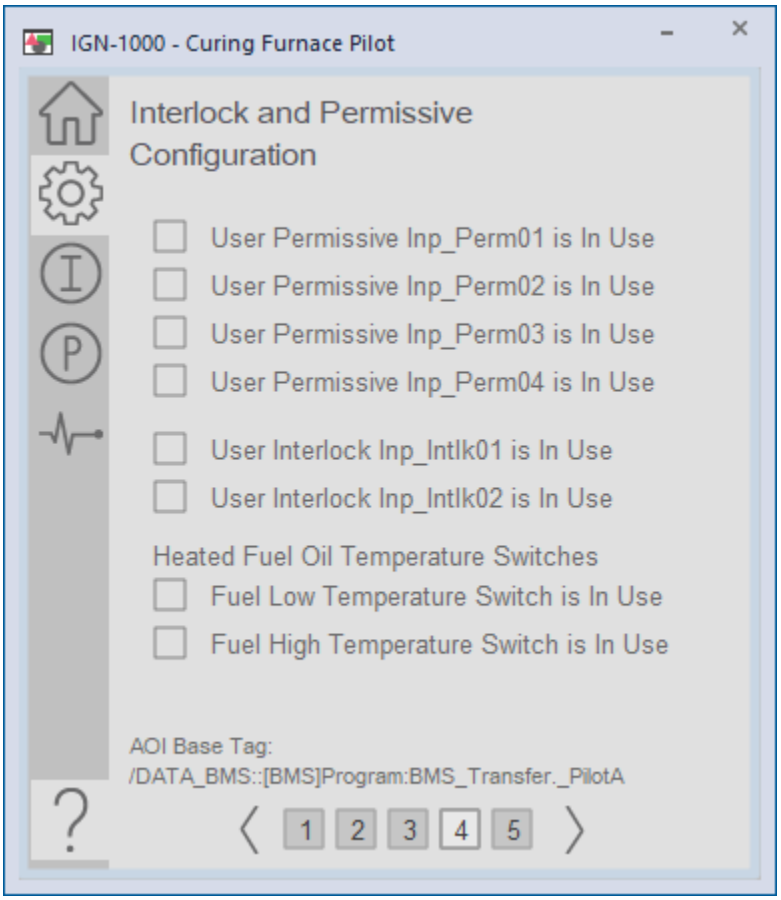
Configuration Page 2 - Pilot Selection and Ignition Timing Configuration



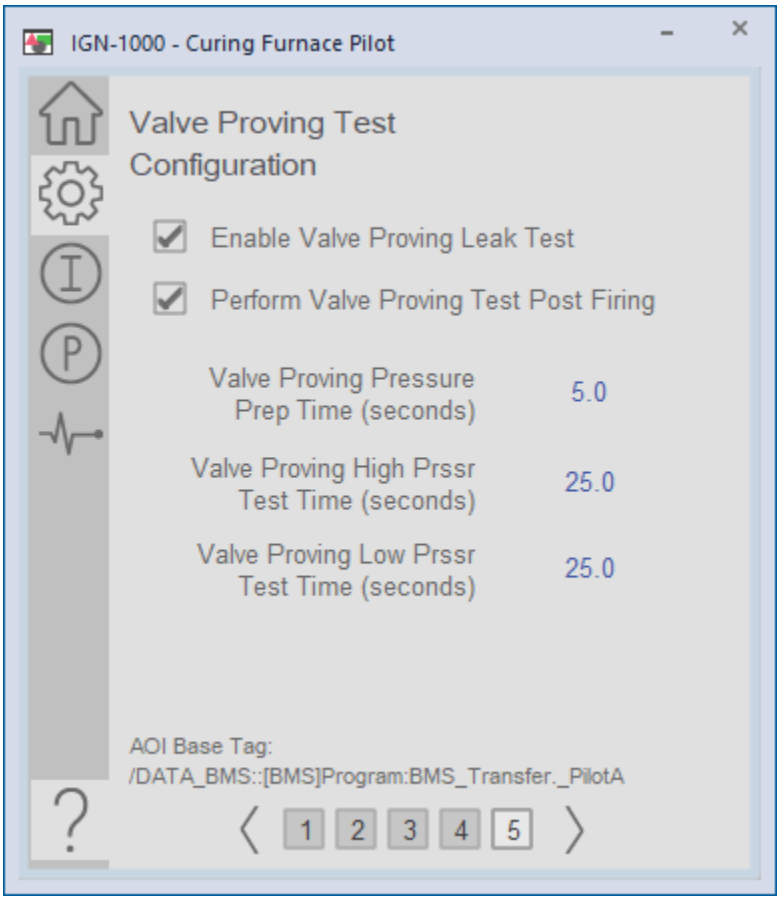
Configuration Page 3 - Flame Sensor Configuration



Configuration Page 4 - Interlock and Permissive Configuration



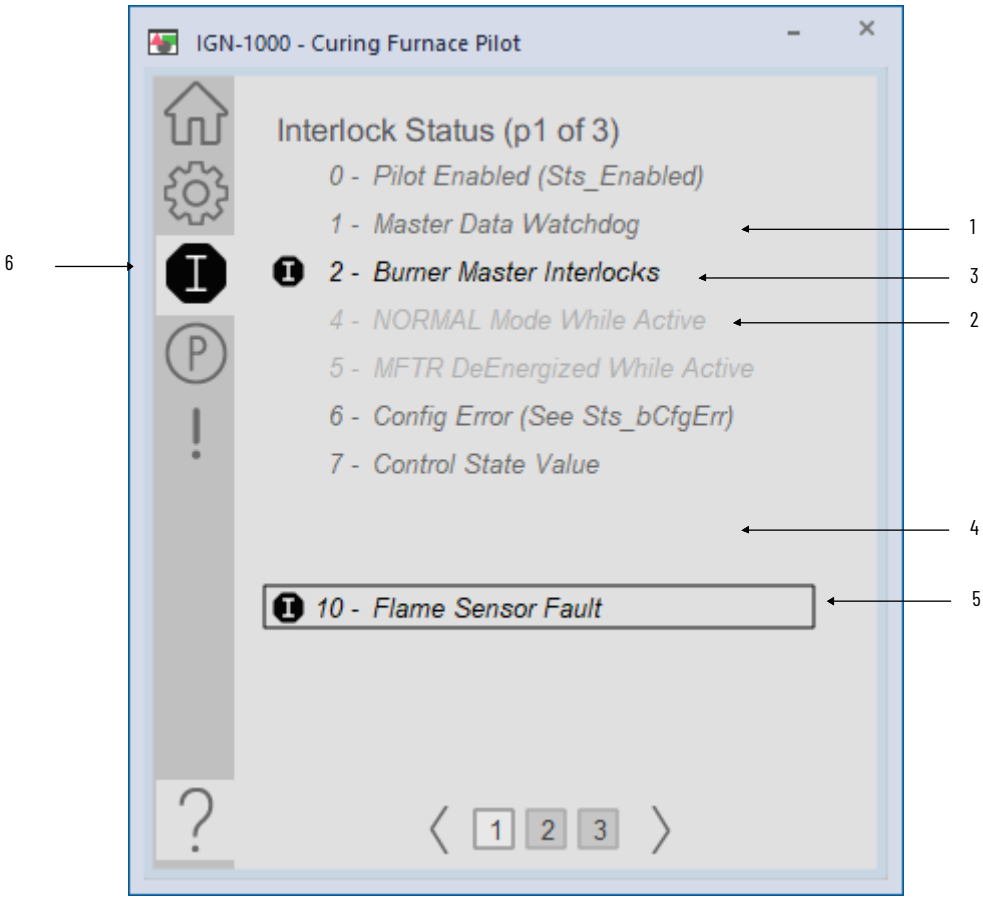
Configuration Page 5 - Valve Proving Test Configuration



Interlock Status

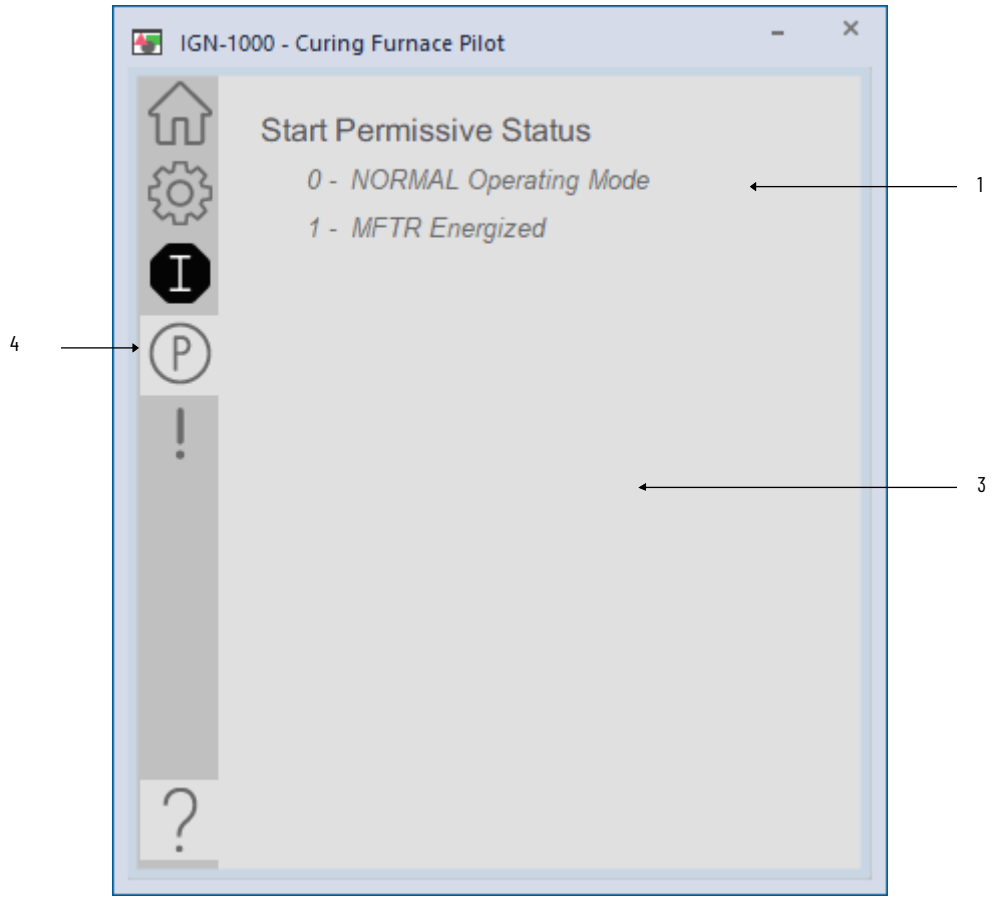
There are 3 pages within the Interlock Status tab - each with 12 interlocks.

Interlock Status Page 1 - Interlocks 0-11



Item	Description
1	Interlock currently monitored, but OK to operate.
2	Interlock configured for use, but currently not monitored.
3	Interlock currently monitored, but NOT OK to operate.
4	Interlock not configured for use (User Defined Interlock Input 1 in this case).
5	Interlock currently monitored, NOT OK, and has generated a Safety Trip.
6	Summary indicator – at least one Interlock is NOT OK.

Permissive Status

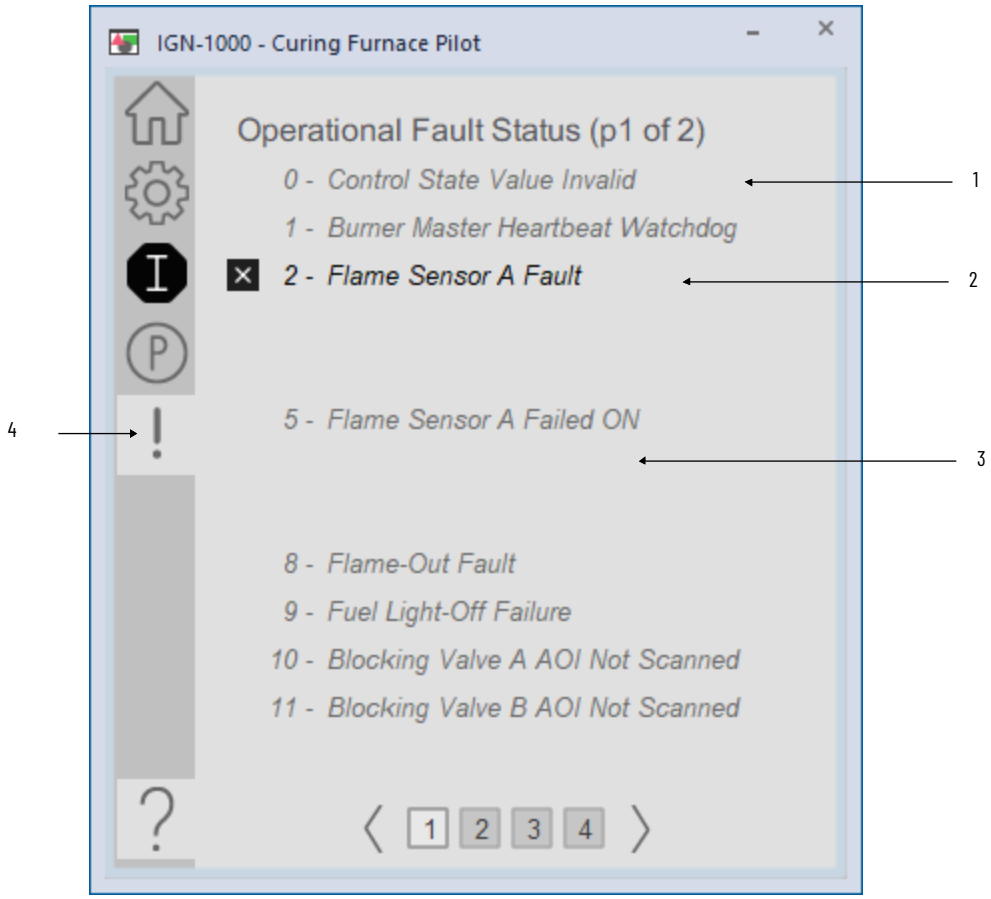


Item	Description
1	Permissive currently monitored, but OK to Start Operation.
3	Permissive not configured for use (User Permissive Input 1 in this case).
4	Summary indicator – at least one Permissive condition is NOT OK.

Diagnostics

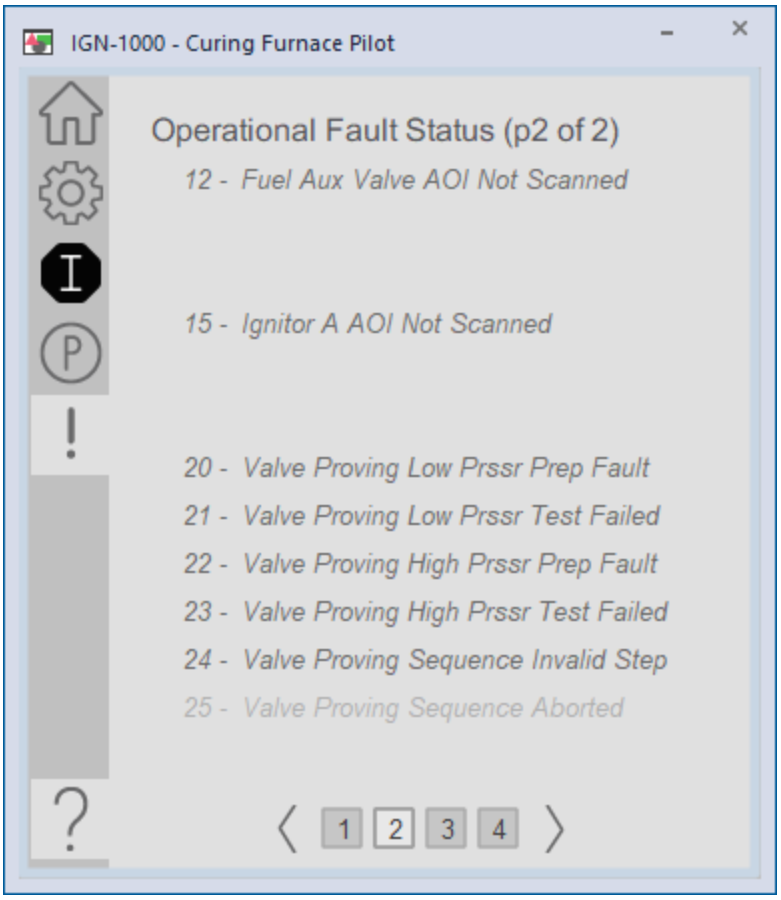
There are 4pages within the diagnostics tab.

Diagnostics Page 1 - Operational Fault Status (1 of 2)

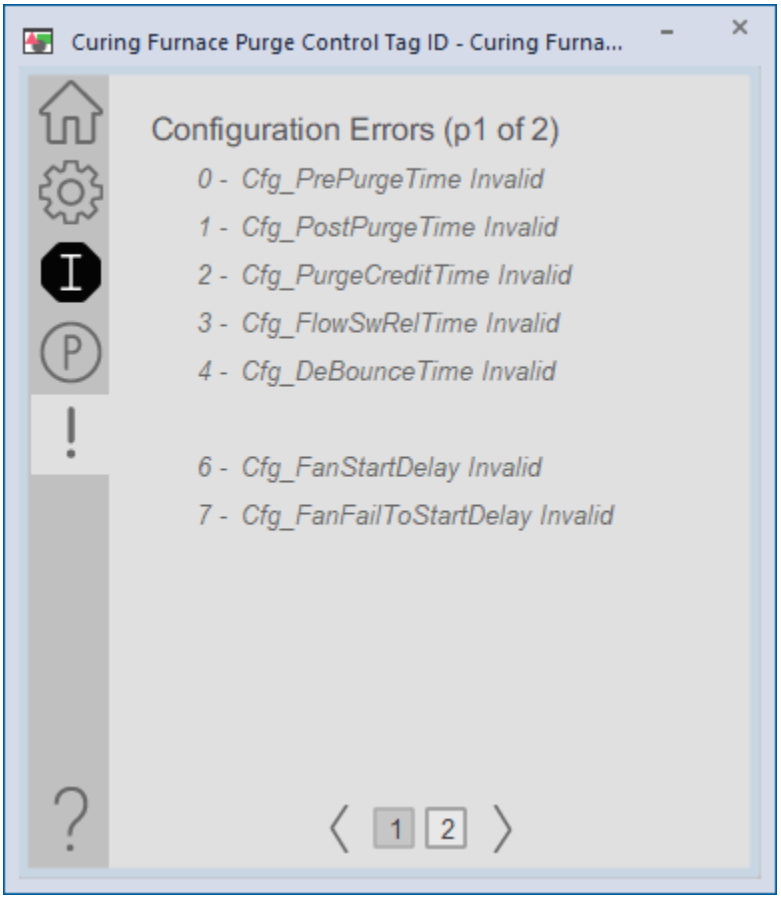


Item	Description
1	Operational Fault currently monitored, and OK to Operate.
2	Operational Fault currently monitored, and NOT OK to Operate.
3	Operational Fault not configured for use (Pilot B Heartbeat Watchdog in this case).
4	Summary indicator – at least one Diagnostic condition is NOT OK.

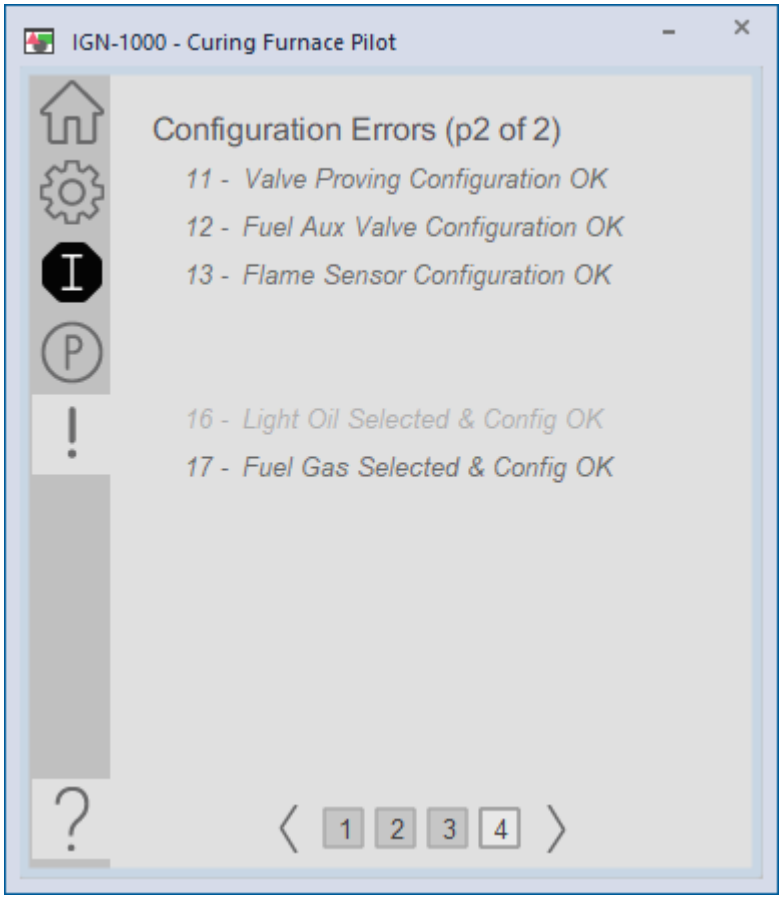
Diagnostics Page 2 - Operational Fault Status (2 of 2)



Diagnostics Page 2 - Configuration Errors (1 of 2)

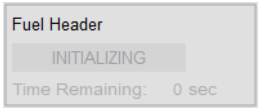
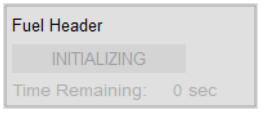
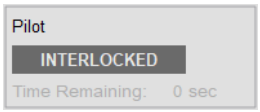
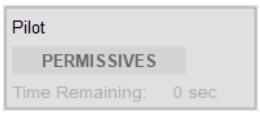


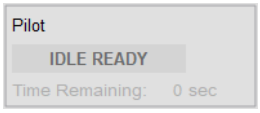
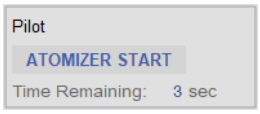
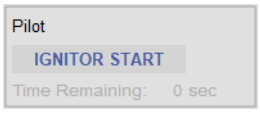
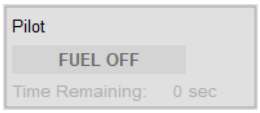
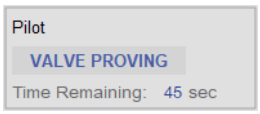
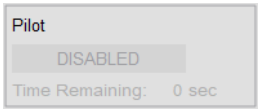
Diagnostics page 4 – Configuration Errors (2 of 2)



State Value Enumerations

The following table identifies the states available to the Pilot/Spark-Ignited Fuel when enabled (Sts_Enabled is True). If Sts_Enabled is False, the Pilot/Spark-Ignited Fuel is allowed to only be in States 0, 1, and 2, and will indicate DISABLED.

Val_eState	State Name	Description	HMI Graphic
0	Initializing	AOI performing 1st scan initialization.	
1	Safety Trip	Pilot Safety Trip present.	
2	Interlocked	Pilot Interlock not met.	
3	Not Ready	Pilot waiting for all Permissives OK.	

4	Idle / Ready	Pilot Idle and Ready for Start Command.	
5	Atomizer Prep	Atomizing Media Valve Energize and confirm media flow.	
6	Ignitor On	Energize Spark Ignitors.	
7	Fuel Ignition	Pilot Trial for Ignition.	
8	Flame Confirmed	Flame Presence confirmed.	
9	Fuel Scavenge	Timed fuel scavenging activity.	
10	Flame Off	Pilot fuel off, confirm loss of flame.	
11	Valve Proving	Blocking valve leak test active.	
0, 1, or 2		Pilot Disabled (Sts_Enabled = 0)	

BMS Fuel (raS_Opr_BMS_Fuel)

Overview

The Fuel AOI manages all interlock signals and devices associated with one Main Fuel train. A burner system supports as many as 4 independent Main Fuel trains, and an optional Fuel Header. The primary function of the Fuel AOI is safe activation and light-off of a Main Fuel using one or more pilot flames. Burners with spark ignited fuels only do not require a Main Fuel AOI but may have a Fuel Header.

Limitations

The following limitations must be considered when implementing this instruction within a BMS Library application.

AOI Class	Safety
Minimum Instances Required	0
Maximum Instances Allowed	4 (Main Fuel) + 1 (Fuel Header)

System Interaction

Each Fuel AOI exchanges data with the BMS Master AOI using a pair of UDT tags, one for data in each direction. The content of these UDTs is detailed in the sections following.

Fuel to Master Data

Data Type: raS_UDT_BMS_Fuel_ToMaster (12 bytes) array of 4 elements for Main Fuels, plus a single element for Fuel Header. Each Main Fuel component uses one element of the array of four, and the element used identifies to the Master which Fuel Instance (i.e. Fuel A, B, C, or D) the data is coming from. It is imperative that each Fuel component uses the same array index value for data to and from the Master. The Fuel Header component has its own element for data to the Master.

Fuel Parameter Reference: Ref_ToMaster (InOut)

Data from each Main Fuel and Fuel Header to Master includes the common component data structure, fuel operational status, pilot requests, pertinent configuration data, and command acknowledge bits. A list of UDT elements is

shown in the following table. Fuel Component in the following table refers to Main Fuel and Fuel Header components alike.

Fuel to Master Data Elements

Element Name	Data Type	Description
Ref_ToMaster	raS_UDT_BMS_Fuel_ToMaster	Fuel Component Status to Master
Ref_ToMaster.Sts	raS_UDT_BMS_CommonStatus	Fuel Component Common Status to Master
Ref_ToMaster.Sts.Enabled	BOOL	Fuel Component Enabled
Ref_ToMaster.Sts.SafetyTrip	BOOL	Fuel Component Generated Safety Trip
Ref_ToMaster.Sts.IntlkOK	BOOL	Fuel Component Interlocks OK
Ref_ToMaster.Sts.Ready	BOOL	Fuel Component Ready for Start
Ref_ToMaster.Sts.Active	BOOL	Fuel Component Operating Sequence is Active
Ref_ToMaster.Sts.Normal	BOOL	Fuel Component in NORMAL MODE
Ref_ToMaster.Sts.TestMode	BOOL	Fuel Component in TEST MODE
Ref_ToMaster.Sts.TestActive	BOOL	Fuel Component in TEST MODE with Device Test Active
Ref_ToMaster.Sts.eState	SINT	Fuel Component Enumerated State Value
Ref_ToMaster.Sts.HBEcho	INT	Fuel Component Communication Heartbeat Echo
Ref_ToMaster.Sts.BuildID (hidden parameter)	DINT	raS_Opr_BMS_Fuel AOI Build ID
Sts_FuelPoC	BOOL	All Fuel Blocking Valves Closed
Sts_AtLightOff	BOOL	Fuel Actuator at Light Off Position
Sts_FuelEnabled	BOOL	Fuel Blocking Valves Energized
Sts_FlamePresent	BOOL	Flame Sensed by Fuel Flame Sensor
Sts_Release	BOOL	Fuel Lit and Flame Stable
Sts_VPActive	BOOL	Valve Proving Test Sequence Active
Sts_VPReqd	BOOL	Valve Proving Test Required Prior to Light-Off
Sts_Scavenging	BOOL	Fuel Scavenging Active
Req_PilotA	BOOL	Pilot A Request
Req_PilotB	BOOL	Pilot B Request
Req_PilotC	BOOL	Pilot C Request
Req_PilotD	BOOL	Pilot D Request
Cfg_Exists	BOOL	Fuel AOI Exists
Cfg_HasVP	BOOL	Fuel Has and Requires Valve Proving
Cfg_HasScavenge	BOOL	Fuel Has and Requires Post-Firing Scavenging
Ack_Start	BOOL	Acknowledge Start Request
Ack_Stop	BOOL	Acknowledge Stop Request
Ack_Reset	BOOL	Acknowledge Reset Request
Ack_VPStart	BOOL	Acknowledge Valve Proving Sequence Start Request
Ack_Scavenge	BOOL	Acknowledge Manual Fuel Scavenge Request

Master to Fuel Data

Data Type: UDT_MasterToFuel (8 bytes) array of 4 elements for Main Fuels, plus a single element for Fuel Header. Each Main Fuel component uses one element of the array of four, and the element used identifies to the Master which Fuel Instance (i.e. Fuel A, B, C, or D) is receiving the data. It is imperative that each Fuel component uses the same array index value for data to and from the Master. The Fuel Header component has its own element. It is imperative that the Fuel Header component uses the designated Fuel Header element for data to and from the Master.

Fuel Parameter Reference: Ref_FromMaster (InOut)

Data from Master to Fuel includes the SysSts structure, start/stop/reset commands, and information gathered from pilot, purge, and fuel AOIs necessary for fuel operation. Settings within this data configure the raS_Opr_BMS_Fuel AOI for Main Fuel or Fuel Header duty. A list of UDT elements is shown in the following table. Fuel Component in the following table refers to Main Fuel and Fuel Header components alike.

Master to Fuel Data Elements

Element Name	Data Type	Description
Ref_FromMaster	raS_UDT_BMS_Master_ToFuel	Data from Master to Fuel component.
Ref_FromMaster.SysSts	raS_UDT_BMS_Master_SysSts	System-wide common status information.
Ref_FromMaster.SysSts.Enable	BOOL	Burner System Enable
Ref_FromMaster.SysSts.IntlksOK	BOOL	Burner System Interlocks OK
Ref_FromMaster.SysSts.MFTR	BOOL	Master Fuel Trip Relay Energized
Ref_FromMaster.SysSts.Normal	BOOL	Burner System in NORMAL Operating Mode
Ref_FromMaster.SysSts.TestMode	BOOL	Burner System in TEST Mode
Ref_FromMaster.SysSts.TestActive	BOOL	Burner in TEST Mode with a Device Test Active
Ref_FromMaster.SysSts.NoFlame	BOOL	Pre-Purge NOT Completed – No Flame Allowed
Ref_FromMaster.SysSts.eState	SINT	Burner System Master State Value
Ref_FromMaster.SysSts.HB	INT	Master Communication Heartbeat
Ref_FromMaster.Cmd_Start	BOOL	Fuel Sequence Start Request
Ref_FromMaster.Cmd_Stop	BOOL	Fuel Sequence Stop Request
Ref_FromMaster.Cmd_Reset	BOOL	Fuel Reset Request
Ref_FromMaster.Cmd_VPStart	BOOL	Manual Fuel Valve Proving Sequence Start Request
Ref_FromMaster.Cmd_Scavenge	BOOL	Manual Fuel Scavenge Request
Ref_FromMaster.Inp_PilotRdyA	BOOL	Pilot A Ready for Fuel Light-Off or Scavenge
Ref_FromMaster.Inp_PilotRdyB	BOOL	Pilot B Ready for Fuel Light-Off or Scavenge
Ref_FromMaster.Inp_PilotRdyC	BOOL	Pilot C Ready for Fuel Light-Off or Scavenge
Ref_FromMaster.Inp_PilotRdyD	BOOL	Pilot D Ready for Fuel Light-Off or Scavenge
Ref_FromMaster.Inp_PilotIntlk	BOOL	Required Pilot(s) Interlocked
Ref_FromMaster.Inp_ActuatorHold	BOOL	Fuel Actuator Hold Current Position
Ref_FromMaster.Inp_ActuatorRel	BOOL	Fuel Actuator Released for Modulation
Ref_FromMaster.Inp_HeaderNotRdy	BOOL	Fuel Header is Not Ready to supply main fuel to burner.
Ref_FromMaster.Sts_HasPilotA	BOOL	Pilot A Exists
Ref_FromMaster.Sts_HasPilotB	BOOL	Pilot B Exists
Ref_FromMaster.Sts_HasPilotC	BOOL	Pilot C Exists
Ref_FromMaster.Sts_HasPilotD	BOOL	Pilot D Exists
Ref_FromMaster.Sts_IsFuelHeader	BOOL	1=Fuel Header, 0=Main Fuel

Fuel AOI Configuration

The Fuel AOI configuration includes fuel type, required interlock devices, fuel train valve configuration, required pilot(s) for light-off, and number/type of flame sensors. Parameters for valve proving, and fuel scavenging configuration are also provided. The Fuel AOI provides up to 2 user-assignable interlock inputs and up to 4 user-assignable permissive inputs that are enabled by configuration. Configuration parameter details are shown below.

Interlocks

The raS_Opr_BMS_Fuel AOI monitors 28 interlock conditions, 2 of which are user selectable. The following table identifies each interlock condition, when each is monitored, and when each will trigger a Safety Trip.

Fuel AOI Interlock List

Interlock Condition	Interlock Enable	Safety Trip Enable
Fuel Component Disabled	Always	Fuel Component Active
Master Comm Heartbeat Watchdog Timeout	Always	Fuel Component Active
System Interlock Exists	Always	Never
Normal Mode Lost (Sts_Normal = 0)	Fuel Component Active	Same as Interlock Enable
MFTR Lost	Fuel Component Active	Same as Interlock Enable
Configuration Error Exists (Sts_bCfgErr <> 0)	Fuel Component Enabled	Fuel Component Active
Invalid State Value	Fuel Component Enabled	Fuel Component Active
Flame Sensor Fault	Fuel Component Enabled and Not Fuel Header	Same as Interlock Enable
Pilot Failed	Fuel Component Enabled and Pilot Required	Same as Interlock Enable
Fuel Actuator Not at Light-Off (Inp_AtLightOffPosn = 0)	Light-Off Pos'n Required and Not Fuel Header	Same as Interlock Enable
Light-Off Flame Failure Fault	Fuel Component Enabled	Same as Interlock Enable
Flame-Out Fault	Fuel Component Enabled	Same as Interlock Enable
Flame Sensor Failed On Fault	Fuel Component Enabled	Same as Interlock Enable
Fuel Low Pressure Trip (Inp_PSL = 0)	Fuel Component Enabled	Same as Interlock Enable
Fuel High Pressure Trip (Inp_PSH = 0)	Fuel Component Enabled	Same as Interlock Enable
Fuel Low Temperature Trip (Inp_TSL = 0)	Fuel Component Enabled and Has Low Temperature Switch	Same as Interlock Enable
Fuel Low Temperature Trip (Inp_TSH = 0)	Fuel Component Enabled and Has High Temperature Switch	Same as Interlock Enable
Atomizing Pressure Loss (Inp_AtomizerPS = 0)	Fuel Component Enabled and Has Atomizing	Same as Interlock Enable
Atomizing Flow Loss (Inp_AtomizerFS = 0)	Fuel Component Enabled and Has Atomizing	Same as Interlock Enable
Valve Proving Sequence Failure	Fuel Component Enabled and Has Valve Proving	Same as Interlock Enable
Fuel Header Tripped	Fuel Component Active and Not Fuel Header	Same as Interlock Enable
Blocking Valve A Failure	Fuel Component Enabled	Fuel Component Active or Valve Proving
Blocking Valve B Failure	Fuel Component Enabled and Has Blocking Valve B	Fuel Component Active or Valve Proving
Fuel Auxiliary Valve Failure	Fuel Component Enabled and Has Fuel Auxililary Valve	Fuel Component Active or Valve Proving
Atomizing Valve Failure	Fuel Component Enabled and Has Atomizing	Fuel Component Active
Scavenging Valve Failure	Fuel Component Enabled and Has Fuel Scavenging	Fuel Component Active
User Interlock 01 Not OK (Inp_Intlk01 = 0)	Fuel AOI Enabled and Has User Interlock 01	Same as Interlock Enable
User Interlock 02 Not OK (Inp_Intlk02 = 0)	Fuel AOI Enabled and Has User Interlock 02	Same as Interlock Enable

Permissives

The raS_Opr_BMSFuel AOI monitors 7 permissive conditions, 4 of which are user selectable. The following table identifies each permissive condition and when each is monitored. Once the Fuel AOI transitions to an Active state, permissive conditions no longer impact operation.

Fuel AOI Permissive List

Permissive Lost Condition	Permissive Enable
Not NORMAL Operating Mode (Sts_Normal = 0)	Fuel Component Not Active
MFTR NOT Energized	Fuel Component Not Active
Fuel Actuator Not at Light-Off	Fuel Component Not Active
Fuel Header Not Ready	Fuel Component Not Active and Not Fuel Header
User Permissive 01 Not OK (Inp_Perm01 = 0)	Fuel Component Not Active and Has User Permissive 01
User Permissive 02 Not OK (Inp_Perm02 = 0)	Fuel Component Not Active and Has User Permissive 02
User Permissive 03 Not OK (Inp_Perm03 = 0)	Fuel Component Not Active and Has User Permissive 03
User Permissive 04 Not OK (Inp_Perm04 = 0)	Fuel Component Not Active and Has User Permissive 04

Operations

Operation Modes

The BMS Library objects provide only physical operation mode. There is no virtual/simulated mode offered.

Normal Operation Mode

The Master must be in NORMAL Mode to operate the burner and this mode status is transmitted to the Fuel component. While in NORMAL Mode, the Fuel component receives requests from the Master, making appropriate operational changes based on the command received. There is no direct interaction with the Fuel component except through the Master's Data Interface.

NORMAL Mode status is echoed back to the Master indicating whether the Fuel component is in NORMAL Mode or not.

Test Enable Mode Operation

The Fuel component is in TEST ENABLE Mode any time the Master is in TEST ENABLE Mode. Any commands to start or otherwise activate the Fuel component are ignored.

TEST ENABLE Mode status is echoed back to the Master indicating whether the Fuel component is in TEST ENABLE Mode or not.

Device Test Active

Device Test Active indicates a device under control of the Fuel component is actively being tested. Any time a device under control of this Fuel component is actively being tested, testing of all other devices is immediately disabled and the Fuel component indicates Device Test Active. This status is sent to the Master.

Operation

The Fuel AOI is intended to support gaseous fuels (e.g. natural gas or LP gas) and light or heavy fuel oil.

The Fuel AOI controls up to 5 physical devices: 1 to 5 valves. At least one valve is required.

The Fuel AOI supports 1 or 2 flame sensors and must use at least one. This flame sensor can be independent of the pilot flame sensor, or the main fuel can use the same sensor(s) as the pilot.

Gas fuel trains typically use 1, 2, or 3 valves: 1 or 2 fuel blocking valves, and an optional vent between 2 blocking valves. The Fuel AOI dedicates device

references to the three valve functions. A Valve Proving sequence is also available for gas fuels to confirm the blocking valves seal when closed.

Oil gas trains may also have 1 or 2 blocking valves, but instead of a vent, the third valve is used for recirculation. Selecting oil as the fuel alters the valve sequencing to accommodate. Oil may require fuel atomization so one device is dedicated to that function and included in the control sequencing as appropriate. The oil fuel train may require scavenging at shut down so one device is dedicated to that function and included in the control sequencing as appropriate.

Typical Main Fuel Control Sequence:

1. Continually monitor all interlocks directly associated with the fuel supply function.
2. A start request is received from the Master, kicking off the fuel sequence.
3. If the fuel requires atomizing, the atomizing media valve is energized and flow confirmed.
   a. Atomizing media flow is confirmed by flow or pressure switch
   b. If atomizing is not required, this step is skipped.
4. A request to the Master for required pilot flame(s) is made.
   a. All requested pilots must be confirmed and maintained throughout the main fuel light-off.
5. Master confirms requested pilots are ready to continue Main Fuel light-off.
6. Fuel blocking valves are energized to admit fuel to burner.
   a. If vent is used, the vent valve is closed before blocking valves open.
   b. If recirc is used, the blocking valves are energized first, then recirc is closed.
7. After a configurable delay not to exceed what is allowed by code, the pilot request is turned off and the flame sensor is checked for flame presence.
   a. Failure to sense flame by the end of this step triggers a Light-Off Failure safety trip.
8. Another configurable delay can be used to ensure flame stability before indicating to the Master that the fuel is ready for modulation.
9. Once flame presence is confirmed and Light-Off is complete, any future flame loss triggers a Flame-Out safety trip unless blocking valves are closed first.
10. Fuel ready for modulation signal is sent to the Master.
    a. Fuel stays active until a stop request is made.
11. Stop request from Master
    a. Blocking valves are closed, vent or recirc is opened.
    b. Modulate signal to Master is removed.
12. If scavenging is required, the pilot is requested
    a. Once required pilot is confirmed by Master, the scavenge valve is opened.
    b. Timed activity
    c. Once scavenging timer is done, turn off spark and close scavenging valve.
    d. Note that fuel scavenging is intended to safely burn any fuel remaining in burner feed piping, but a continuous and steady flame

cannot be guaranteed. Thus, flame confirmation is not reliable for this activity, and flame loss will not generate any type of trip during fuel scavenging.

e. If scavenging not required, this step is skipped

13. Fuel AOI returns to idle.

Fuel Header Operation

A Fuel AOI can be configured to control a header supplying multiple Main Fuel entry points into the burner combustion chamber. Each entry point will be considered a separate fuel under control of its own Fuel AOI. The control sequence for a Fuel Header will not include atomization or scavenging, nor will it require flame sensors. An abbreviated control sequence is used:

Typical Fuel Header Control Sequence:

1. Continually monitor all interlocks directly associated with the fuel supply function.
2. A start request is received from the Master, kicking off the fuel sequence.
3. Fuel blocking valves are energized to admit fuel to the fuel supply header.
   a. If vent is used, the vent valve is closed before blocking valves open.
   b. If recirc is used, the blocking valves are energized first, then recirc is closed.
4. A configurable delay can be used to ensure supply stability.
5. Fuel Header ready signal is sent to the Master.
   a. Fuel header stays active until a stop request is made.
6. Stop request from Master
   a. Blocking valves are closed, vent or recirc is opened.
   b. Fuel Header Ready signal to Master is removed.
7. Fuel AOI returns to idle.

Blocking Valve Proving Sequence

For gaseous fuels, a Valve Proving sequence can be selected to run either before fuel light-off or after fuel shutdown. This sequence is not intended to supersede annual manual valve leak testing as required by code but does provide a tacet confirmation that the blocking valves have not failed and are closing on request. A Valve Proving test failure generates a safety trip, removing that fuel from service until the trip is reset.

The Valve Proving test relies on two digital inputs tied to pressure switches monitoring the cavity between the fuel blocking valves. One switch monitors a high pressure threshold, and the other monitors a low pressure threshold. Both thresholds are tested in sequence to confirm both blocking valves can seal and do not have significant leakage failures.

Typical Valve Proving Sequence

1. Confirm both blocking valves are deenergized to start.
2. Energize the auxiliary vent valve to close it.
   a. Skip this step if the valve train does not include a vent.

3. Open the downstream blocking valve to release any latent pressure between the blocking valves.
   a. Both pressure switches must indicate deenergized inputs (pressure is below the threshold for both switches).
   b. Failure of either or both pressure switches to open constitutes a sequence failure. Generate a safety trip and abort this sequence.
4. After a few seconds, close the downstream blocking valve and monitor the low pressure switch.
   a. If at any time during this step the low pressure switch is activated, the upstream blocking valve has failed and is leaking fuel into the test chamber. Generate a safety trip and abort this sequence.
5. After the low pressure test time, energize the upstream blocking valve to pressurize the test chamber.
   a. Both pressure switches must indicate energized inputs (pressure is above the threshold for both switches).
   b. Failure of either or both pressure switches to close constitutes a sequence failure. This may indicate a severely leaking vent valve. Generate a safety trip and abort this sequence.
6. After a few seconds, close the upstream blocking valve and monitor the high pressure switch.
   a. If at any time during this step the high pressure switch is deactivated, the downstream blocking valve may have failed and is leaking fuel out of the test chamber. Generate a safety trip and abort this sequence.
   b. Note that a leaking vent valve may also cause this test to fail. Additionally, a severely leaking vent valve may mask a leakage failure of the upstream blocking valve.
7. After the high pressure test time, the test is complete.
   a. Deenergize the auxiliary vent valve if it exists.

Required Files

Objects include Add-On Instructions (AOIs) and HMI faceplates. The revision number (e.g. 1.01) used in filenames can change as new revisions are created.

Controller Files

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

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

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

Device/Item	Add-On Instruction	ROUTINE Import
BMS Interface	raS_Opr_BMS_Fuel_1.00_AOI.L5X	raS_Opr_BMS_Fuel_1.00_SftyROUTINE.L5X

FactoryTalk® View HMI Files

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

Device/Item	Type	FactoryTalk® View SE Faceplate
BMSMaster Faceplate	Display	(raS-1_00-SE) raS_Opr_BMS_Fuel-Faceplate.gfx
Graphic Symbols	Global Object	(raS-1-SE) Graphic Symbols - BMS.ggfx
Toolbox	Global Object	(raS-1-SE) Toolbox - BMS.ggfx

Add-On Instruction I/O Data

The following tables identify raS_Opr_BMS_Fuel AOI parameters.

InOut Parameters

:raS_Opr_BMS_Fuel AOI InOut Reference Parameters

Parameter Name	Data Type	Function
Ref_FromMaster	raS_UDT_BMS_Master_ToFuel	Command Data from Master to Fuel Control
Ref_ToMaster	raS_UDT_BMS_Fuel_ToMaster	Status Data From Fuel Control to Master
Ref_FuelBlkVlvA	raS_Dvc_BMS_D2S	Fuel Train Downstream Blocking Valve Device Tag
Ref_FuelBlkVlvB	raS_Dvc_BMS_D2S	Fuel Train Upstream Blocking Valve Device Tag
Ref_FuelAuxVlv	raS_Dvc_BMS_D2S	Fuel Train Auxiliary Valve Device Tag
Ref_Atomizer	raS_Dvc_BMS_D2S	Atomizing Media Valve Device Tag
Ref_ScavengeVlv	raS_Dvc_BMS_D2S	Fuel Scavenging Valve Device Tag

Configuration Parameters

raS_Opr_BMS_Fuel AOI Configuration Parameters

Parameter Name	Data Type	Function
Cfg_FuelType	DINT	0=Gasseous 1=Light Oil 2=Heavy Oil
Cfg_PilotA	BOOL	Pilot A is Required
Cfg_PilotB	BOOL	Pilot B is Required
Cfg_PilotC	BOOL	Pilot C is Required
Cfg_PilotD	BOOL	Pilot D is Required
Cfg_FullStrokeTime	REAL	Fuel Actuator Full Stroke Time at Max Rate (seconds)
Cfg_PilotIntermittent	BOOL	Pilot Not Shut Down After Light-Off
Cfg_HasFlameB	BOOL	Flame Sensor B Exists
Cfg_FlameAQNC	BOOL	Inp_FlameAQ Is NC Contact (1=NoFlame)
Cfg_FlameBQNC	BOOL	Inp_FlameBQ is NC Contact (1=NoFlame)
Cfg_FlameIndep	BOOL	Main Fuel Flame Sensor Independent from Pilot Flame Sensor
Cfg_FlameOffDelay	REAL	Flame Sensor Off Debounce Delay (seconds)
Cfg_FlameOnDelay	REAL	Flame Sensor On Debounce Delay (seconds)

raS_Opr_BMS_Fuel AOI Configuration Parameters

Parameter Name	Data Type	Function
Cfg_IgnitionTime	REAL	Light-Off Pilot Active Time (seconds)
Cfg_TFITime	REAL	Proof of Ignition Delay Time (seconds)
Cfg_StabilizeTime	REAL	Flame Stabilization Time (seconds)
Cfg_HasTSL	BOOL	Fuel Low Temperature Switch Exists
Cfg_HasTSH	BOOL	Fuel High Temperature Switch Exists
Cfg_SingleFuelBlkVlv	BOOL	Single Fuel Blocking Valve. (Ref_FuelBlkVlvA Only)
Cfg_HasFuelAuxVlv	BOOL	Fuel Train Auxiliary (Vent or Recirc) Valve Exists
Cfg_FuelAuxVlvFirst	BOOL	Energized Fuel Auxiliary Valve Before Blocking Valve(s)
Cfg_HasAtomizing	BOOL	Fuel Requires Atomizing Media
Cfg_AtomizerDelay	REAL	Delay Before Activating Atomizer Flow Interlock (seconds)
Cfg_HasFuelScavenge	BOOL	Fuel Requires Post-Firing Scavenging
Cfg_FuelScavengeTime	REAL	Post-Firing Fuel Scavenge Time (seconds)
Cfg_HasVP	BOOL	Gas Fuel Train with Valve Proving
Cfg_VPPostFiring	BOOL	Perform Valve Proving Post-Firing
Cfg_VPPrepTime	REAL	Valve Proving Evacuate and Pressurize Time (seconds)
Cfg_VPLoTestTime	REAL	Valve Proving Low Pressure Test Time (seconds)
Cfg_VPHiTestTime	REAL	Valve Proving High Pressure Test Time (seconds)
Cfg_HasPerm01	BOOL	User Permissive Input 1 In Use
Cfg_HasPerm02	BOOL	User Permissive Input 2 In Use
Cfg_HasPerm03	BOOL	User Permissive Input 3 In Use
Cfg_HasPerm04	BOOL	User Permissive Input 4 In Use
Cfg_HasIntlk01	BOOL	User Interlock Input 1 In Use
Cfg_HasIntlk02	BOOL	User Interlock Input 2 In Use

Input Parameters

raS_Opr_BMS_Fuel AOI Input Parameters

Parameter Name	Data Type	Function
Inp_Enable	BOOL	Enable Fuel Control AOI
Inp_FlameA	BOOL	Flame Sensor A Input Normally Open (1=Flame present)
Inp_FlameAQ	BOOL	Flame Sensor A OK or Normally Closed
Inp_FlameStrengthA	REAL	Flame Sensor A Strength Value
Inp_FlameB	BOOL	Flame Sensor B Input Normally Open (1=Flame present)
Inp_FlameBQ	BOOL	Flame Sensor B OK or Normally Closed
Inp_FlameStrengthB	REAL	Flame Sensor B Strength Value
Inp_PSL	BOOL	Fuel Low Pressure Switch
Inp_PSH	BOOL	Fuel High Pressure Switch
Inp_TSL	BOOL	Heated Fuel Oil Low Temperature Switch
Inp_TSH	BOOL	Heated Fuel Oil High Temperature Switch
Inp_VPSL	BOOL	Valve Proving Low Pressure Threshold Switch
Inp_VPSH	BOOL	Valve Proving High Pressure Threshold Switch
Inp_AtLightOffPosn	BOOL	Fuel Actuator at Light-Off Position
Inp_AtomizerPS	BOOL	Atomizing Media Supply Pressure Switch
Inp_AtomizerFS	BOOL	Atomizing Media Flow Confirmation
Inp_BlkVlvAPoC	BOOL	Fuel Blocking Valve A Proof-of-Closure Switch
Inp_BlkVlvBPoC	BOOL	Fuel Blocking Valve B Proof-of-Closure Switch
Inp_Perm01	BOOL	User Permissive Input 1
Inp_Perm02	BOOL	User Permissive Input 2
Inp_Perm03	BOOL	User Permissive Input 3
Inp_Perm04	BOOL	User Permissive Input 4
Inp_Intlk01	BOOL	User Interlock Input 1
Inp_Intlk02	BOOL	User Interlock Input 2

Control Output Parameters

raS_Opr_BMS_Fuel AOI Control Output Parameters

Parameter Name	Data Type	Function
Out_LightOff	BOOL	Fuel Actuator to Light-Off Position
Out_Hold	BOOL	Fuel Actuator Hold at Current Position
Out_Release	BOOL	Fuel Actuator Released for Modulation
Val_eState	DINT	Fuel Control State Engine Enumeration Value
Val_TimeRemaining	REAL	Current Activity Time Remaining (seconds)
Val_FlameA	REAL	Flame Sensor A Strength Value
Val_FlameB	REAL	Flame Sensor B Strength Value

Status and Information Output Parameters

:raS_Opr_BMS_Fuel AOI Status and Information Output Parameters

Parameter Name	Data Type	Function
Sts_InitDN	BOOL	First Scan Initialization Complete
Sts_Enabled	BOOL	Fuel Control Enabled
Sts_ER	BOOL	Configuration Error Operation is Disabled
Sts_bCfgErr	DINT	Configuration Error Status (bit-mapped)
Sts_Faulted	BOOL	Operational Fault Exists
Sts_bFaults	DINT	Fault Status Word (bit-mapped)
Sts_PermsOK	BOOL	Fuel Sequence Permissives OK
Sts_bPerms	DINT	Permissives Lost Status Word (bit-mapped)
Sts_IntlksOK	BOOL	All Interlocks OK
Sts_bIntlks	DINT	Interlocks Lost Status Word (bit-mapped)
Sts_SafetyTrip	BOOL	Interlock Safety Trip Reset Required
Sts_bSafetyTrip	DINT	Safety Trip First Out Capture (bit-mapped)
Sts_Ready	BOOL	Ready To Operate
Sts_Active	BOOL	Fuel Control is Active
Sts_Normal	BOOL	Normal Operating Mode
Sts_Test	BOOL	Maintenance Test Mode Enabled
Sts_TestActive	BOOL	Fuel Device Maintenance Testing is Active
Sts_FuelPoC	BOOL	Fuel Blocking Valves Confirmed Closed
Sts_FuelOn	BOOL	Fuel to Burner Enabled
Sts_Release	BOOL	Fuel Lit and Flame Stable or Fuel Header Ready for Main Fuel Supply
Sts_FlamePresent	BOOL	Flame Presence Sensed
Sts_FlameA	BOOL	Flame Sensed at Detector A
Sts_FlameB	BOOL	Flame Sensed at Detector B
Sts_FuelAtLightOffPosn	BOOL	Fuel Actuator At Light-Off Position
Sts_FuelScavenging	BOOL	Fuel Scavenging Active
Sts_VPActive	BOOL	Valve Proving Sequence Active
Sts_VPOK	BOOL	Valve Proving Test Successful
Sts_VPReqd	BOOL	Valve Proving Test Required Prior to Next Burner Light-Off
Sts_VPFailure	BOOL	Valve Proving Test Unsuccessful
Sts_FuelBlkVlvA	BOOL	Upstream Fuel Blocking Valve Energized
Sts_FuelBlkVlvB	BOOL	Downstream Blocking Valve Energized
Sts_FuelAuxVlv	BOOL	Auxiliiary Fuel Valve (Vent or Recirc) Energized
Sts_Atomizer	BOOL	Atomizing Media Valve Energized
Sts_ScavengeVlv	BOOL	Fuel Scavenging Valve Energized
Sts_FuelHeader	BOOL	Fuel Header Control
Sts_FuelGas	BOOL	Gasseous Fuel Selected

:raS_Opr_BMS_Fuel AOI Status and Information Output Parameters

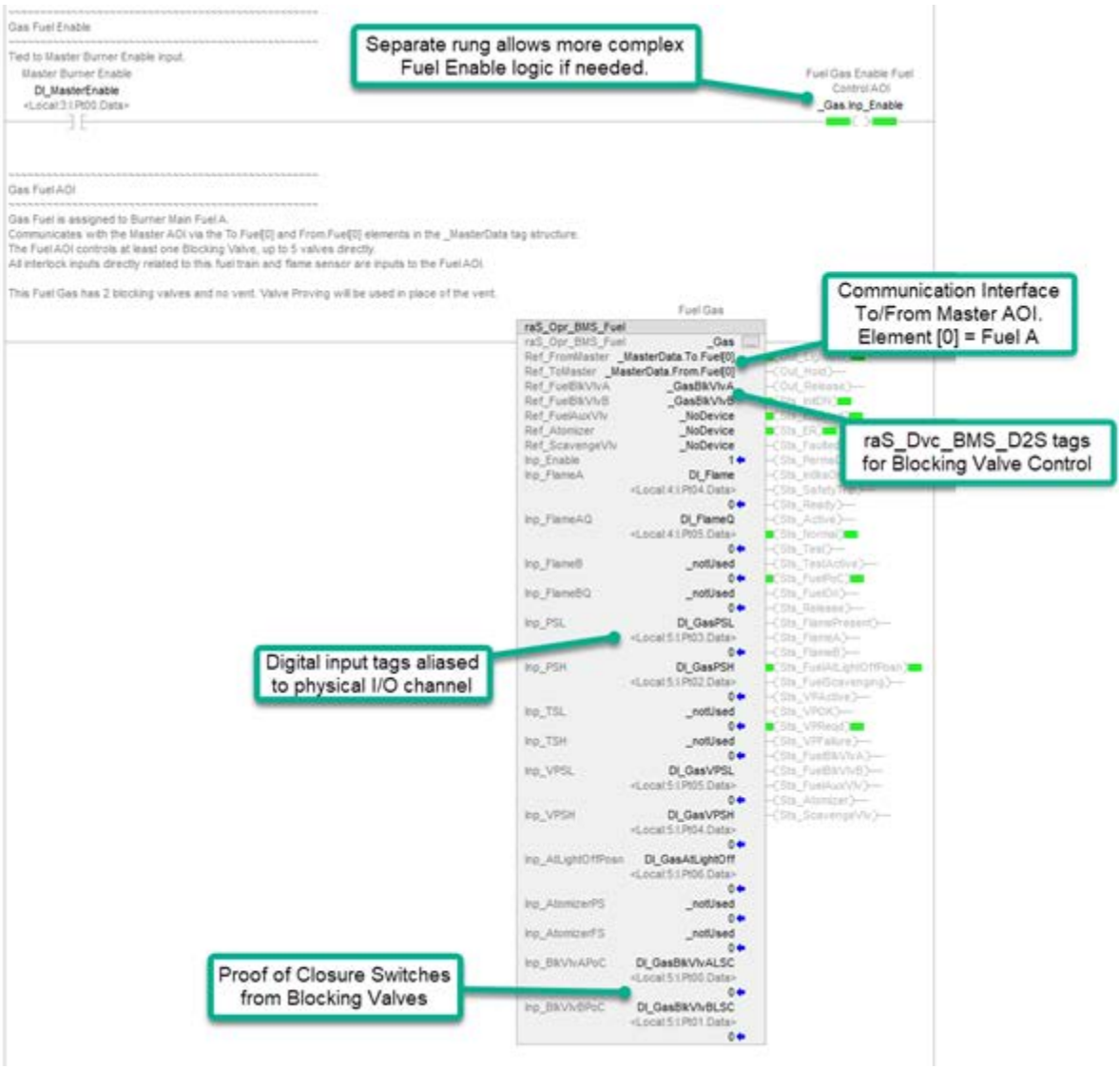
Parameter Name	Data Type	Function
Sts_FuelOilLt	BOOL	Light Fuel Oil (#2, #4) Selected
Sts_FuelOilHvy	BOOL	Heavy Fuel Oil (#5, #6) Selected
Rdy_Reset	BOOL	Fault or Safety Trip Reset Required
Inf_Label	BOOL	Fuel Control (@Description Used by HMI)
Inf_Tag	BOOL	Tag ID (@Description Used by HMI)
Inf_BuildID	DINT	raS_Opr_BMS_Fuel AOI Build ID

Programming Example

General arrangement for this example is shown in the Figure 1 - Example Task, Program, and Routine Arrangement. The raS_Opr_BMS_Fuel, being a Safety AOI, is located in the BMS_Example program, d_Gas and d_Oil routines. The rungs that follow are taken from that routine:

- Rung 1: Fuel Enable logic, tied to same input as Master AOI, shown here.
- Rung 2: raS_Opr_BMS_Fuel AOI for Fuel Gas, shown here.
- Rung 3: Rung for Blocking Valve A control, example in raS_Dvc_BMS_D2S AOI section.
- Rung 4: Rung for Blocking Valve B control, example in raS_Dvc_BMS_D2S AOI section.
-

Fuel AOI Example



Graphic Symbols

Graphic Symbols are used as launch buttons within HMI applications to open up faceplate displays. See Basic Launch Button Attributes section for details on configuration and indicators.

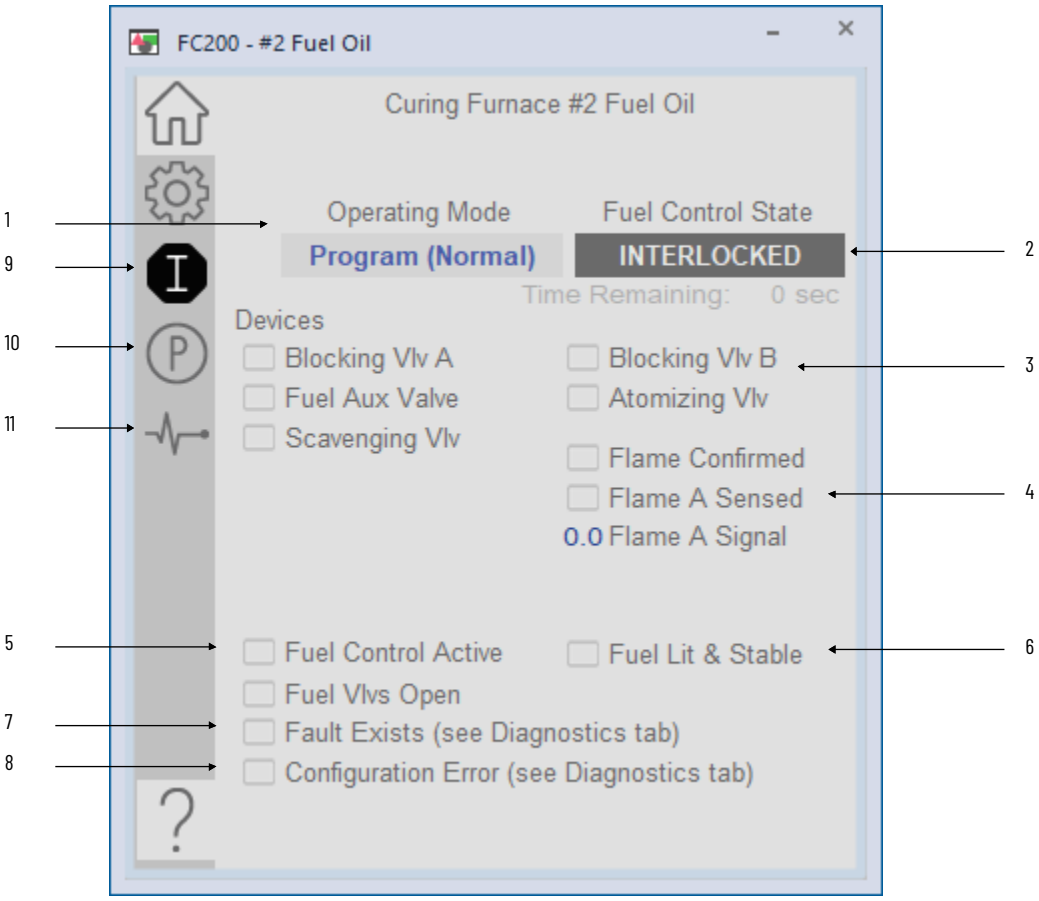
FactoryTalk® View SE Graphic Symbols

Graphic Symbol Name	Graphic Symbol	Description	Global Object Parameter Values
GO_FuelStatus		Fuel Status with faceplate navigation. This launch button graphic object allows the user to navigate to the object faceplate. The text on the button face is set to the tag's .Inf_Label.@Description field. Also shown are the operating state and Fuel time to next external watchdog relay test.	#102: AOI Instance (e.g. {::[PAC]Program::SafetyProgram._Instance Name}) #103: HMI Control Tag #120: Faceplate X Position, pixels from left edge of display (e.g. /X100) #121: Faceplate Y Position, pixels from top edge of display (e.g. Y100)
GO_FuelStatusNoNav		Fuel Status without navigation. . The text on the object face is set to the tag's .Inf_Label.@Description field. Also shown are the operating state and Fuel time to next external watchdog relay test.	#102: AOI Instance (e.g. {::[PAC]Program::SafetyProgram._Instance Name})

Faceplates

There are basic faceplate attributes that are common across all instructions. See Basic Faceplate Attributes on page 21.

Home



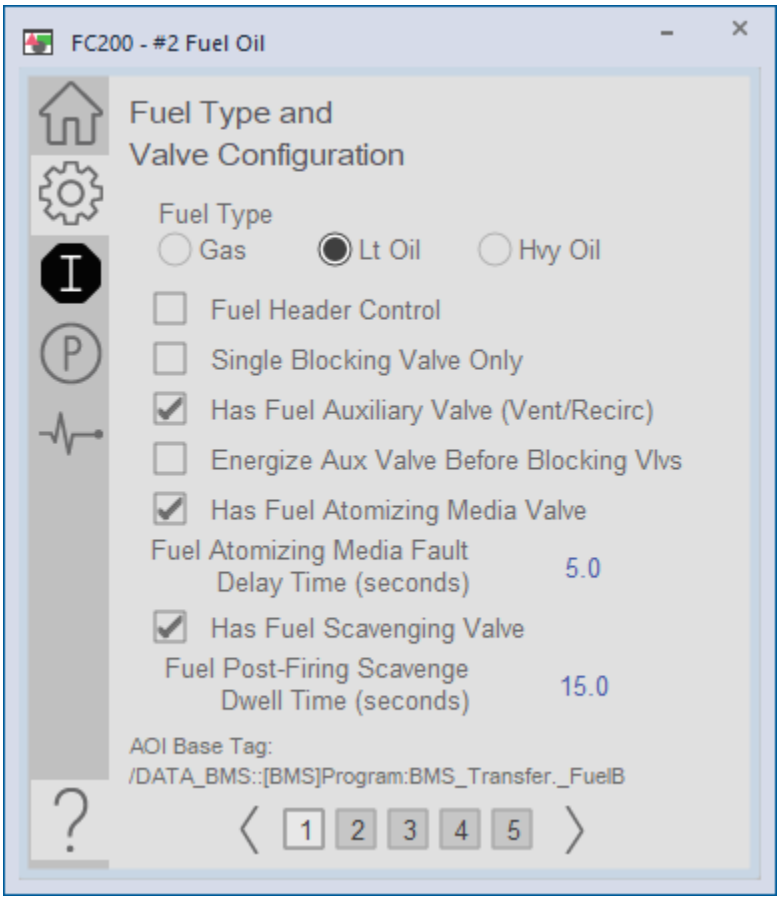
Item	Description
1	Fuel Operating Mode indicator.
2	Fuel Operating State indicator.
3	Fuel Device Energized status indicators.

4	Fuel Flame Status summary.
5	Fuel Control and Blocking Valve status indicators.
6	Fuel Lit and Stable (ready for modulation) status indicator.
7	Fuel Fault Status Summary indicator.
8	Fuel Configuration status summary indicator.
9	Fuel Interlock status summary indicator, showing at least one Interlock condition NOT OK.
10	Fuel Permissive status summary indicator, showing all Permissive conditions OK.
11	Fuel Diagnostic status summary indicator, showing all Diagnostic conditions OK.

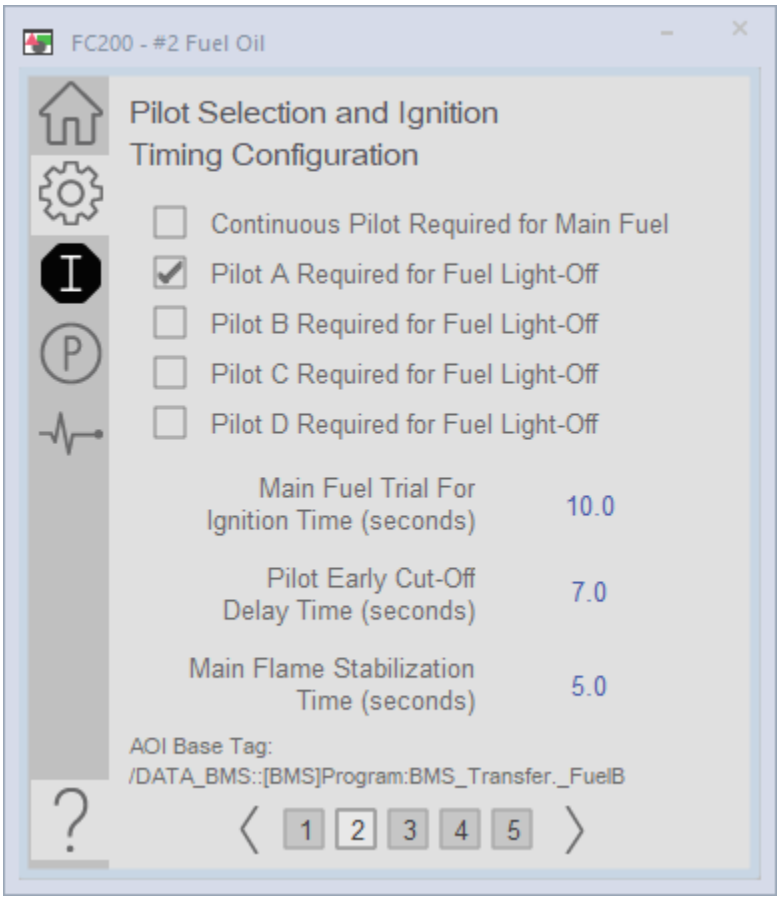
Configuration

There are five pages within the configuration tab.

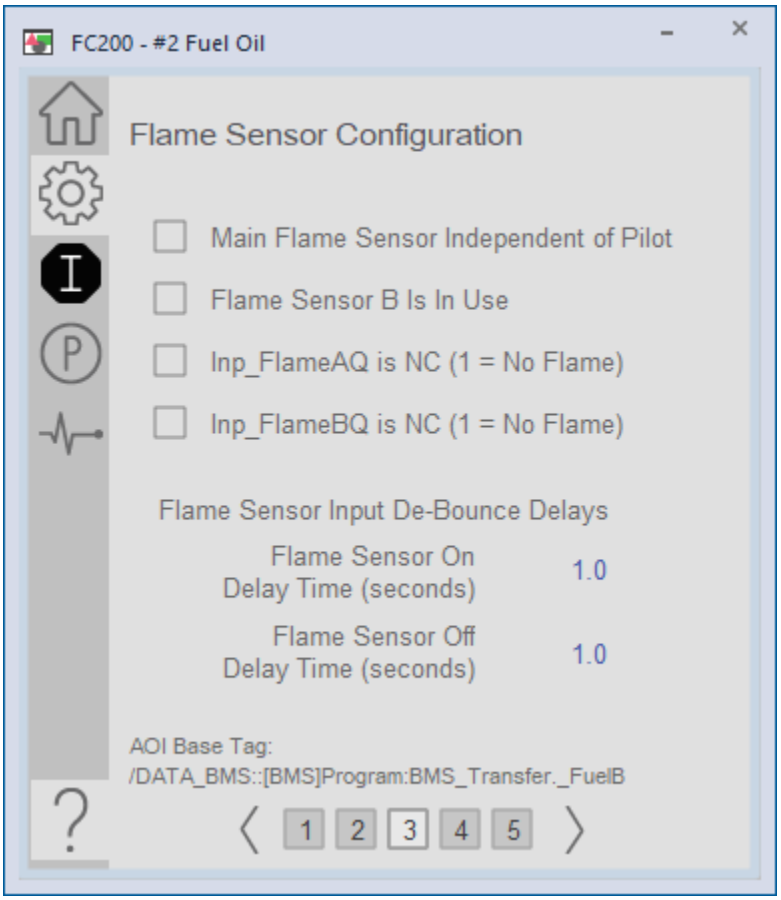
Configuration Page 1 - Fuel Type and Valve Configuration



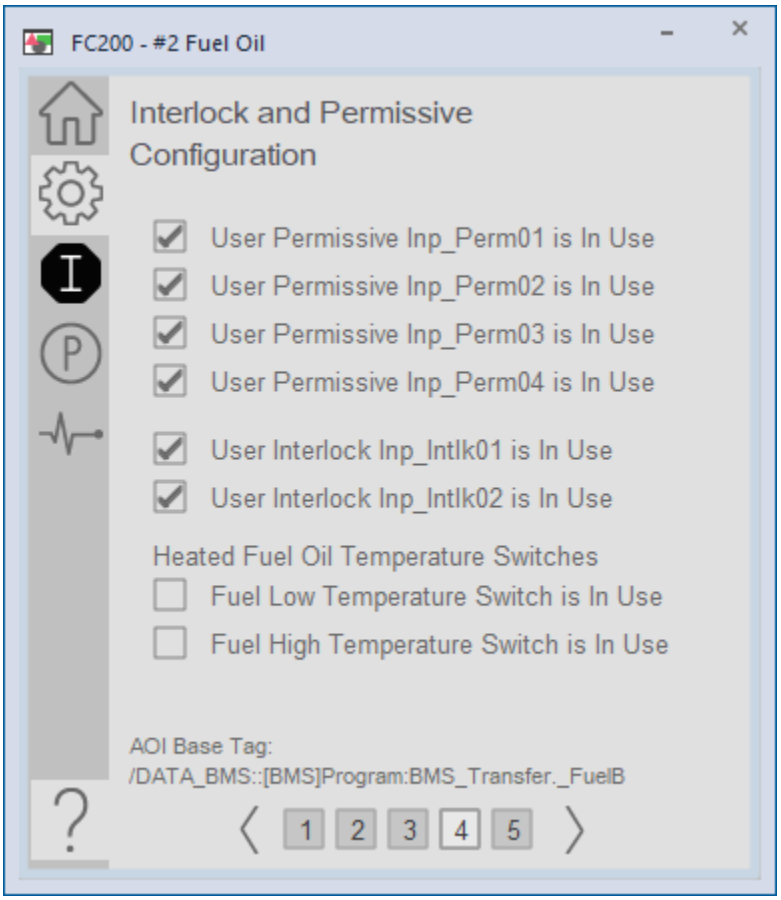
Configuration Page 2 - Pilot Slection and Ignition Timing Configuration



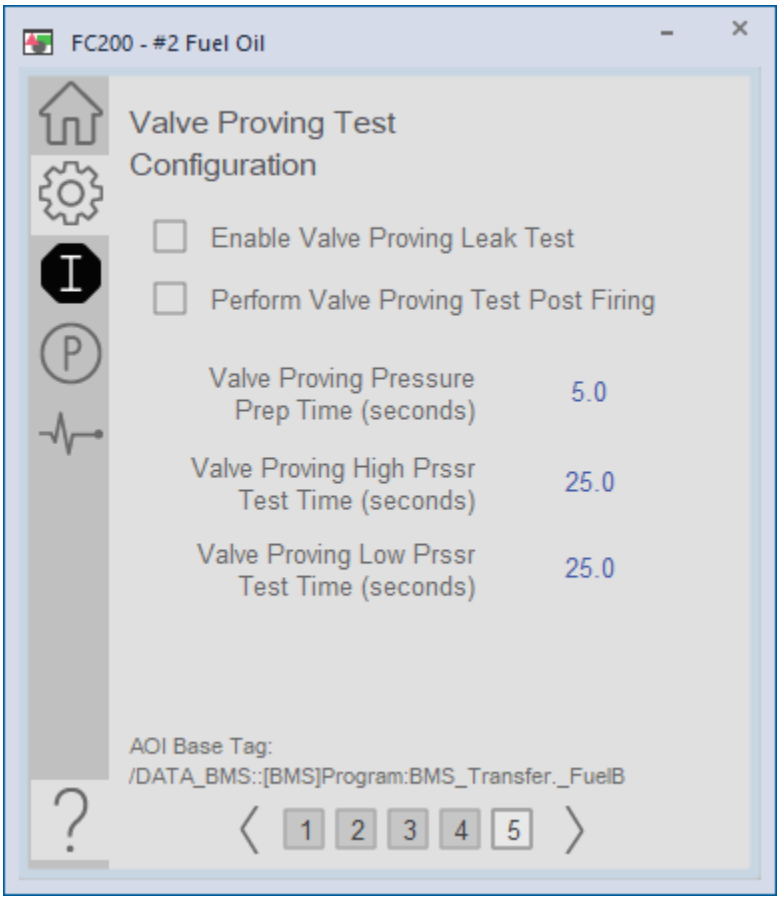
Configuration Page 3 - Flame Sensor Configuration



Configuration Page 4 - Interlock and Permissive Configuration



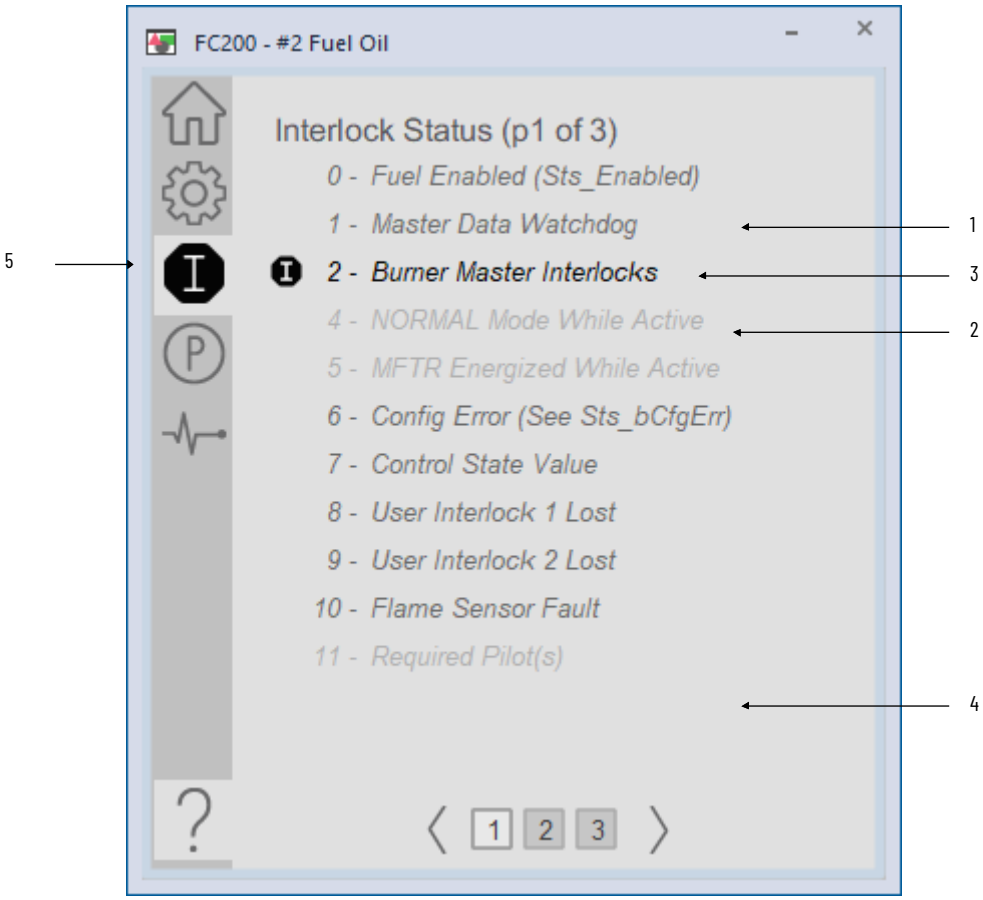
Configuration Page 5 - Valve Proving Test Configuration



Interlock Status

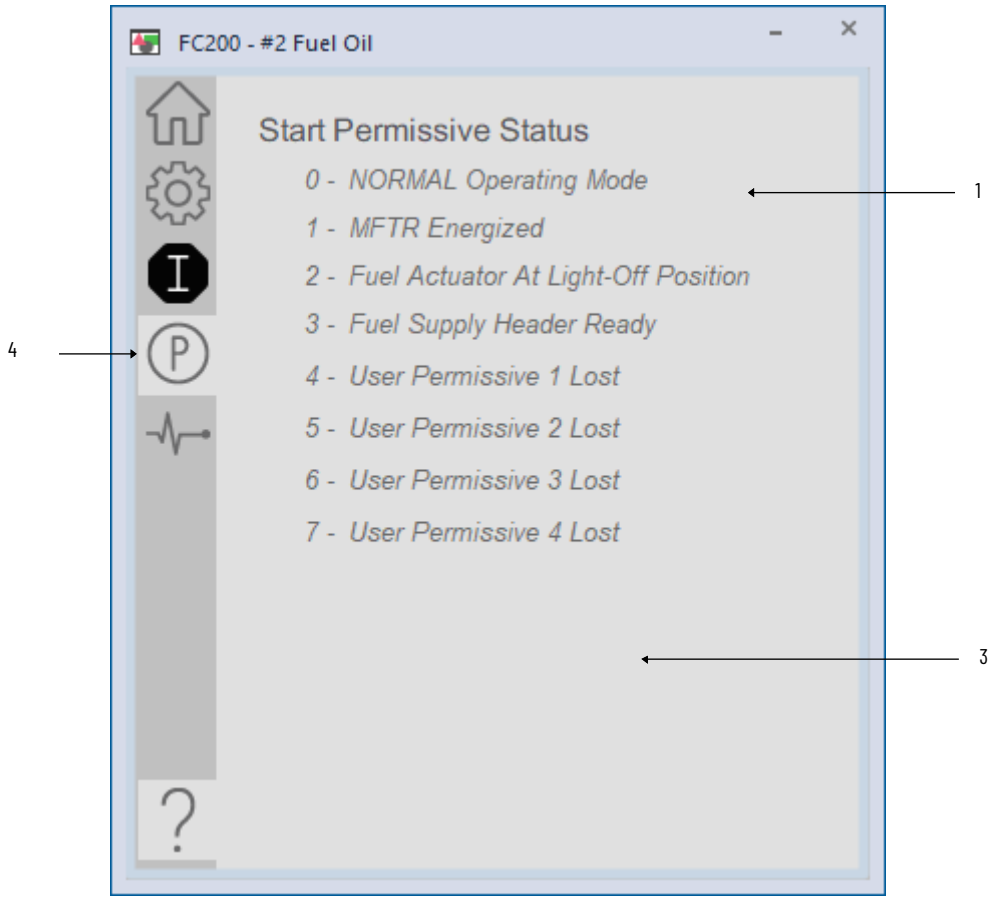
There are 3 pages within the Interlock Status tab - each with 12 interlocks.

Interlock Status Page 1 - Interlocks 0-11



Item	Description
1	Interlock currently monitored, but OK to operate.
2	Interlock configured for use, but currently not monitored.
3	Interlock currently monitored, but NOT OK to operate.
4	Interlock not configured for use (User Defined Interlock Input 1 in this case).
5	Summary indicator – at least one Interlock is NOT OK.

Permissive Status

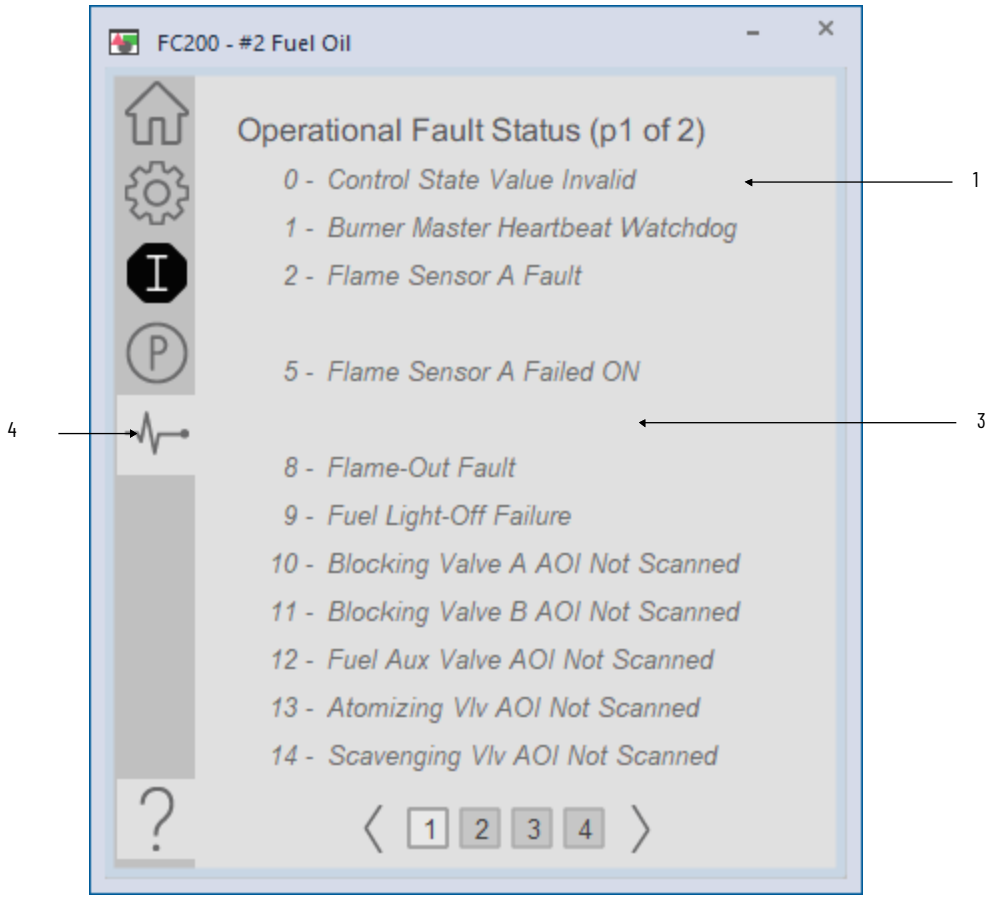


Item	Description
1	Permissive currently monitored, but OK to Start Operation.
3	Permissive not configured for use (User Permissive Input 1 in this case).
4	Summary indicator – at least one Permissive condition is NOT OK.

Diagnostics

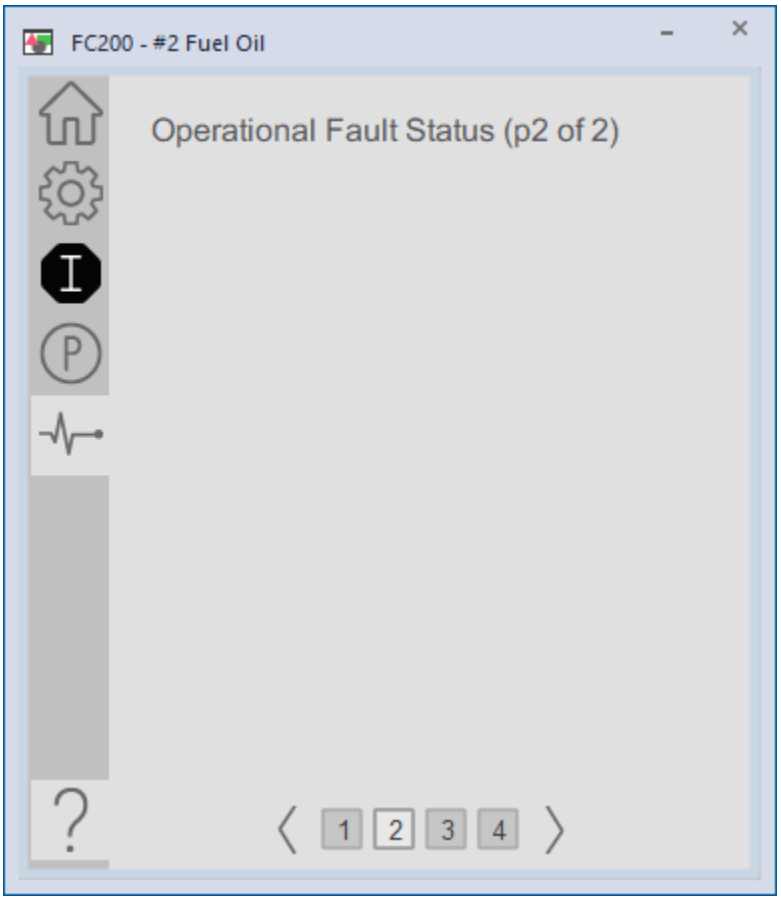
There are 4pages within the diagnostics tab.

Diagnostics Page 1 - Operational Fault Status (1 of 2)

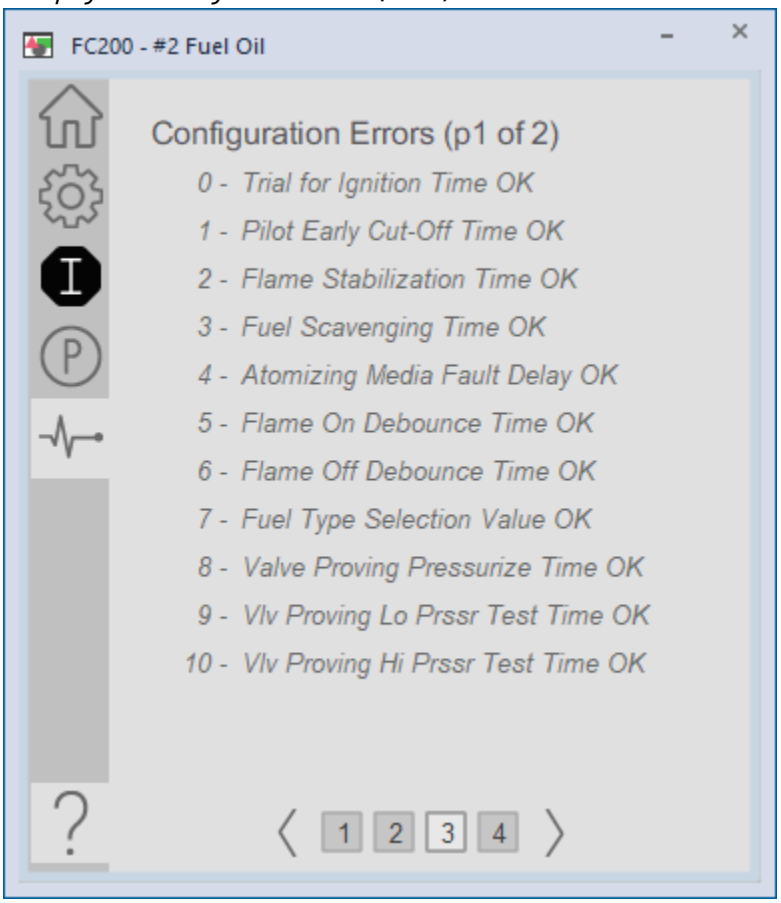


Item	Description
1	Operational Fault currently monitored, and OK to Operate.
3	Operational Fault not configured for use
4	Summary indicator – at least one Diagnostic condition is NOT OK.

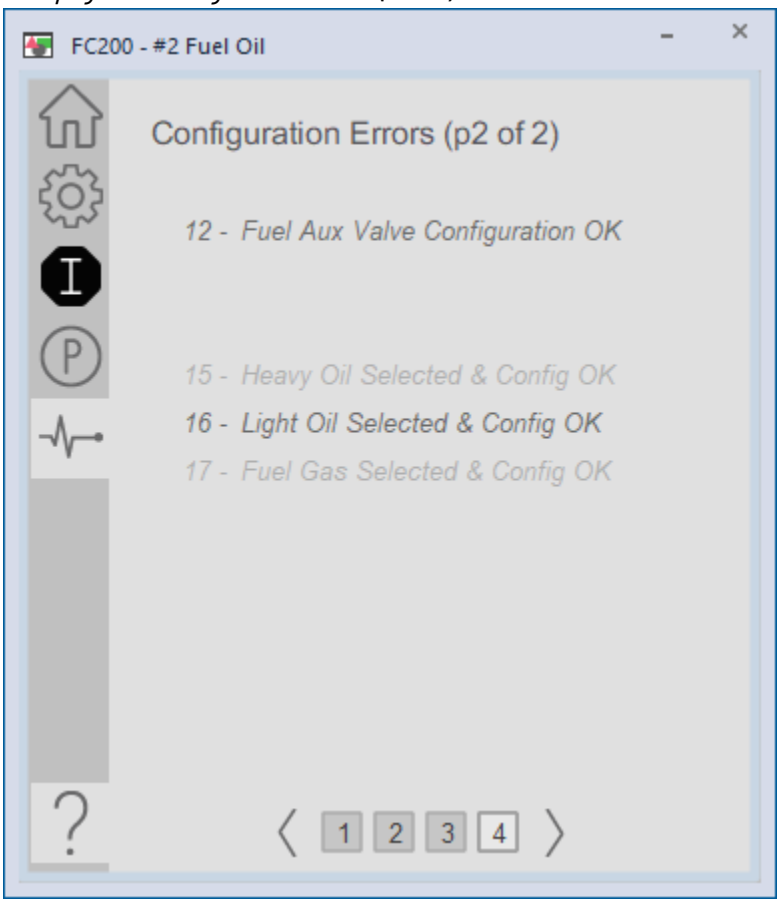
Diagnostics Page 2 - Operational Fault Status (2 of 2)



Diagnostics page 3 – Configuration Errors (1 of 2)



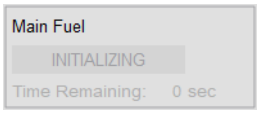
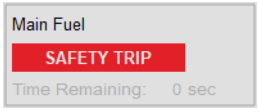
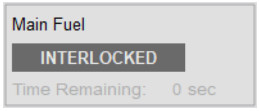
Diagnostics page 4 – Configuration Errors (2 of 2)

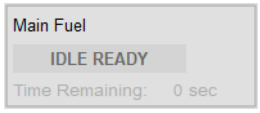
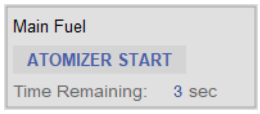
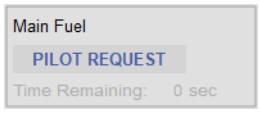
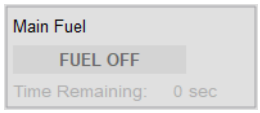
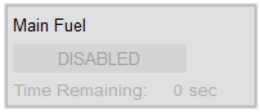


State Value Enumerations

Main Fuel State Enumerations

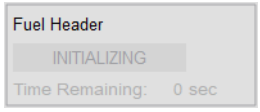
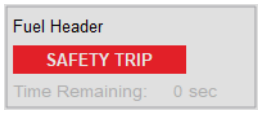
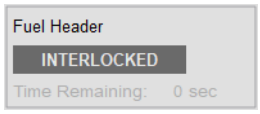
The following table identifies the states available to the Main Fuel when enabled (Sts_Enabled is True). If Sts_Enabled is False, the Main Fuel is allowed to only be in States 0, 1, and 2, and will indicate DISABLED.

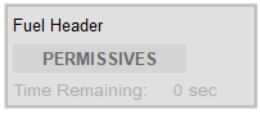
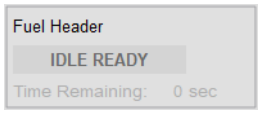
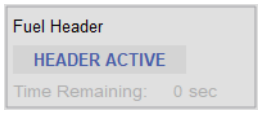
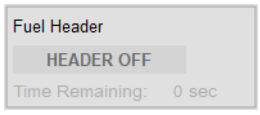
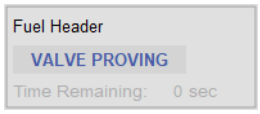
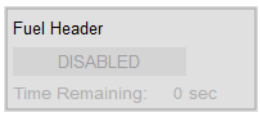
Val_eState	State Name	Description	HMI Graphic
0	Initializing	AOI performing 1st scan initialization.	
1	Safety Trip	Fuel Safety Trip present.	
2	Interlocked	Fuel Interlock not met.	
3	Not Ready	Fuel waiting for all Permissives OK.	

4	Idle / Ready	Fuel Idle and Ready for Start Command.	
5	Atomizer Prep	Atomizing Media Valve Energize and confirm media flow.	
6	Pilot Request	Request pilot for ignition.	
7	Fuel Ignition	Main Fuel Trial for Ignition.	
8	Flame Confirmed	Flame Presence confirmed.	
9	Fuel Scavenge	Timed fuel scavenging activity.	
10	Flame Off	Main fuel off, confirm loss of flame.	
11	Valve Proving	Blocking valve leak test active.	
0, 1, or 2		Fuel Disabled (Sts_Enabled = 0)	

Fuel Header State Enumerations

The following table identifies the states available to the Fuel Header when enabled (Sts_Enabled is True). If Sts_Enabled is False, the Fuel Header is allowed to only be in States 0, 1, and 2, and will indicate DISABLED.

Val_eState	State Name	Description	HMI Graphic
0	Initializing	AOI performing 1st scan initialization. (Enabled: Sts_Enabled = 1)	
1	Safety Trip	Fuel Header Safety Trip present. (Enabled: Sts_Enabled = 1)	
2	Interlocked	Fuel Header Interlock not met. (Enabled: Sts_Enabled = 1)	

3	Not Ready	Fuel Header waiting for all Permissives OK.	
4	Idle / Ready	Fuel Header Idle and Ready for Start Command.	
5	n/a	State not valid for Fuel Header	
6	n/a	State not valid for Fuel Header	
7	n/a	State not valid for Fuel Header	
8	Header Active	Header Ready to supply fuel to burner(s).	
9	n/a	State not valid for Fuel Header	
10	Deactivate	Stop supply of fuel to burner(s).	
11	Valve Proving	Blocking valve leak test active.	
0, 1, or 2		Header Disabled (Sts_Enabled = 0)	

BMS 2-State Device (raS_Dvc_BMS_D2S)

Overview

Common device control Add-On Instruction (AOI) for spring-return valves, control relays, spark ignitors, or any other On-Off device where safe state is deenergized. Provides a single digital control output that is true to energize the physical device and false to deenergize the physical device. Configurable feedback inputs can be implemented to confirm energized and deenergized positions.

Two timers are provided for each position, one timer monitoring immediate action (stall timer) to confirm the valve begins its commanded movement within a specified time, the other being a transition timer to confirm the device completes its commanded movement within a specified time. If a feedback device is configured, the timers trigger faults for that direction of travel. For example, if the Deenergized position feedback exists, the Energized Stall timer will generate a stall fault if the device is energized and the Deenergized feedback is not cleared before the timer expires. Likewise, the Deenergized transition timer will generate a fault if the device is deenergized and the feedback is not made before the transition timer expires.

The raS_Dvc_BMS_D2S AOI inputs and outputs are single Boolean parameters, but additional safety instructions can be implemented externally to impart dual channel reliability where the application requires.

The Device raS_Dvc_BMS_D2S AOI is tied to its host control AOI (Master, Pilot, or Fuel) via InOut reference. All commands from and status back to the host component use this InOut reference, eliminating the need for external control logic. The raM_Dvc_BMSD2S AOI does need to be executed on its own rung, however.

Limitations

The following limitations must be considered when implementing this instruction within a BMS Library application.

AOI Class	Safety
Minimum Instances Required	3
Maximum Instances Allowed	As required, up to 56 per burner system

System Interaction

Each Device AOI exchanges data with its Host component using selected parameters of its AOI base tag. The parameters used for this purpose are listed below. This interaction is created by embedding the raS_Dvc_BMS_D2S AOI Base Tag in the Host component as an InOut parameter. Each Device controlled by a Host component has one InOut parameter for each Device controlled.

Device to Host Data

Data Type: raS_Dvc_BMS_D2S (Device AOI Base Tag), one tag for each Device controlled by a Host component.

Device to Host Parameter Reference: Ref_<device> (InOut)

Device to Host Interface Parameters

Element Name	Data Type	Description
Ref_<device>	raS_Dvc_BMS_D2S	Device Control AOI Base Tag
Ref_<device>.Sts_Enabled	BOOL	Device Enabled
Ref_<device>.Sts_Exists	BOOL	Device Exists Watchdog Handshake
Ref_<device>.Sts_IntlksOK	BOOL	Device Interlocks OK
Ref_<device>.Sts_Normal	BOOL	Device in NORMAL MODE
Ref_<device>.Sts_Test	BOOL	Device in TEST MODE
Ref_<device>.Sts_TestActive	BOOL	Device in TEST MODE with Device Test Active
Ref_<device>.Sts_DeEnergized	BOOL	Device Confirmed in De-Energized Position
Ref_<device>.Sts_CmdEnergize	BOOL	Device Commanded to Energized Position
Ref_<device>.Sts_Energized	BOOL	Device Confirmed in De-Energized Position
Ref_<device>.Sts_CmdDeEnergize	BOOL	Device Commanded to De-Energized Position
Ref_<device>.Rdy_Reset	BOOL	Device Needs Fault/Safety Trip Reset
Ref_<device>.Inf_BuildID	DINT	raS_Dvc_BMS_D2S AOI Build ID

Host to Device

Data Type: raS_Dvc_BMS_D2S (Device AOI Base Tag), one tag for each Device controlled by a Host component.

Host to Device Parameter Reference: Ref_<device> (InOut)

Host to Device Interface Parameters

Element Name	Data Type	Description
Ref_<device>	raS_Dvc_BMS_D2S	Device Control AOI Base Tag
Ref_<device>.Inp_Enable	BOOL	Device Enable Input from Host
Ref_<device>.Inp_IntlksOK	BOOL	External Interlocks OK
Ref_<device>.Inp_TestEnable	BOOL	TEST ENABLE Mode input
Ref_<device>.PCmd_DeEnergize	BOOL	NORMAL Mode De-Energize request
Ref_<device>.PCmd_Energize	BOOL	NORMAL Mode Energize request
Ref_<device>.PCmd_Reset	BOOL	NORMAL Mode Device Reset request
Ref_<device>.Sts_Exists	BOOL	Device Exists Watchdog Handshake

Device AOI Configuration

The Device AOI configuration includes feedback selection and timing values. Position feedback for energized and de-energized positions can be selected for use. Time delays for transition and stall failures are also configurable. Configuration parameter details are shown in the following sections.

Interlocks

The raS_Dvc_BMS_D2S AOI monitors 14 interlock conditions, none of which are user selectable. The following table identifies each interlock condition, when each is monitored, and when each will trigger a Safety Trip.

Device AOI Interlock List

Interlock Condition	Interlock Enable	Safety Trip Enable
Device Disabled	Always	Device Active
External Interlocks OK	Device Enabled	Device Active
Device Stalled in DeEnergized Position	Device Enabled and Has DeEnergized Feedback	Device Active and Has DeEnergized Feedback
Device Failed to Reach DeEnergized Position	Device Enabled and Has DeEnergized Feedback	Device Active and Has DeEnergized Feedback
Device Lost DeEnergized Position Feedback	Device Enabled and Has DeEnergized Feedback	Device Active and Has DeEnergized Feedback
Device Stalled in Energized Position	Device Enabled and Has Energized Feedback	Device Active and Has Energized Feedback
Device Failed to Reach Energized Position	Device Enabled and Has Energized Feedback	Device Active and Has Energized Feedback
Device Lost Energized Position Feedback	Device Enabled and Has Energized Feedback	Device Active and Has Energized Feedback
Device Feedback Conflict	Device Enabled and Has Energized and DeEnergized Feedback	Device Active and Has Energized and DeEnergized Feedback
Invalid State Value	Device Enabled	Device Active
Invalid DeEnergized Stall Time	Device Enabled	Device Active
Invalid DeEnergize Transition Time	Device Enabled	Device Active
Invalid Energized Stall Time	Device Enabled	Device Active
Invalid Energize Transition Time	Device Enabled	Device Active

Permissives

The raS_Dvc_BMS_D2S AOI does not monitor any conditions as permissives.

Operations

Operation Modes

The BMS Library objects provide only physical operation mode. There is no virtual/simulated mode offered.

Normal Operation Mode

The Device must be in NORMAL Mode to respond to commands from the Host. While in NORMAL Mode, the Device does not respond to any requests issued via the MCmd input. Clearing Inp_TestEnable places the Device in NORMAL Mode.

NORMAL Mode status is echoed back to the Host indicating whether the Device is in NORMAL Mode or not.

Test Enable Mode Operation

As part of the Host-Device Interface, TEST ENABLE Mode is dictated and controlled by the Host component setting Inp_TestEnable. When Inp_TestEnable is set, MCmd.0 input (Acquire Test Active Request) is used to activate Device Test Active.

If Inp_TestEnable is cleared while the Device is Test Active, Device Test Active will be cancelled and the device will be de-energized if energized. The Device will then return to NORMAL Mode in a de-energized state.

TEST ENABLE Mode status is echoed back to the Host indicating the Device is in TEST ENABLE Mode.

Device Test Active

Device Test Active indicates the Device is actively being tested and can be energized and de-energized based on commands received via the MCmd input. The Device must be maintained in TEST ENABLE Mode while in Device Test Active. Removal of the TEST ENABLE Mode input (Inp_TestEnable) by the Host aborts Device Test Active and de-energizes the Device if energized. Maintenance Operator can relinquish Device Test Active by setting MCmd.1 (Relinquish Test Active Request). Relinquishing Test Active Mode will de-energize the device if energized.

Required Files

Device Objects include Add-On Instructions (AOIs) and HMI faceplates. The revision number (e.g. 1.01) used in filenames can change as new revisions are created.

Controller Files

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

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

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

Device/Item	Add-On Instruction	RUNG Import
D2S	raS_Dvc_BMS_D2S_1.00_AOI.L5X	raS_Dvc_BMS_D2S_1.00_SftyRUNG.L5X
D2S with Output Monitoring		raS_Dvc_BMS_D2S_wOMon_1.00_SftyRUNG.L5X

FactoryTalk® View HMI Files

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

Device/Item	Type	FactoryTalk® View SE Faceplate
BMS D2S Faceplate	Display	(raS-1_00-SE) raS_Dvc_BMS_D2S-Faceplate.gfx
Graphic Symbols	Global Object	(raS-1-SE) Graphic Symbols - BMS.ggfx
Toolbox	Global Object	(raS-1-SE) Toolbox - BMS.ggfx

Add-On Instruction I/O Data

The following tables identify raS_Dvc_BMS_D2S AOI parameters.

Configuration Parameters

raS_Dvc_BMS_D2S AOI Configuration Parameters

Parameter Name	Data Type	Function
Cfg_HasFdbkDeEnergized	BOOL	Device Has and Requires DeEnergized Position Feedback
Cfg_HasFdbkEnergized	BOOL	Device Has and Requires Energized Position Feedback
Cfg_DeEnergizeTime	REAL	Delay Before Transition Fault On DeEnergize
Cfg_DeEnergizedStallTime	REAL	Delay Before Stall Fault On Energize Command
Cfg_EnergizeTime	REAL	Delay Before Transition Fault On Energize
Cfg_EnergizedStallTime	REAL	Delay Before Stall Fault On DeEnergize Command

Input and Command Parameters

raS_Dvc_BMS_D2S AOI Input and Command Parameters

Parameter Name	Data Type	Function
Inp_Enable	BOOL	Device Enable
Inp_DeEnergized	BOOL	DeEnergized Position Confirmation Input
Inp_Energized	BOOL	Energized Position Confirmation Input
Inp_IntlksOK	BOOL	External Interlocks are OK
Inp_TestEnable	BOOL	TEST ENABLE Mode (Enable Maintenance Operator Test Mode Acquisition and Control)
MCmd	SINT	Maintenance Operator Test Command Input Word
MCmd.0	BOOL	Acquire Test Active Request
MCmd.1	BOOL	Relinquish Test Active Request
MCmd.2	BOOL	Test Active Device Reset Request
MCmd.3	BOOL	Test Active Device DeEnergize Request
MCmd.4	BOOL	Test Active Device Energize Request
MCmd.5	BOOL	<reserved>
MCmd.6	BOOL	<reserved>
MCmd.7	BOOL	<reserved>
PCmd_Energize	BOOL	NORMAL Mode (Program/Logic) Command Energize Control Output
PCmd_DeEnergize	BOOL	NORMAL Mode (Program/Logic) Command DeEnergize Control Output
PCmd_Reset	BOOL	NORMAL Mode (Program/Logic) Command Fault Reset

Control Output Parameters

raS_Dvc_BMS_D2S AOI Control Output Parameters

Parameter Name	Data Type	Function
Out_Energize	BOOL	Energize Device Output
Out_CmdAck	BOOL	Test Operator Command Acknowledge
Val_eState	DINT	Device Control State Engine Enumeration Value

Status and Information Output Parameters

raS_Dvc_BMS_D2S AOI Status and Information Output Parameters

Parameter Name	Data Type	Function
Sts_Enabled	BOOL	Device is Enabled for Use
Sts_Exists	BOOL	Device Object Exists and is Being Scanned
Sts_ER	BOOL	Configuration Error Operation is Disabled
Sts_Faulted	BOOL	Device Fault Exists

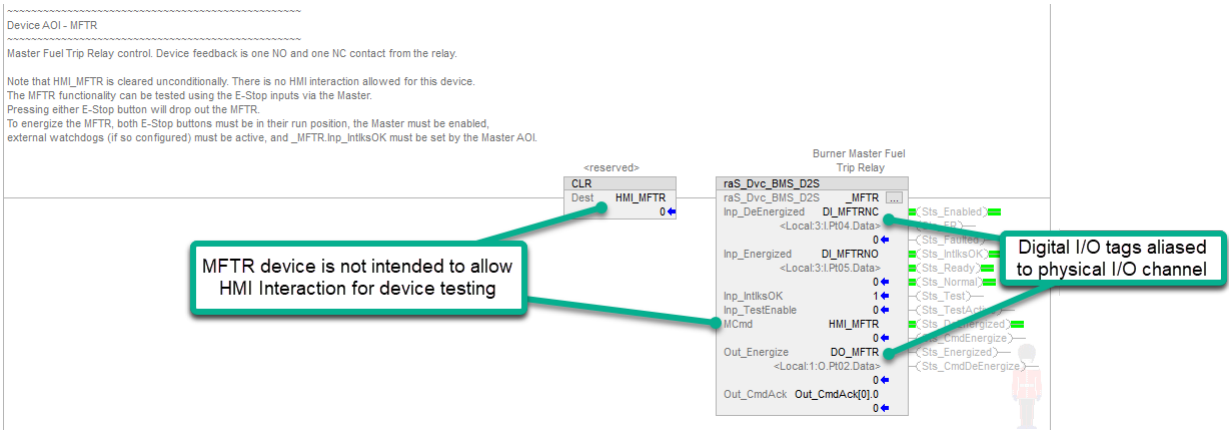
raS_Dvc_BMS_D2S AOI Status and Information Output Parameters

Parameter Name	Data Type	Function
Sts_IntlksOK	BOOL	All Enabled Interlocks OK
Sts_bIntlks	DINT	Interlocks Lost (bit-mapped)
Sts_bIntlksEnab	DINT	Interlocks Enabled (bit-mapped)
Sts_SafetyTrip	BOOL	Interlock Trip Reset Required
Sts_bSafetyTrip	DINT	Safety Trip First-Out Capture (bit-mapped)
Sts_Ready	BOOL	Ready To Energize
Sts_Normal	BOOL	NORMAL Mode (Program/Logic Command Mode)
Sts_Test	BOOL	TEST ENABLE Mode (Maintenance Test Mode Enabled)
Sts_TestActive	BOOL	Device Test Active (Maintenance Test Operator Command Mode)
Sts_DeEnergized	BOOL	DeEnergized Position Confirmed
Sts_CmdEnergize	BOOL	Energize Commanded
Sts_Energized	BOOL	Energized Position Confirmed
Sts_CmdDeEnergize	BOOL	DeEnergize Commanded
Rdy_Reset	BOOL	Fault or Safety Trip Reset Required
Inf_Label	BOOL	2-State Device Control (@Description Used by HMI)
Inf_Tag	BOOL	Tag ID (@Description Used by HMI)
Inf_BuildID	DINT	raS_Dvc_BMS_D2S AOI Build ID

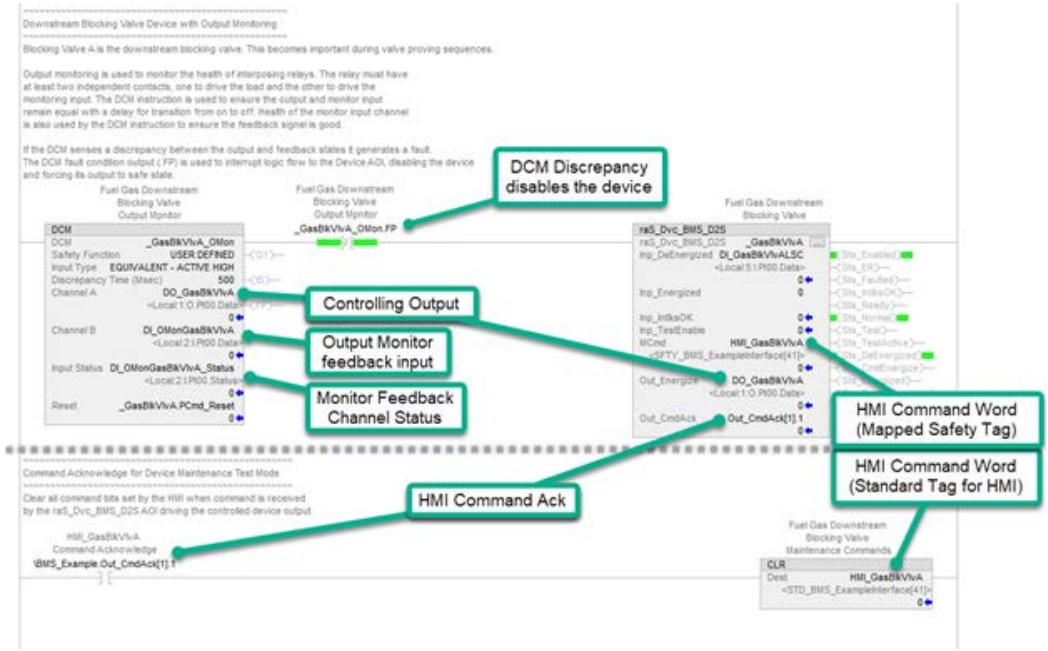
Programming Example

General arrangement for this example is shown in the . The raS_Dvc_BMS_D2S, being a Safety AOI, is located in the BMS_Example program, and used in multiple routines. Each device controlled by the Master, Pilot, or Fuel AOIs will require use of one raS_Dvc_BMS_D2S AOI. Two example rungs are shown below, one driving the physical device directly with a safety output, and the other using an interposing relay with monitor feedback.

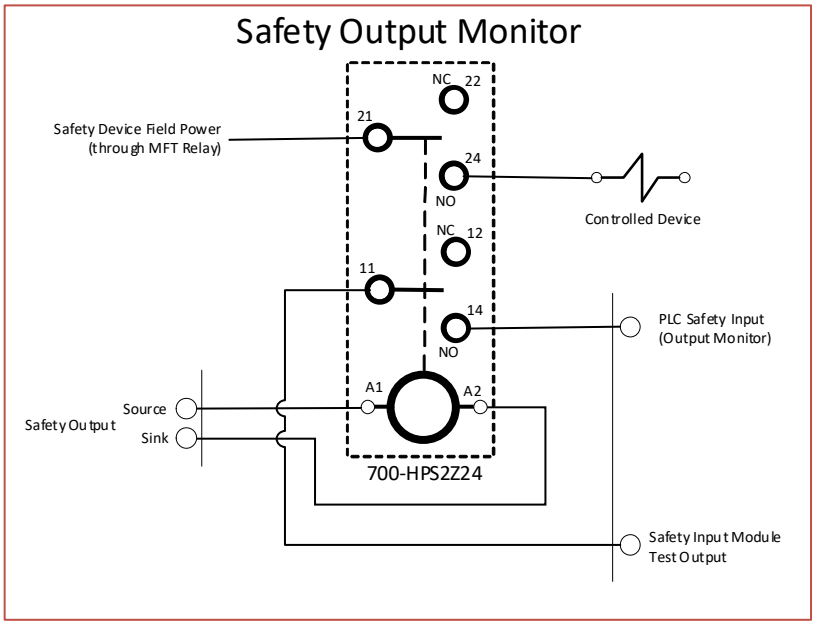
raS_Dvc_BMS_D2S Example Without Monitor Feedback or HMI Interaction



raS_Dvc_BMS_D2S Example with Output Monitor and HMI Interaction



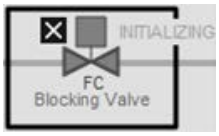
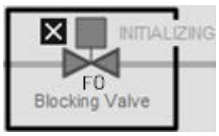
Example Schematic for Output Monitor Feedback

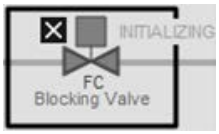
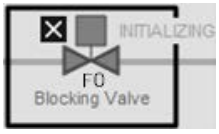


Graphic Symbols

Graphic Symbols are used as launch buttons within HMI applications to open up faceplate displays. See Basic Launch Button Attributes section for details on configuration and indicators.

FactoryTalk® View SE Graphic Symbols

Graphic Symbol Name	Graphic Symbol	Description	Global Object Parameter Values
GO_BMSD2S_Gen		Generic D2S device with faceplate navigation. This launch button graphic object allows the user to navigate to the object faceplate. The text on the button face is set to the tag's .Inf_Label.@Description field.	#102: AOI Instance (e.g. {::[PAC]Program::SafetyProgram._InstanceName}) #103: HMI Control Tag #120: Faceplate X Position, pixels from left edge of display (e.g. /X100) #121: Faceplate Y Position, pixels from top edge of display (e.g. Y100)
GO_BMSD2S_FC		Fail Close D2S device with faceplate navigation. This launch button graphic object allows the user to navigate to the object faceplate. The text on the button face is set to the tag's .Inf_Label.@Description field.	#102: AOI Instance (e.g. {::[PAC]Program::SafetyProgram._InstanceName}) #103: HMI Control Tag #120: Faceplate X Position, pixels from left edge of display (e.g. /X100) #121: Faceplate Y Position, pixels from top edge of display (e.g. Y100)
GO_BMSD2S_FO		Fail Open D2S device with faceplate navigation. This launch button graphic object allows the user to navigate to the object faceplate. The text on the button face is set to the tag's .Inf_Label.@Description field.	#102: AOI Instance (e.g. {::[PAC]Program::SafetyProgram._InstanceName}) #103: HMI Control Tag #120: Faceplate X Position, pixels from left edge of display (e.g. /X100) #121: Faceplate Y Position, pixels from top edge of display (e.g. Y100)
GO_BMSD2S_GenNoNav		Generic D2S device without navigation. The text on the button face is set to the tag's .Inf_Label.@Description field.	#102: AOI Instance (e.g. {::[PAC]Program::SafetyProgram._InstanceName}) #103: HMI Control Tag

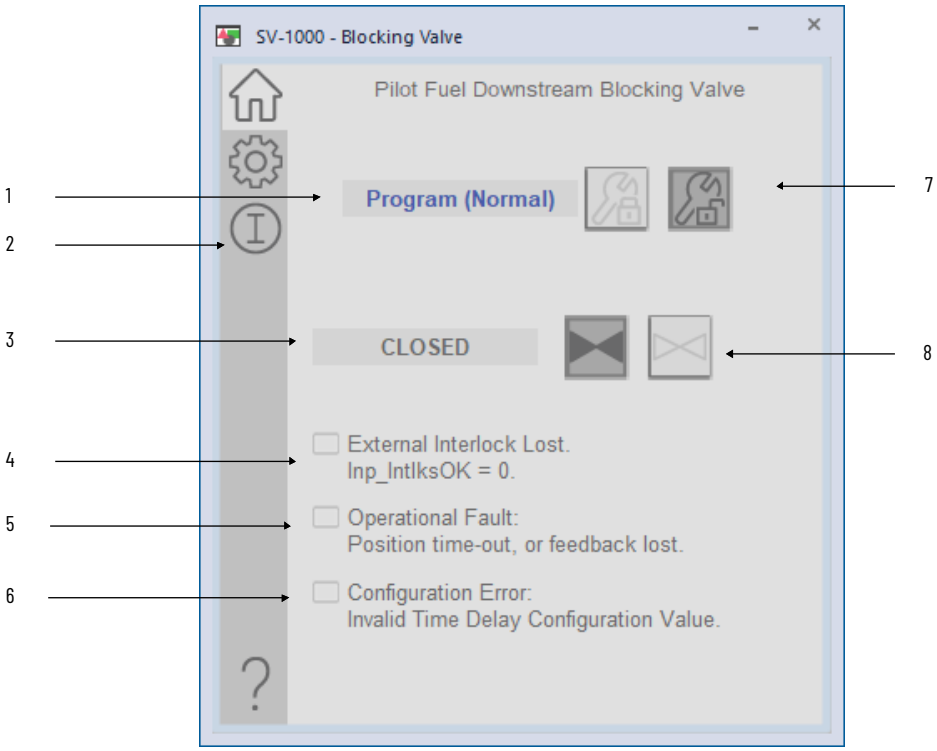
Graphic Symbol Name	Graphic Symbol	Description	Global Object Parameter Values
GO_BMSD2S_FCNoNav		Fail Close D2S device without navigation. The text on the button face is set to the tag's .Inf_Label.@Description field.	#102: AOI Instance (e.g. {::[PAC]Program::SafetyProgram._InstanceName}) #103: HMI Control Tag
GO_BMSD2S_FONoNav		Fail Open D2S device without navigation. The text on the button face is set to the tag's .Inf_Label.@Description field.	#102: AOI Instance (e.g. {::[PAC]Program::SafetyProgram._InstanceName}) #103: HMI Control Tag
GO_D2SStatus		Generic D2S device with faceplate navigation. This launch button graphic object allows the user to navigate to the object faceplate. The text on the button face is set to the tag's .Inf_Label.@Description field.	#102: AOI Instance (e.g. {::[PAC]Program::SafetyProgram._InstanceName}) #103: HMI Control Tag #120: Faceplate X Position, pixels from left edge of display (e.g. /X100) #121: Faceplate Y Position, pixels from top edge of display (e.g. Y100) #122: Personality tag (Const\Num0=Generic, Const\Num1=FC,Const\Num2=FO)

Faceplates

There are basic faceplate attributes that are common across all instructions. See Basic Faceplate Attributes on page 21.

Home

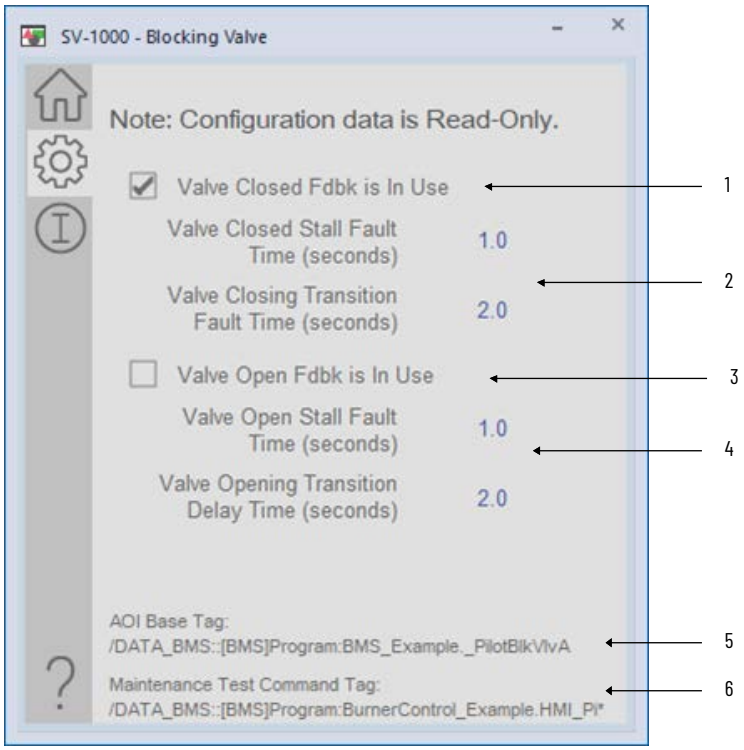
The Home tab is the main tab of the faceplate. It provides the status of the device along with the operating mode and related interlocks, faults, and errors.



Item	Description
1	Device Operating Mode indicator.
2	Device Interlock status summary indicator, showing all Interlock conditions OK.
3	Device Operating State indicator.
4	Device External Interlock Input status.
5	Device Operational Fault status.
6	Device Configuration Fault status.
7	Device Maintenance Test Request and Relinquish Buttons.
8	Maintenance Test Mode Command Buttons.

Configuration

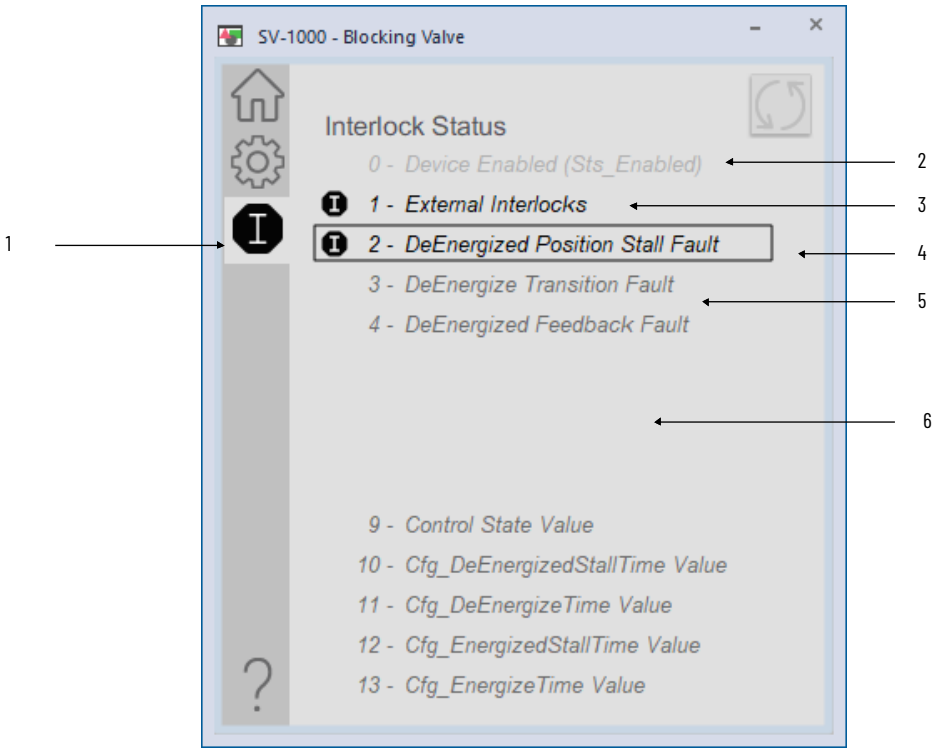
The Home tab is the main tab of the faceplate. It provides the status of the device along with the operating mode and related interlocks, faults, and errors.



Item	Description
1	Device De-Energized Position Feedback in use selector.
2	Device De-Energized Position Fault Delay times.
3	Device Energized Position Feedback in use selector.
4	Device Energized Position Fault Delay times.
5	Device AOI base tag.
6	Device AOI Maintenance Test Commands HMI tag.

Interlock Status

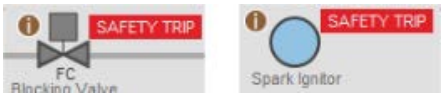
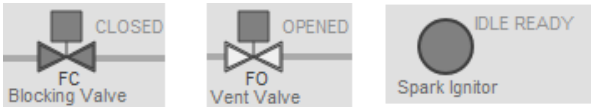
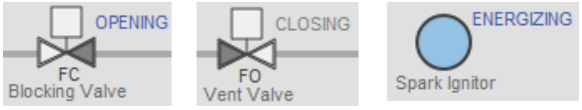
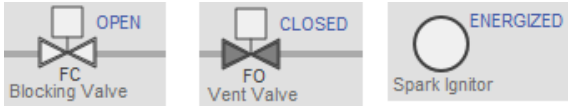
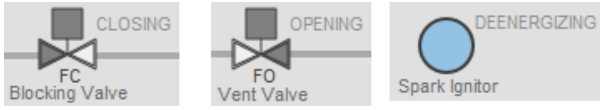
The Home tab is the main tab of the faceplate. It provides the status of the device along with the operating mode and related interlocks, faults, and errors.



Item	Description
1	Summary indicator – at least one Interlock is NOT OK.
2	Interlock configured for use, but currently not monitored.
3	Interlock currently monitored, but NOT OK to operate.
4	Interlock currently monitored, NOT OK, and has generated a Safety Trip.
5	Interlock currently monitored, but OK to operate.
6	Interlock not configured for use (Energized Position Stall Fault in this case).

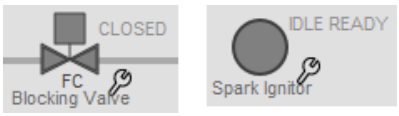
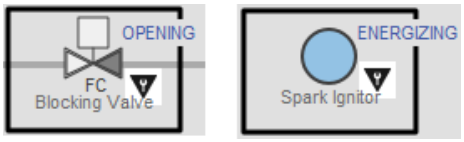
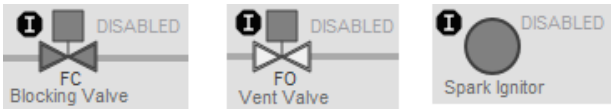
State Enumerations

The following table identifies the states available to the Device when enabled (Sts_Enabled is True). If Sts_Enabled is False, the Device is allowed to only be in States 0, 1, and 2, and will indicate DISABLED.

Val_eState	State Name	Description	HMI Graphic
0	Initializing	AOI performing 1st scan initialization.	
1	Safety Trip	Device Safety Trip present.	
2	Interlocked	Device Interlock not met.	
3	N/A	Undefined	
4	Idle / Ready	Device Idle (De-Energized) and Ready for Energize Command.	
5	Energizing	Energize Command issued, position not confirmed.	
6	Energized	Energize Command issued and position confirmed.	
7	De-Energizing	De-Energize Command issued, position not confirmed.	

Modes

Mode Name	Description	HMI Graphic
Normal	Normal Operating Mode (Sts_Normal = 1)	See State Table

Mode Name	Description	HMI Graphic
Test Enable	Test Mode Enabled (Sts_Test = 1)	
Device Test Active	Device Testing is Active (Sts_TestActive = 1)	
Disabled	Device Disabled (Sts_Enabled = 0)	

Implementation Checklist

Purpose

The following checklist is intended to aid the user in their implementation of the Burner Management Suite Library. It highlights specific items that may impact the effectiveness of the Burner Management Suite but is not intended to be an all-inclusive list addressing every possible implementation. Because of the variety of uses for this Library and information presented in this User Manual, the user or those responsible for applying this information must satisfy themselves as to the acceptability of each application and use of the Burner Management Suite Library and must confirm that their implementation of this Library meets all applicable regulatory codes.

Checklist

1. When using the raS_Opr_BMS_Interface AOI, ensure that the SINT array (standard tag) used as the Ref_ToMaster parameter is mapped to its Safety Tag equivalent used by the raS_Opr_BMS_Master AOI. Note that both arrays need to be the same dimension for successful mapping.
2. Ensure proper interaction with the burner's combustion control (air/fuel ratio control) logic.
   a. Out_FuelAirA..D: Signals from raS_Opr_BMS_Master indicating which fuel or fuels are currently active and need to be considered in air/fuel ratio calculations.
   b. Inp_CCSFuelA..D: Signals from combustion control logic to raS_Opr_BMS_Interface AOI confirming active fuels are included in air/fuel calculations.
   c. Inp_FAROK:
   d. Signal from combustion control logic to raS_Opr_BMS_Interface AOI confirming air/fuel calculations are valid and actuators/flows are within specified ranges.
   e. Out_Modulate: Signal from raS__Opr_BMS_Master for modulation release.
   f. Out_RunFans: Signal from raS_Opr_BMS_Purge requesting all fans to run.
   g. Out_LightOff: Signal from Purge for Air Actuator(s) to Light-Off request, and from Fuel/Pilot for Fuel Actuator(s) to Light-Off position request.
   h. Out_Hold: Signal from Purge for Air Actuator(s) to Hold Position request, and from Fuel/Pilot for Fuel Actuator(s) to Hold Position request.
   i. Out_Release: Signal from Purge for Air Actuator(s) to release for modulation, and from Fuel/Pilot for Fuel Actuator(s) to release for modulation.
3. Confirm proper configuration of raS_Opr_BMS_Master.
   a. Review and record settings for all configuration parameters.
   b. Confirm all I/O connections.

a. Ensure the number of Pilots and Fuels configured matches the number of AOI instances in your BMS logic.
b. If a multi-burner application, ensure the raS_Opr_BMS_Master configuration parameter Cfg_AnyFlameOK is NOT set.
c. The Cfg_AnyFlameOK parameter is used for single burner applications only and must be clear/zero for multi-burner applications.
d. Ensure the tag used in the Ref_MFTR parameter is the same tag used by the raS_Dvc_BMS_D2S driving the master fuel trip relay.

4. Confirm proper configuration of raS_Opr_BMS_Purge.
   a. Review and record settings for all configuration parameters.
   b. Confirm all I/O connections.
   c. Ensure raS_Opr_BMS_Purge uses the appropriate elements for its Ref_FromMaster and Ref_ToMaster parameters from the same tag used by the raS_Opr_BMS_Master.
5. Confirm proper configuration of all raS_Opr_BMS_Pilot instances.
   a. Review and record settings for all configuration parameters for each instance.
   b. Confirm all I/O connections.
   c. Ensure each raS_Opr_BMS_Pilot instance uses the appropriate elements for its Ref_FromMaster and Ref_ToMaster parameters from the same tag used by the raS_Opr_BMS_Master, and that each instance uses the same array element in both parameters.
   d. Note that element [0] correlates to Pilot A, [1] correlates to Pilot B, [2] correlates to Pilot C, and [3] correlates to Pilot D.
   e. Ensure the tags used in the raS_Opr_BMS_Pilot's device InOut parameters are the same tags used by the raS_Dvc_BMS_D2S driving each device.
6. Confirm proper configuration of all raS_Opr_BMS_Fuel instances.
   a. Review and record settings for all configuration parameters for each instance.
   b. Confirm all I/O connections.
   c. Ensure each raS_Opr_BMS_Fuel instance uses the appropriate elements for its Ref_FromMaster and Ref_ToMaster parameters from the same tag used by the raS_Opr_BMS_Master, and that each instance uses the same array element in both parameters. Note that element [0] correlates to Fuel A, [1] correlates to Fuel B, [2] correlates to Fuel C, and [3] correlates to Fuel D.
   d. Ensure the tags used in the raS_Opr_BMS_Fuel's device InOut parameters are the same tags used by the raS_Dvc_BMS_D2S driving each device.
7. Confirm proper correlation between each raS_Dvc_BMS_D2S instance's MCmd and Out_CmdAck.
   a. Ensure each raS_Dvc_BMS_D2S instance uses a unique tag for its MCmd.
   b. Ensure the standard version of MCmd tied to the HMI faceplate is mapped correctly to the intended device's raS_Dvc_BMS_D2S.
   c. Ensure each raS_Dvc_BMS_D2S uses a unique tag for its Out_CmdAck, and that tag clears the standard version of that D2S's MCmd tag.
8. Apply Safety Task Signature.
9. Test Burner for proper operation.
   a. If modifications to configuration or logic are required, the Safety Task Signature must be removed to make any such modifications.

b. Once changes are made, reapply Safety Task Signature and retest.

10. Upon successful test and acceptance of operation, record the BMS program's Safety ID and Safety Updated date/time stamp. These values will change only if logic or static tag values (e.g. configuration) within the BMS Program are modified. Tag values that are modified by safety logic will not impact the Safety ID nor the Safety Updated date/time stamp.

Rockwell Automation Support

Use these resources to access support information.

Technical Support Center	Find help with how-to videos, FAQs, chat, user forums, Knowledgebase, and product notification updates.	rok.auto/support
Local Technical Support Phone Numbers	Locate the telephone number for your country.	rok.auto/phonesupport
Technical Documentation Center	Quickly access and download technical specifications, installation instructions, and user manuals.	rok.auto/techdocs
Literature Library	Find installation instructions, manuals, brochures, and technical data publications.	rok.auto/literature
Product Compatibility and Download Center (PCDC)	Download firmware, associated files (such as AOP, EDS, and DTM), and access product release notes.	rok.auto/pcdc

Documentation Feedback

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

Waste Electrical and Electronic Equipment (WEEE)

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

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

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

Connect with us.

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

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

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

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

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

Publication BMS-RM001A-EN-P - May 2025

Supersedes Publication N/A - N/A