Logix SIS

GuardLogix 5580 Controllers

Allen-Bradley
by ROCKWELL AUTOMATION

Safety 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.

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

Table of Contents

Preface

This manual describes safety considerations for Logix SIS (safety instrumented system) operations, which are type-approved and certified for use in SIL 2 or SIL safety applications.

Use this manual for the development, operation, and maintenance of safety components that use Logix SIS. Read and understand the safety concepts and the requirements in this manual and familiarize yourself with applicable standards, such as IEC 61508, IEC 62061, IEC 61511, and ISO 13849-1.

Summary of Changes

The following is new or updated information. This list includes substantive updates only and is not intended to reflect all changes. Translated versions are not always available for each revision.

Topic	Page
Updated PDF_{avg} (20 yr) and LORF values	Logix SIS Safety Data on page 11

Additional Resources

These documents contain additional information concerning related products from Rockwell Automation. You can view or download publications at rok.auto/literature.

Resource	Description
ControlLogix and GuardLogix Controllers Specifications Technical Data, publication 1756-TD001	Lists product specifications and certifications for ControlLogix® and GuardLogix® controllers.
High Availability Systems Reference Manual, publication HIGHAV-RM002	Provides information to help design and plan high availability systems.
Redundancy Systems User Manual, publication 1756-UM015	Describes how to set up, configure, program, monitor, and troubleshoot Logix SIS, ControlLogix® 5580, and ControlLogix® 5570 redundancy systems.
ControlLogix 5580 and GuardLogix 5580 Controllers User Manual, publication 1756-UM543	Provides information on how to install, configure, program, and use ControlLogix® 5580 controllers and GuardLogix® 5580 controllers in Studio 5000 Logix Designer® projects.
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.
FLEX 5000 Standard and Safety I/O Modules User Manual, publication 5094-UM001	Describes how to useFLEX 5000® standard and safety I/O modules in Logix 5000® control systems.
FLEX 5000 Analog Isolated Current/Voltage/HART Standard and Safety I/O Modules User Manual, publication 5094-UM007	Describes how to use FLEX 5000® I/O analog standard and safety HART modules in Logix 5000® control systems.
CIP Safety: Safety Networking for Today and Beyond White Paper, publication SAFETY-WP038	Describes the benefits and implementation of CIP Safety™ networks.
EtherNet/IP Device Level Ring Application Technique, publication ENET-AT007	Describes Device Level Ring (DLR) topologies, configuration considerations, and diagnostic methods.

Resource	Description
EtherNet/IP Parallel Redundancy Protocol Application Technique, publication ENET-AT006	Describes Parallel Redundancy Protocol (PRP) topologies, configuration considerations, and diagnostic methods.
ControlLogix EtherNet/IP Network Devices User Manual, publication 1756-UM004	Describes how to useEtherNet/IP™ communication modules with a Logix 5000® controller and communicate with various devices on the EtherNet/IP™ network.
Safety Function Application Techniques (SFAT) Index, publication SAFETY-AT999	Lists the available application technique publications for safety functions.
Logix SIS technical documentation	Quickly access and download technical specifications, installation instructions, and user manuals.
System Security Design Guidelines Reference Manual, 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.
Industrial Automation Wiring and Grounding Guidelines, 1770-4.1	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Selection and Configuration tools, rok.auto/systemtools	Helps configure complete, valid catalog numbers and build complete quotes based on detailed product information.
Product Certifications website, rok.auto/certifications	Provides declarations of conformity, certificates, and other certification details.

Logix SIS Safety Concept

The Safety Integrity Level (SIL) is the relative level of risk-reduction that is provided by a safety function.

Logix SIS is rated for the following SIL and Performance Level (PL) functional safety standards.

Table 1. Logix SIS Safety Ratings

Certification	Rating
IEC 61508	Type-approved and certified for use in safety applications up to and including the following: • SIL 2 • SIL 3
IEC 62061	Suitable for use in safety applications up to and including the following: • SIL CL 2 • SIL CL 3
IEC 61511	Suitable for use in safety applications up to and including the following: • SIL 2 • SIL 3
ISO 13849-1	Suitable for use in safety applications up to and including the following: Performance Level PLe (Cat. 3)

Understand the following conventions used throughout this documentation:

- SIL 2 represents SIL 2, SIL CL 2, and PLd
- SIL 3 represents SIL 3, SIL CL 3, and PLe
- MRT refers to both MRT (mean repair time) and MTTR (mean time to respond). The diagnostic time to detect a failure is so quick that MRT is effectively equivalent to MTTR for our products.

TÜV Rheinland has approved Logix SIS for use in safety-related applications where the de-energized state is considered to be the safe state.

All I/O examples in this manual are based on achieving de-energization as the safe state for typical machine safety and emergency shutdown (ESD) systems.

IMPORTANT:

As the system user, you are responsible for these items:

- The setup, SIL rating, and validation of any sensors or actuators that are connected to the system
- Project management and functional test
- Access control to the safety system, including password management
- Application programming and the device configurations in accordance with the information in this safety reference manual and the Redundancy Systems User Manual, publication 1756-UM015.

When applying functional safety, restrict access to qualified, authorized personnel who are trained and experienced.

Use the Studio 5000 Logix Designer® application to create programs for safety controllers.

IMPORTANT: Only the safety task, not standard tasks, can be used for safety functions.

SIL Requirements

A risk assessment determines whether a safety function requires SIL 2 or SIL 3.

ATTENTION: The safety signature is required for the controller to operate at a SIL 2 or SIL 3 rating. Controller operation without a safety signature is only suitable during development.

IMPORTANT:

High-demand SIL 2 applications that follow the IEC 61511 standard require a Hardware Fault Tolerance of 1 and must use the guidance for SIL 3 requirements throughout this documentation.

High-demand SIL 2 applications that follow the IEC 62061 standard require a Hardware Fault Tolerance of 0 and must use at least the guidance for SIL 2 requirements throughout this documentation.

SIL 2 Requirements

The following applies to SIL 2 safety functions:

- Controller redundancy is not required, but recommended for high availability.
- There is no mean repair time (MRT) requirement. Timely system repair is not required, but is recommended for high availability.

IMPORTANT: If operating above 55 °C (131 °F) in a SIL 2 application, modules greater than 6.2 W must not be installed in slots that are next to a safety controller.

SIL 3 Requirements

To following applies to SIL 3 safety functions:

- Controller redundancy is required, and you must monitor the system for a loss of redundancy.
- There is a mean repair time (MRT) requirement. If a loss of redundancy occurs, timely system repair is required within your specified MRT.
- If the system is not repaired within the MRT, you must take a specified action to maintain or achieve a safe state.
- Upon power-up, SIL 3 safety functions are not permitted to be reset until controller redundancy is achieved.

IMPORTANT: The safety task can contain a number of safety functions. For a particular function to be SIL 3, the entire chain of devices and programming from the sensor to the actuator must be SIL 3. Be careful that you do not use a SIL 2 input signal for a safety function that requires SIL 3.

We recommend that you monitor the Redundancy Status bit (S:R) for a loss of redundancy and start a timer if the S:R bit goes to 0:

- When the timer reaches your specified MRT, execute logic to achieve or maintain a safe state.
- If system requalification causes the S:R bit to go to 1 before the timer expires, stop the timer and resume normal execution.

For more information about the S:R bit, see Redundancy Status on page 73.

Safety Application Requirements

IEC 61508 requires you to perform various proof tests of the equipment that is used in the system. Proof tests are performed at user-defined times. For example, proof tests can be once a year, once every 15 years, or whatever time frame is appropriate. The controllers have a useful life of 20 years, and no proof test is required. Other components of the system, such as safety I/O devices, sensors, and actuators can have different useful life times.

IMPORTANT: Your specific applications determine the time frame for the proof test interval.

Controller Specifications and Certifications

For specifications and the agency certifications for the products, see the following:

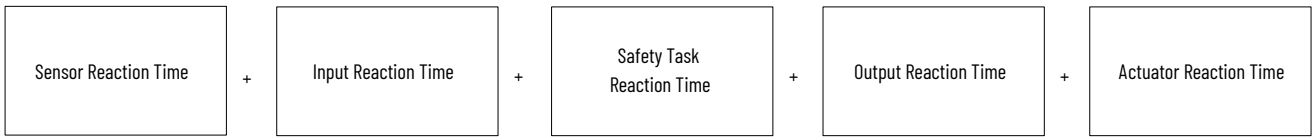
- The Specifications in the Technical Data publication for your product.
- Agency certifications on the product labels
- Declarations of conformity, certificates, and certification details at rok.auto/certifications

System Reaction Time

The system reaction time is the worst-case time from a safety-related event as input to the system or as a fault within the system, until the time that the system is in the safe state.

This worst-case definition includes the effects of asynchronous communications and multiple potential faults within the system. For high demand applications, the system reaction time includes safety function muting time.Actual reaction times are typically faster.

Figure 1. System Reaction Time



Each of the reaction times is dependent on factors such as the type of I/O device and instructions that are used in the program.

For more information about reaction time calculations, see Reaction Times on page 91.

Contact Information if Device Failure Occurs

If you experience a failure with any device, contact Rockwell Automation Technical Support: rok.auto/knowledgebase

Your local Rockwell Automation sales office or Allen-Bradley distributor can also initiate the following actions:

- Return the device to us so the failure is logged for the affected catalog number and a record is made of the failure.
- Request a failure analysis to try to determine the cause of the failure.

Logix SIS Safety Data

Use the following information to determine probability of a dangerous failure on demand (PFD) and average frequency of a dangerous failure per hour (PFH) values. These values apply to GuardLogix® 5580 controllers in a Logix SIS 1oo2 SIL 3 system.

Useful Life

The useful life of safety controllers is 20 years.

Safety Data

For safety I/O devices safety data, including PFD and PFH values, see the manuals for those products.

Product Failure Rates

For a mean repair time (MRT) of 72 hours, use the safety calculations in the following table.

Table 2. Safety Calculations

Attribute	MRT = 72 hr
PFH	1.367E-10
PFD_{avg} (20 yr)	1.923E-05

Assumptions for safety calculations:

- Component failure rates are constant over the life of the product.
- Within the specified useful life (20 years), no proof test is needed.

IMPORTANT: To minimize system degradation during repair time, use the shortest MRT possible for your application. Exceeding an MRT of 72 hours is not recommended. If you can justify an MRT longer than 72 hours for your application, then you must use the formulas below to compute PFD_{avg} and PFH.

For an MRT other than 72 hours, use the following PFD and PFH formulas to calculate PFD_{avg} or PFH. Repair time is included in the calculations.

$$PFD_{avg} = \frac{1}{2}\beta\lambda_{DU}T + \frac{1}{3}(1-\beta)^2\lambda_{DU}^2T^2 + (\lambda_{DD} + \lambda_S + \lambda_{NPED})\lambda_{DU}MRT^2 + LORF$$

$$PFH = \beta\lambda_{DU} + (1-\beta)^2\lambda_{DU}^2T + (\lambda_{DD} + \lambda_S + \lambda_{NPED})\lambda_{DU}MRT$$

IMPORTANT: These calculations use assumptions that become invalid when the MRT value gets too large. The maximum MRT value permitted for SIL 3 safety functions in Logix SIS is 730 hours.

Table 3. Safety Parameters

Parameter	Value	Description
λ_{DU}	$6.40 \times 10^{-9} hr^{-1}$	Dangerous undetected failure rate of a single controller
λ_{DD}	$6.54 \times 10^{-7} hr^{-1}$	Dangerous detected failure rate of a single controller

Table 3. Safety Parameters (continued)

Parameter	Value	Description
λ_S	$6.61 \times 10^{-7} hr^{-1}$	Safe failure rate of a single controller
λ_{NPED}	$2.58 \times 10^{-6} hr^{-1}$	No part/effect detected failure rate of a single controller
λ_{Det}	$3.895 \times 10^{-6} hr^{-1}$	Total detected failure rate of a single controller ($\lambda_{DD}+\lambda_S+\lambda_{NPED}$)
T	≤ 20 years (17,5200 hours)	Mission time Mission time is the length of time over which the device maintains the stated PFD, PFH, and λ ratings before replacement is required.
T_D	< SRT	Diagnostic test interval
β	2%	Common cause percentage
HFT	1[1]	Hardware fault tolerance The hardware fault tolerance equals *n*, where *n*+1 faults could cause the loss of the safety function. An HFT of 1 means that 2 faults are required before safety is lost.
LORF	7.61×10^{-6}	Loss of redundancy factor
MRT	[User-defined value]	Mean repair time for a failed controller
SC	3	Systematic capability Systematic capability is defined in IEC 61508-4 as the confidence that the systematic safety integrity meets the requirements of the specified SIL rating.

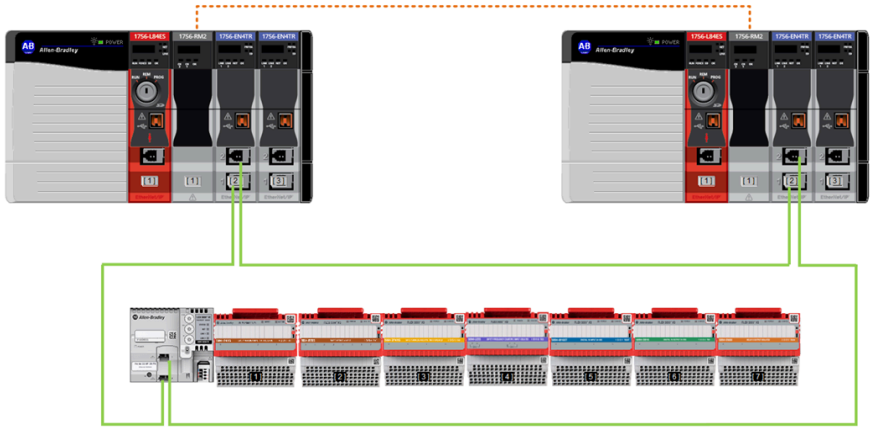
1. When a controller chassis fails, the safety function continues to function with a HFT of 0 unit until the controller chassis synchronize.

Logix SIS Architecture

Logix SIS architecture is defined by these characteristics:

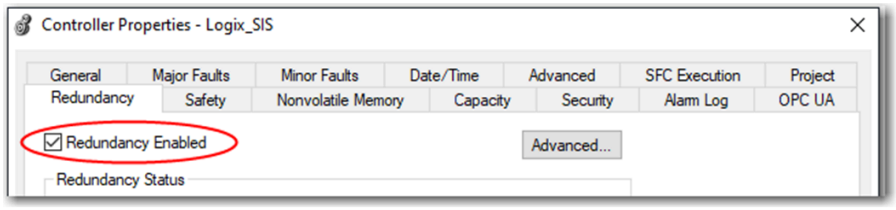
- Redundant safety controllers and other hardware components that control safety-related functions.
- Concurrent execution of the safety task on primary and secondary qualified safety controllers.
- Dynamic transitions between a 1oo1 and 1oo2 safety architecture during these events:
  - Qualification of the redundant chassis pair—Once safety controllers are qualified and synchronized in a redundant chassis pair, the system transitions from a 1oo1 to 1oo2 architecture. In this operational state, both controllers cross-check the safety task results and either controller can respond to a safety demand.
  - Loss of redundancy—If the system loses redundancy, the system transitions from a 1oo2 to a 1oo1 architecture. In this operational state, one of the redundant controllers no longer participates in the safety function while the other continues to execute the safety function alone.
- Concurrent communication with remote FLEX 5000® safety I/O modules.

Figure 2. Logix SIS Architecture



To operate as described, Logix SIS requires these configuration settings in the controller properties:

- Redundancy must be enabled on the Redundancy tab.



- The safety level on the Safety tab must be set appropriately for the firmware revision you are using as shown below. In firmware revision 38, the safety level is static and cannot be changed when the controller is configured for Logix SIS.

Table 4. Safety Level in Controller Properties

Firmware Revision	Safety Level
38 or later	SIL3/PLe when synchronized or disqualified within MRT; SIL2/PLd when disqualified
37	SIL2/PLd

IMPORTANT:

- With firmware revision 38 or later, the GSV SafetySILConfiguration attribute always shows a SIL 3 value even when your system complies with a SIL 2 safety function.
- With firmware revision 37, the GSV SafetySILConfiguration attribute always shows a SIL 2 value even when your system complies with a SIL 3 safety function. A SIL2/PLd safety level is the required configuration for controllers that are enabled for redundancy.

Safety Functions of Logix SIS Components

The following sections discuss the safety functions of Logix SIS components.

For supported catalog numbers and configuration requirements for Logix SIS system components, see the Redundancy Systems User Manual, publication 1756-UM015.

Redundant Safety Controllers

Redundant safety controllers provide the following:

- Control for safety-related functions in the system.
- Powerup and runtime functional-diagnostic tests of all safety-related components in the controller.

Redundant Chassis

The redundant chassis provides the physical connections between modules and the system. Any failure is detected as a failure by one or more of the active components of the system. As a result, the chassis is not relevant to the safety discussion.

Redundant Power Supplies

No extra configuration or wiring is required for SIL 2 or SIL 3 operation of redundant power supplies. Any failure is detected as a failure by one or more of the active components of the system. As a result, the power supply is not relevant to the safety discussion.

Communication Modules

Logix SIS uses EtherNet/IP™ communication modules to control and exchange safety data on the EtherNet/IP™ network.

IMPORTANT: The EtherNet/IP™ connection from the front port of a safety controller is not supported.

For communication with remote FLEX 5000® safety I/O, Logix SIS requires that you use one of the following communication modules that is configured for concurrent communication:

1756-EN4TR, 1756-EN4TRK, 1756-EN4TRXT

Standard I/O modules that communicate via concurrent communication are supported on the same communication modules.

For more information, see Concurrent Communication in Logix SIS on page 18.

Safety I/O

To perform safety I/O functions, redundancy-enabled safety controllers can interface only with remote FLEX 5000 safety I/O modules through communication modules that are configured for concurrent communication. These requirements enable a safety controller to execute safety I/O functions without disruption if a loss of redundancy occurs.

Safety I/O devices, like sensors and actuators, can be connected to remote safety I/O modules. The safety controller monitors and controls the devices.

Safety I/O communication uses the CIP Safety™ protocol. Safety logic is processed in the safety controller.

For more information, see Safety I/O on page 27.

Human Machine Interfaces

Follow these precautions and guidelines for HMI devices in SIL-rated safety systems.

In SIL-rated safety systems, you must exercise precautions and implement specific techniques on HMI devices. These precautions include, but are not restricted to the following:

- Limited access and security
- Specifications, testing, and validation
- Restrictions on data and access
- Limits on data and parameters

Use sound techniques in the application software within the HMI and controller.

HMI-related functions consist of two primary activities:

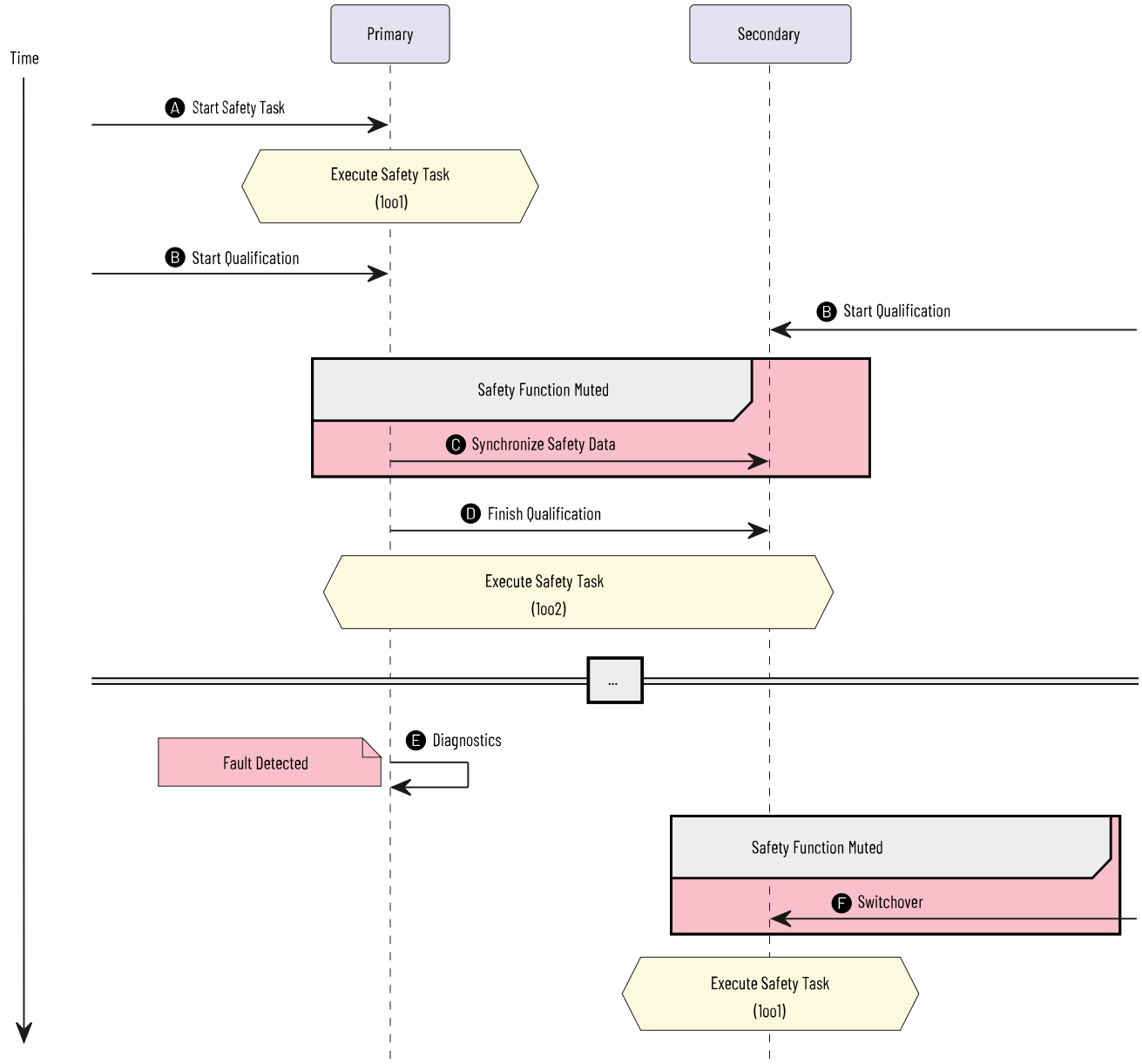
- Reading data
- Writing data

Reading data is unrestricted because reading does not affect the behavior of the safety system. However, the number, frequency, and size of the data being read can affect controller availability. To avoid safety-related spurious trips, use good communication practices to limit the impact of communication processing on the controller. Do not set read rates to the fastest rate possible.

Safety Task Execution

The following diagram shows an example of safety task execution in Logix SIS. Each step in the process corresponds to a 1oo1 or 1oo2 safety architecture, which can dynamically transition depending on the current operational state of the system.

Figure 3. Example of Safety Task Execution



Item	Description	Safety Architecture
A	The unsynchronized primary controller executes the safety task in a 1oo1 configuration.	1oo1
B	The qualification and synchronization process begins between the primary and secondary controllers in the redundant chassis pair.	1oo1
C	Safety data synchronizes between the primary and secondary controllers. The safety function is temporarily muted and cannot respond to safety demands.	1oo1

Item	Description	Safety Architecture
D	The qualification process finishes, the primary and secondary controllers are synchronized. Both controllers execute the safety task concurrently in a 1oo2 configuration.	1oo2
E	A safety diagnostic detects a fault on the primary controller. The safety function is temporarily muted and cannot respond to safety demands.	1oo1
F	The secondary controller assumes control of the safety task in a 1oo1 configuration.	1oo1

Safety Function Muting

Safety functions are temporarily suspended, or muted, during the following scenarios:

- Qualification of the redundant chassis pair

  When the safety function is muted, the percent complete shown on the RMCT Synchronization Status tab in the redundancy module properties is 30%. When the percent complete reaches 35%, the safety function is no longer muted even though qualification is still in progress.

  Once qualification is completed, the Monitor tab in the safety task properties shows the total time that the safety function was muted. The muting time appears in the Max Interval time field. To record future safety function muting times, you must reset this statistic.

- Loss of redundancy, such as during disqualification of the redundant chassis pair or a switchover
- Lock for update, such as during the Redundancy System Update (RSU) process

The following tables define the maximum duration of time that the safety function is muted depending on the scenario and controller that you are using. In most cases, the muting time is less than the maximum time shown in the following tables.

Table 5. Safety Function Muting Time

Scenario	Muting Time
Qualification	Up to 1 second + 1 safety task period.
Loss of redundancy	The muting time depends on the duration of the safety task period: • If the safety task period is below 50 ms, the muting time is up to 50 ms + 1 safety task period • If the safety task period is greater or equal to 50 ms, the muting time is up to 1 safety task period
Lock for update	Up to 2 seconds + 1 safety task period.

IMPORTANT: Muting time impacts safety reaction time in high-demand applications.

Low Demand Considerations

The Logix SIS reliability model considers safety function muting time. The PFD calculations conservatively assume that a system encounters 120 loss-of-redundancy events per year or an average of 10 per month. As a result, low demand safety functions do not need to account for muting time in overall safety reaction time calculations.

ATTENTION:

If your redundant chassis pair encounters more than 120 loss-of redundancy events per year, the PFD numbers in this manual are not valid, and you must include safety function muting time in your safety reaction time calculations.

PFD calculations consider a failed qualification attempt that reaches at least 30% to be a loss of redundancy event, which mutes the safety function.

High Demand Considerations

Because high demand safety functions use PFH as a safety performance target, you must include safety function muting time in your safety reaction time calculations. The amount of muting time to add to your calculations depends on the following scenarios:

- To account for a possible loss of redundancy, you must always add the loss of redundancy muting time to your calculations for high-demand safety functions.
- If the Auto-synchronization parameter in your redundancy module is set to Always or Conditional, you must add the qualification muting time to your calculations.
- If you do not use auto-synchronization, but rely on the safety function when you use the Synchronize Secondary redundancy command, you must add the qualification muting time to your calculations.

  If you can design your system so that safety demands do not occur during synchronization, then you may be able to omit the qualification muting time from your safety reaction time.

- If you use the RSU feature and rely on the safety function during the lock for update process, you must add the lock for update muting time to your calculations.

Concurrent Communication in Logix SIS

In Logix SIS, redundant controllers own the same safety I/O module via concurrent connections. Both controllers store the configuration data that you define for the I/O module. This configuration controls how the I/O modules operate in the system.

Synchronized, redundant controllers use concurrent communication to exchange safety data with remote FLEX 5000® safety I/O modules. A concurrent connection allows both controllers to maintain separate network paths to the same safety I/O module:

- Both controllers receive safety inputs and agree on which data to use for each safety task cycle.
- Both controllers send safety outputs and cross-check to verify that the outputs match.

If a fault occurs on one of the controllers, the other controller continues to execute the safety task alone. Because the concurrent connection is already established, there is no loss of service to the safety I/O module.

For concurrent communication system requirements for Logix SIS, see the Redundancy Systems User Manual, publication 1756-UM015.

Safety Signature

A safety signature verifies the integrity of a safety application:

- The safety signature applies to the entire safety portion of the controller project. The ability to create, record, and verify the safety signature is a mandatory part of the safety-application development process. The safety signature must be present to operate as a SIL 2/PLd or SIL 3/PLe safety controller. Nothing in the standard application is included in the safety signature.
- The safety signature is a hierarchy of multiple safety signature elements. For example, the safety task, programs, and routines are examples of safety signature elements. Safety signature elements can help you during impact analysis by identifying the individual changes within a controller project. If your validation plan does not require revalidation of unchanged elements, your certification effort can be reduced.
  All safety signature elements are created at the time that you generate the safety signature for the project. To view all safety signature elements for a project, you can run the Safety Signature report.

The safety signature and each of its elements have the following:

- Safety signature—IDA unique 64-character alphanumeric identification number.
- Time stamp—The date and time that the safety signature was generated. For a safety signature element, the time stamp changes whenever its signature ID changes. The time stamp is based on the local clock of the computer that generated the signature.

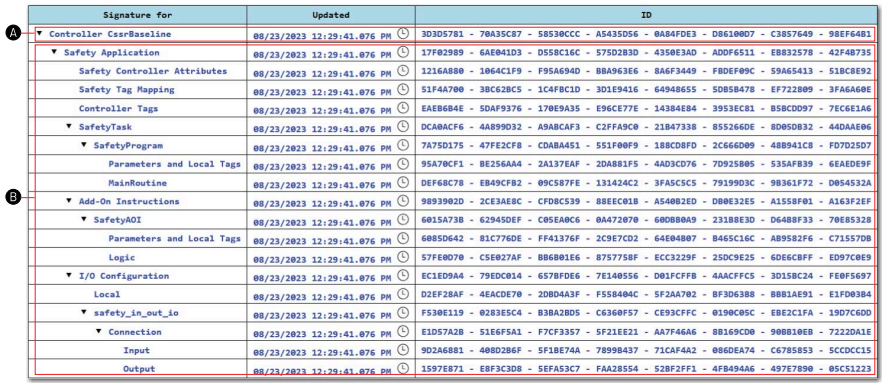
Figure 4. Safety Signature

Safety Signature Hierarchy

The following image shows an example of the safety signature hierarchy:

- A signature with ID and time stamp exists for each element in the hierarchy.
- Expand the safety signature at the top to view its underlying elements.
- Expand a parent element to view its underlying child elements.

Figure 5. Safety Signature Hierarchy



Item	Description
A	Safety signature

Item	Description
B	Safety signature elements

Aggregate Signatures

Safety elements that include a collection of tags and parameters have aggregate signatures. An aggregate signature represents the validity of all tags and parameters within the element.

The following elements have aggregate signatures:

- Controller-scoped safety tags
- Safety program parameters and tags
- Safety Add-On Instructions
- Safety mapped tags
- Safety I/O connections

If one of the preceding elements has a collection with no members, such as no controller-scoped safety tags, the aggregate signature value appears as 64 zeros.

View Safety Signature Elements

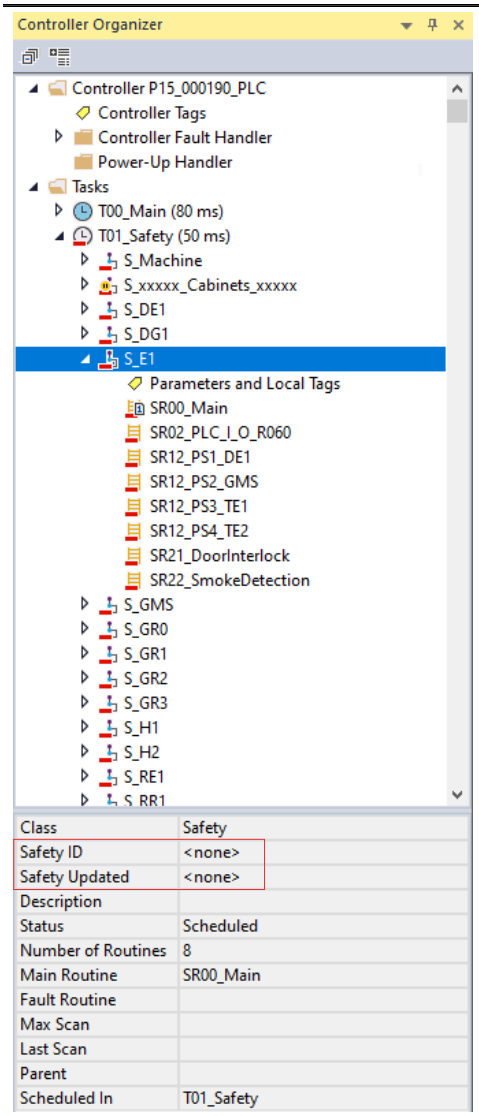
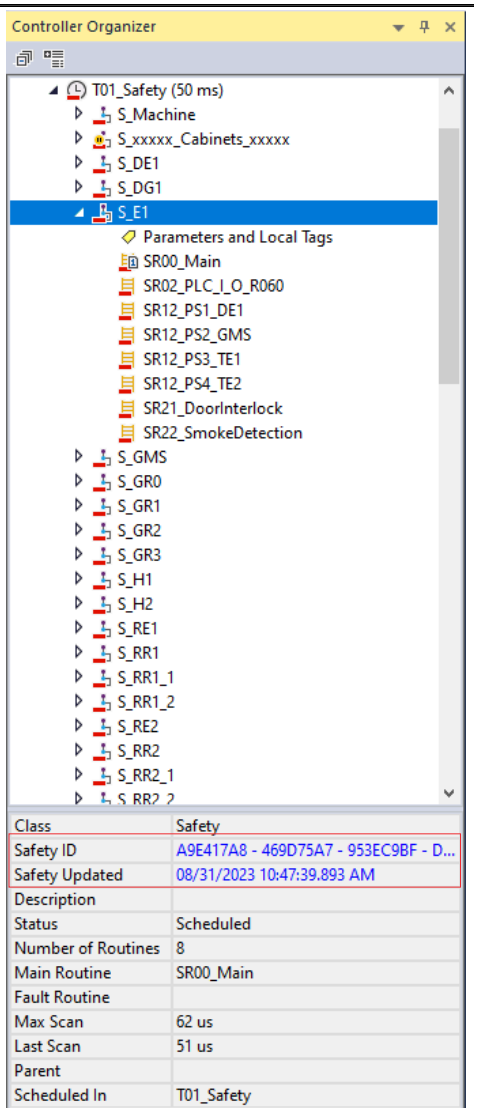
There are multiple places where you can view safety signature elements:

- Quick View pane
- Safety Signature report
- Compare Tool

Quick View Pane

When you select a safety element in the Controller Organizer, such as a safety program, its signature appears in the Quick View pane. Before the safety signature is generated, the Safety ID and Safety Updated fields in the Quick View pane display <none>.

To display all 64 characters of the safety signature ID, hover over the ID in the Quick View pane. From the tooltip, you can copy the signature ID.

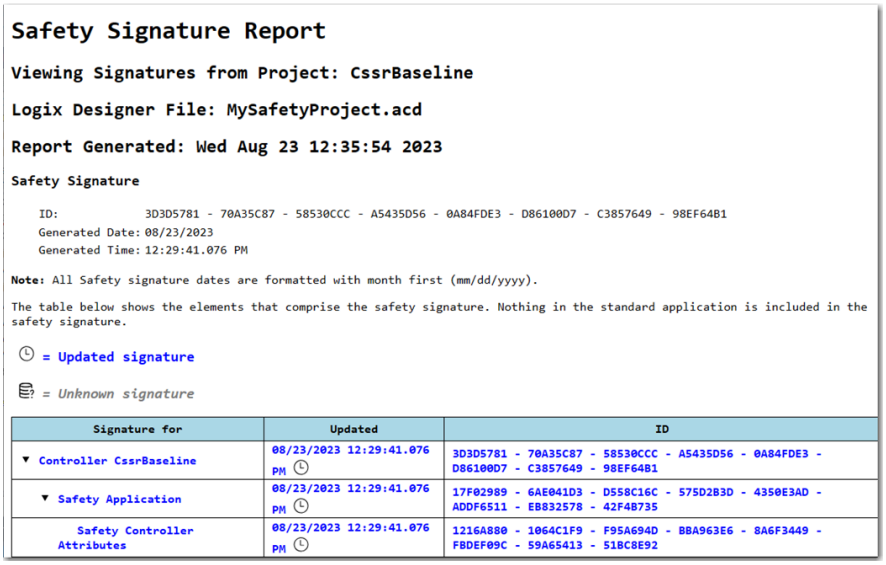
Before Safety Signature Generation	After Safety Signature Generation
	

Safety Signature Report

The Safety Signature Report includes all safety signature elements for the controller project. To view the report in the Logix Designer application, select Tools > Safety > Generate Signature Report. The report opens in a web browser and shows the following:

- The safety signature appears in the report header
- The safety signature elements appear in the report table.
- The state of the signature for an element is indicated by its color, tooltip, and icon.

Figure 6. Safety Signature Report



Safety Signature Report

Viewing Signatures from Project: CssrBaseline

Logix Designer File: MySafetyProject.acd

Report Generated: Wed Aug 23 12:35:54 2023

Safety Signature

ID: 3D3D5781 - 70A35C87 - 58530CCC - A5435D56 - 0A84FDE3 - D86100D7 - C3857649 - 98EF64B1
Generated Date: 08/23/2023
Generated Time: 12:29:41.076 PM

Note: All Safety signature dates are formatted with month first (mm/dd/yyyy).

The table below shows the elements that comprise the safety signature. Nothing in the standard application is included in the safety signature.

= Updated signature

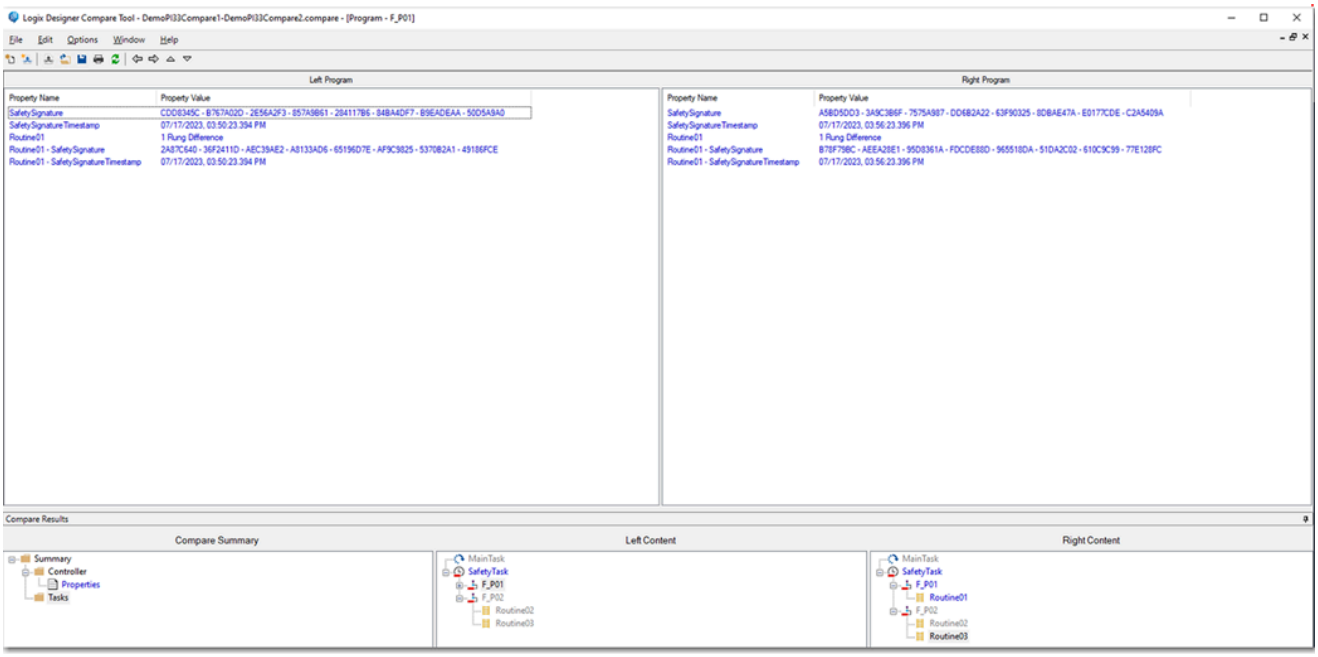
= Unknown signature

Signature for	Updated	ID
▼ Controller CssrBaseline	08/23/2023 12:29:41.076 PM	3D3D5781 - 70A35C87 - 58530CCC - A5435D56 - 0A84FDE3 - D86100D7 - C3857649 - 98EF64B1
▼ Safety Application	08/23/2023 12:29:41.076 PM	17F02989 - 6AE041D3 - D558C16C - 575D2B3D - 4350E3AD - ADDF6511 - EB832578 - 42F4B735
Safety Controller Attributes	08/23/2023 12:29:41.076 PM	1216A880 - 1064C1F9 - F95A694D - BBA963E6 - 8A6F3449 - FBDEF09C - 59A65413 - 51BC8E92

Logix Designer Compare Tool

The Logix Designer Compare Tool can compare safety signature elements in two controller projects. The following image compares two safety routine elements.

Figure 7. Logix Designer Compare Tool



Safety Signature States

A signature can have three states as indicated by its color: blue, black, or gray.

Blue Signature

A signature appears blue in these scenarios:

- After the first generation of the safety signature
- After any subsequent generation of the safety signature to indicate a change to a safety element or one of its child elements

In the following example, the Safety Signature report shows blue signatures as a result of the following workflow:

1. Generate a safety signature for the first time.
2. Create a safety tag in the S_DE1 program.
3. Generate the safety signature a subsequent time.
4. Generate the Safety Signature report.

The creation of the safety tag in step 2 results in blue signatures for these elements:

- Parameters and Local Tags child element
- S_DE1, T01_Safety, and Safety Application parent elements

Figure 8. Blue Parent and Child Signatures

Signature for	Updated	ID
▼ Controller P15_000190_PLC	08/31/2023 10:56:56.448 AM	92EE81B3 - 6D5EEF2A - 973E5DF7 - D248C4E6 - 0B440298 - 25EBDBDF - B05528F4 - E62AC3D0
▼ Safety Application	08/31/2023 10:56:56.448 AM	2339EBE5 - DFFA0683 - DB5E82D1 - 0BDB24E5 - 8FD99865 - 790E3559 - 9E79172C - 1FC51B33
Safety Controller Attributes	08/31/2023 10:47:39.893 AM	CB50C9B5 - 40ABA086 - C2DBF434 - B4A5AD5F - D6954D00 - E414236A - 56EF0365 - 4A148457
Safety Tag Mapping	08/31/2023 10:47:39.893 AM	7624105E - 14DC1627 - 70480CD0 - D149CB70 - CAF8ACF8 - F60FF95F - C375B865 - 607E245F
Controller Tags	08/31/2023 10:47:39.893 AM	94523844 - 802F6054 - 3E48FBBC - 2CD96795 - 84C96B0B - 2E54924E - 55BF47DD - 89560B6D
▼ T01_Safety	08/31/2023 10:56:56.448 AM	F95E9CFA - 7FE9096F - A61FC452 - C10379E0 - 934F55B5 - D514FD5E - 33F87770 - BF78B833
▼ S_DE1	08/31/2023 10:56:56.448 AM	841B5D04 - 64F14011 - C29CBCF0 - 29CA8465 - 0F6C2163 - D31F8E93 - 72FB1A0A - FD05E2B3
Parameters and Local Tags	08/31/2023 10:56:56.448 AM	01BCF740 - EEBE9E4A - 6ACB4F8F - 7324005F - 57676C7D - EAAE0D8C - 0F417DB6 - B2C44D72
SR00_Main	08/31/2023 10:47:39.893 AM	07887482 - 5BDD2C57 - CCE231D5 - 9ABA344D - 17E49D47 - AC0A1DE3 - 6FD68F96 - 052E95FB
SR02_PLC_I_O_R070	08/31/2023 10:47:39.893 AM	D4BF3380 - 1350D53E - 7700A2F0 - 71346E41 - 8DD46478 - 9A828ECB - BC31F0E6 - 71044374
SR11_PS1_DrivesSpoolChamber	08/31/2023 10:47:39.893 AM	758BC88A - D89A0E17 - BDFBE133 - 63AF8B7C - 18FC816B - B00AA734 - 99E9A629 - B12DE867
SR11_PS2_DrivesCrossChamber	08/31/2023 10:47:39.893 AM	F71AF1FB - B515EF3A - CCF8EB98 - 9AA93BFC - 47750943 - B9D56DF4 - 4720012D - B7BB5D14
SR11_PS3_DrivesPivotingPlatform	08/31/2023 10:47:39.893 AM	90DE4BFE - 2E89FC2C - 46DD3105 - 949951F5 - F2007F45 - 6325CC6C - B14BD602 - A32783C2
SR11_PS4_DrivesReactor1	08/31/2023 10:47:39.893 AM	27027869 - 79F283CA - 07AEF4CF - 049A137E - 8A473F77 - E29A011C - D79B0676 - 8DE09793
SR11_PS5_DrivesReactor2	08/31/2023 10:47:39.893 AM	0B91D7B3 - 7988D678 - F765E09D - EA08C187 - A98D4750 - E0C769F2 - 8E1437F4 - B77EFDAC
SR11_PS6_DrivesReactor3	08/31/2023 10:47:39.893 AM	8A6C1C8E - 51D7FCE2 - 2EB01ABA - 1190BCC4 - F35CE654 - 827FBD12 - 48D2194E - 312D02A6
SR20_CompressedAir	08/31/2023 10:47:39.893 AM	00919E67 - 69492280 - 102DB561 - 60168987 - 0BF1ED4B - 94448331 - EB04AB4A - 68BDB47D
SR21_DoorInterlock	08/31/2023 10:47:39.893 AM	C590CC40 - E6C5AA6A - 8DF80746 - C242DA93 - 118FAF12 - 283E1AD8 - 84172EE8 - 08B85DEF
SR22_SmokeDetection	08/31/2023 10:47:39.893 AM	17E1B3FF - 78480838 - 88A84206 - D6189781 - 330B77CC - D1C2360C - 7EC0A1D4 - AAC576E0
▼ S_DG1	08/31/2023 10:47:39.893 AM	CD1B4A25 - B22B1202 - A9A181CC - 0EA18119 - FDF7D1FB - 70797419 - 47507019 - A4DEF6D3

Black Signature

A signature appears black after any subsequent generation of the signature when no change occurred to the safety element or any of its child elements.

Gray Signature

A gray signature is an unknown signature. An unknown signature occurs when the Logix Designer application cannot verify that the signature ID matches the associated value. Unknown signatures cannot be used for certification and must be updated.

A signature appears gray when you perform any of the following workflows:

- Delete the safety signature
- Create a controller project from an imported project with an existing safety signature
- Copy a safety signature element with an existing signature

Archive Safety Signature Reports for Audit

You can compare an original Safety Signature report to an updated Safety Signature report to show which elements changed and which elements did not change for auditing purposes. If your

validation plan does not require revalidation of unchanged elements, your certification effort can be reduced, such as in these scenarios:

- You want to migrate a validated safety application to a future release without revalidating the entire application.
- You want to add or update some elements of a validated safety application without revalidating the unchanged elements.
- You want to copy some validated elements to another project without revalidating the copied elements.

Safety Signature reports are archived as .htm files in the following default directory:

C:\Users\Public\Documents\Rockwell Automation\Studio 5000\Logix Designer\Safety Signature Reports

Signature Authentication

If a signature for an element cannot be authenticated, the following occurs:

- A warning appears in the Errors pane below the Controller Organizer.
- No signature is shown for the element in the Quick View pane.

For example, if you copy a main routine, but the routine was changed before the copy, warnings appear to indicate that the signature could not be authenticated. No signature appears for the copied routine in the Quick View pane.

Figure 9. Signature Authentication Failed—Warning

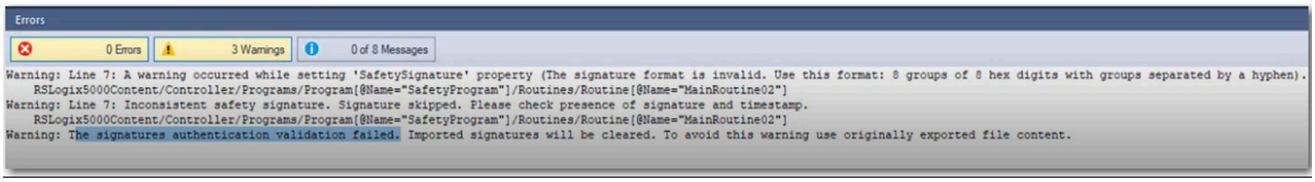
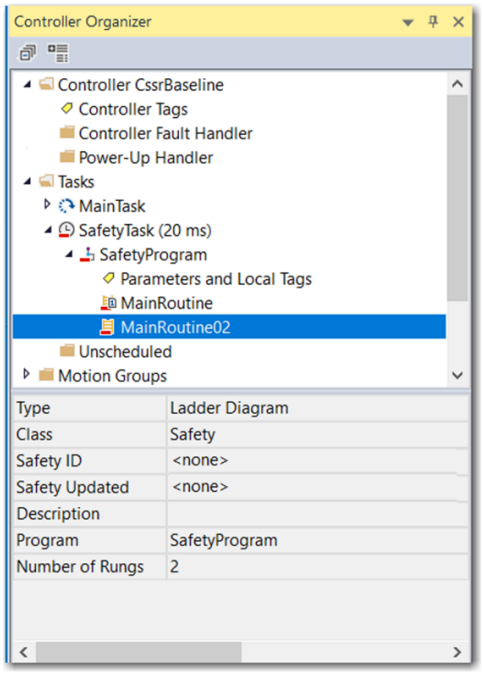


Figure 10. Signature Authentication Failed—No Signature



Settings that Do Not Affect the Safety Signature

In the Logix Designer application, version 36 or later, you can make the following changes to a safety project with no effect on the safety signature:

- Change the controller slot number, if applicable.
- Modify controller IP settings.
- Change the EtherNet/IP™ mode on a controller that support multiple modes.
  To change the EtherNet/IP™ mode, you must first delete the safety signature. After changing the mode, you can regenerate the safety signature with the same value.

Programmatic Changes to the Safety Application Signature

If there are no changes to your safety application, then the safety application signature and all of its underlying elements remain constant across software and firmware releases with the following possible exceptions:

- You update the controller with a firmware revision that contains a change to the offline compiler or internal signature algorithm. These changes can trigger a change to the safety signature. These types of changes are infrequent.
- You use the copy/paste or import function in a way that causes an internal difference in a component. For example, if you copy a safety routine from Project1 to Project2 and TagA is program-scoped in Project1, but controller-scoped in Project2, then the safety signature for the routine changes.

Unlike the safety application signature, the controller safety signature always changes with any firmware release change, so that you complete some validation testing at the functional controller level, even if no changes were made to the safety project.

Figure 11. Controller and Safety Application Signatures

Signature for	Updated	ID
Controller P15_000190_PLC	08/31/2023 10:56:56.448 AM	92EE81B3 - 6D5EEF2A - 973E5DF7 - D248C4E6 - 0B440298 - 25EBDBDF - B05528F4 - E62AC3D0
Safety Application	08/31/2023 10:56:56.448 AM	2339EBE5 - DFFA0683 - DB5E82D1 - 0BDB24E5 - 8FD99865 - 790E3559 - 9E79172C - 1FC51B33
Safety Controller Attributes	08/31/2023 10:47:39.893 AM	CB50C9B5 - 40ABA086 - C2DBF434 - B4A5AD5F - D6954D00 - E414236A - 56EF0365 - 4A148457
Safety Tag Mapping	08/31/2023 10:47:39.893 AM	7624105E - 14DC1627 - 70480CD0 - D149CB70 - CAF8ACF8 - F60FF95F - C375B865 - 607E245F
Controller Tags	08/31/2023 10:47:39.893 AM	94523844 - 802F6054 - 3E48FBBC - 2CD96795 - 84C96B0B - 2E54924E - 55BF47DD - 89560B6D
T01_Safety	08/31/2023 10:56:56.448 AM	F95E9CFA - 7FE9096F - A61FC452 - C10379E0 - 934F55B5 - D514FD5E - 33F87770 - BF7BB833
S_DE1	08/31/2023 10:56:56.448 AM	841B5D04 - 64F14011 - C29C8CF0 - 29CA8465 - 0F6C2163 - D31F8E93 - 72FB1A0A - FD05E2B3
Parameters and Local Tags	08/31/2023 10:56:56.448 AM	01BCF740 - EEBE9E4A - 6ACB4F8F - 7324005F - 57676C7D - EAAE0D8C - 0F417DB6 - B2C44D72
SR00_Main	08/31/2023 10:47:39.893 AM	07887482 - 5BDD2C57 - CCE231D5 - 9ABA344D - 17E49D47 - AC0A1DE3 - 6FD68F96 - 052E95FB
SR02_PLC_I_O_R070	08/31/2023 10:47:39.893 AM	D4BF3380 - 1350D53E - 7700A2F0 - 71346E41 - BDD46478 - 9A828ECB - BC31F0E6 - 71044374
SR11_PS1_DrivesSpoolChamber	08/31/2023 10:47:39.893 AM	758BCB8A - D89A0E17 - BDFBE133 - 63AF887C - 18FC816B - B00AA734 - 99E9A629 - B12DE867
SR11_PS2_DrivesCrossChamber	08/31/2023 10:47:39.893 AM	F71AF1FB - B515EF3A - CCF8EB98 - 9AA93BFC - 47750943 - B9D56DF4 - 4720012D - B7BB5D14
SR11_PS3_DrivesPivotingPlatform	08/31/2023 10:47:39.893 AM	90DE4BFE - 2E89FC2C - 46DD3105 - 949951F5 - F2007F45 - 6325CC6C - B14BD602 - A32783C2
SR11_PS4_DrivesReactor1	08/31/2023 10:47:39.893 AM	27027869 - 79F283CA - 07AEF4CF - 049A137E - 8A473F77 - E29A011C - D79B0676 - 8DE09793
SR11_PS5_DrivesReactor2	08/31/2023 10:47:39.893 AM	0B91D7B3 - 7988D678 - F765E09D - EA08C187 - A98D4750 - E0C769F2 - 8E1437F4 - B77EFDAC
SR11_PS6_DrivesReactor3	08/31/2023 10:47:39.893 AM	8A6C1C8E - 51D7FCE2 - 2EB01ABA - 1190BCC4 - F35CE654 - 827FBD12 - 4BD2194E - 312D02A6
SR20_CompressedAir	08/31/2023 10:47:39.893 AM	00919E67 - 69492280 - 102D8561 - 60168987 - 0BF1ED4B - 94448331 - EB04AB4A - 68BDB47D
SR21_DoorInterlock	08/31/2023 10:47:39.893 AM	C590CC40 - E6C5AA6A - 8DF80746 - C242DA93 - 118FAF12 - 283E1AD8 - 84172EE8 - 08B85DEF
SR22_SmokeDetection	08/31/2023 10:47:39.893 AM	17E1B3FF - 78480838 - 88A84206 - D6189781 - 330B77CC - D1C2360C - 7EC0A1D4 - AAC576E0
S_DG1	08/31/2023 10:47:39.893 AM	CD1B4A25 - B22B1202 - A9A181CC - 0EA18119 - FDF7D1FB - 70797419 - 47507019 - A4DEF6D3

Safety Signature Validation in Logix SIS

Safety signatures in redundant controllers must match and are validated by Logix SIS during these operations:

- Safety signature generation on a synchronized controller
- Qualification of the redundant chassis
- Download of a safety application with a safety signature

IMPORTANT:

When you perform the Redundancy System Update (RSU) process, the change in firmware revision causes a safety signature mismatch on the primary and secondary controllers.

After an online firmware update, it is your responsibility to validate the new safety application on the secondary controller before switching control to the secondary controller.

Validation During Safety Signature Generation

When you generate a safety signature on a synchronized controller, both controllers in the redundant chassis pair generate safety signatures and cross-check to validate that the safety signatures match.

Validation During Redundant Chassis Qualification

If a safety signature exists in the primary controller during qualification, the safety signature undergoes the following validation process:

1. The unqualified primary controller transfers its safety signature to the unqualified secondary controller.
2. The unqualified secondary controller independently validates the safety signature that is received from the unqualified primary controller.
3. If the unqualified secondary controller cannot validate the received safety signature independently, then the qualification process ends and neither controller becomes qualified. Qualification is aborted and the primary controller continues to operate without a qualified secondary controller.

Validation During Safety Application Download

If a safety signature exists in a safety application that you download to a qualified primary controller, the safety signature undergoes the following process:

1. The qualified primary controller attempts to validate the safety signature:
   - If the primary controller validates the safety signature, it downloads the safety application and then transfers the safety signature to the secondary controller.
   - If the primary controller cannot validate the safety signature, then it does not download the safety application. The controller remains in PROG mode with no application.
2. When the qualified secondary controller receives a safety signature from the qualified primary controller, it attempts to validate the safety signature independently:
   - If the secondary controller successfully validates the safety signature, then it downloads the safety application.
   - If the secondary controller cannot validate the safety signature, then the safety application remains downloaded only on the primary controller and does not download to the secondary controller. The primary controller remains in PROG mode.

IMPORTANT: If a safety signature fails validation, then neither standard nor safety logic within the safety application downloads.

Safety I/O

Safety I/O has most of the attributes of standard I/O except it features mechanisms that are certified to SIL 2 or SIL 3 for data integrity.

Before you use safety I/O, do the following:

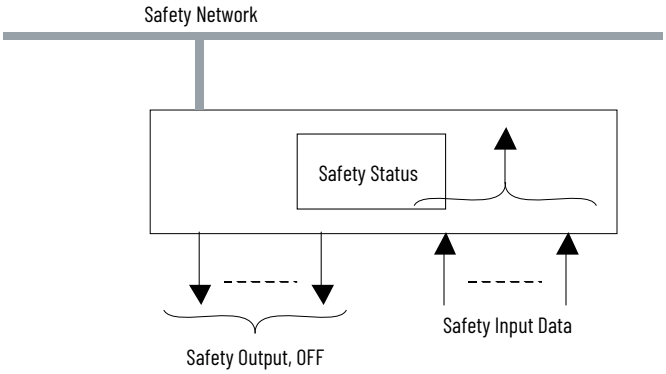
- Read, understand, and follow all safety information in the product documentation for those products.
- Commission all devices with a node or IP address and communication rate before their installation on a safety network.

Safety I/O devices, like sensors and actuators, can be connected to safety input and output modules. The controller monitors and controls the devices. For safety data, I/O communication is performed through safety connections by using the CIP Safety™ protocol. Safety logic is processed in the controller.

Safety I/O treats the following as the safe state:

- Safety outputs: OFF
- Safety input data to controller: OFF

Figure 12. Safe State



Use safety I/O devices for applications that are in the safe state when the safety output turns OFF.

Diagnostics

Safety I/O devices perform self-diagnostics when the power is turned ON and periodically during operation. If a diagnostic failure is detected, safety input data to the controller and local safety outputs are set to their safe state (OFF).

Status Data

Safety I/O devices support status data to monitor device and I/O circuit health. For specific product capabilities, see the product documentation for your device.

Status Indicators

Safety I/O devices include status indicators. For details on status indicator operation, see the product documentation for your specific device.

On-delay or Off-delay Function

Some safety I/O devices support on-delay and off-delay functions for input signals:

- Safety inputs can require an on-to-off delay to filter out the low pulse test in an output signal switching device (OSSD). Though the pulse test duration is measured in microseconds, the safety inputs can detect the low pulse as a transition to the safe state. The smallest configurable millisecond delay can be enough to filter out the pulse test.
- An on-to-off delay filter can help to filter out noise that affects the input logic level.
- Be sure to count any configured delays into the system reaction time.

SIL 2 and SIL 3 Considerations

The following information is only typical and may not be required for all safety I/O that can be used with safety controllers. For information specific to your I/O product, see the product user manual.

A difference between the safety integrity levels is that single-channel I/O devices are possible for SIL 2, and dual-channel I/O devices are typically required for SIL 3.

From a safety architecture perspective, one channel means that the hardware fault tolerance (HFT) is zero. When the HFT is zero, there are guidelines that state that faults must be detected and the safety function must be taken to a safe state within the process safety time. An exception applies if the diagnostic test rate is 100 times the demand rate. If you use safety I/O modules in single channel SIL 2 applications, consider the following:

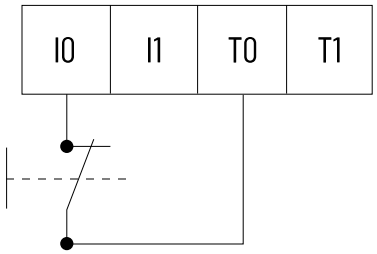
- The input or output channel must be configured for Safety Pulse Test
- A process safety time greater than 600 ms (the typical safety I/O pulse test interval) or the demand rate must be less than one demand per minute, such as one per hour

Digital safety input modules support single-channel SIL 2 and dual-channel SIL 3 safety input circuits. Because these modules are rated for both SIL 2 and SIL 3 operation, you can mix SIL 2 and SIL 3 circuits on the same module.

The following figure shows how to wire SIL 2 safety circuits to safety input modules.

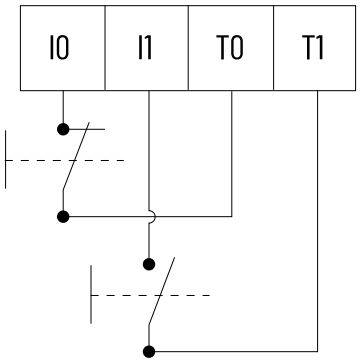
IMPORTANT: The test source must be configured for pulse testing.

Figure 13. Input Wiring



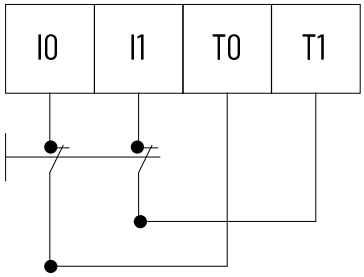
If you have two SIL 2 safety circuits, you can add a second as shown below.

Figure 14. Input Wiring Pairs



A typical SIL 3 wiring diagram is shown below.

Figure 15. SIL 3 Input Wiring



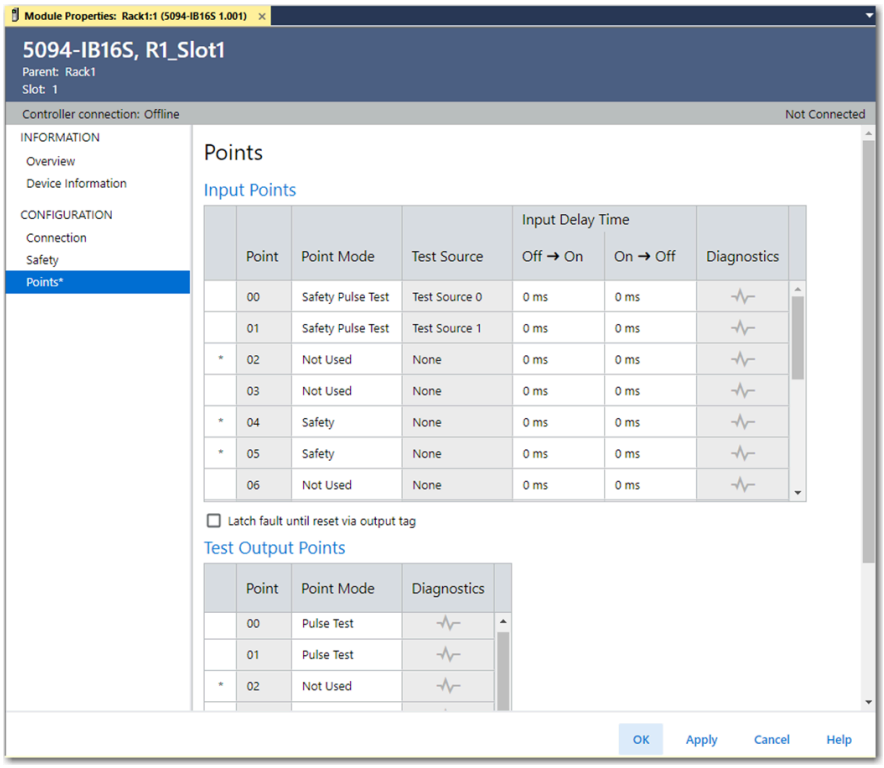
IMPORTANT: These wiring drawings are examples of possible wiring configurations. Depending on your I/O device and system configuration, other wiring configurations can also be used.

IMPORTANT: The onboard pulse test outputs (T0...Tx) are typically used with field devices that have mechanical contacts. If a safety device that has electronic outputs is used (to feed safety inputs), they must have the appropriate safety ratings.

Input Operation

FLEX 5000® safety input modules have single-channel input points.

Figure 16. Safety Module Input Points



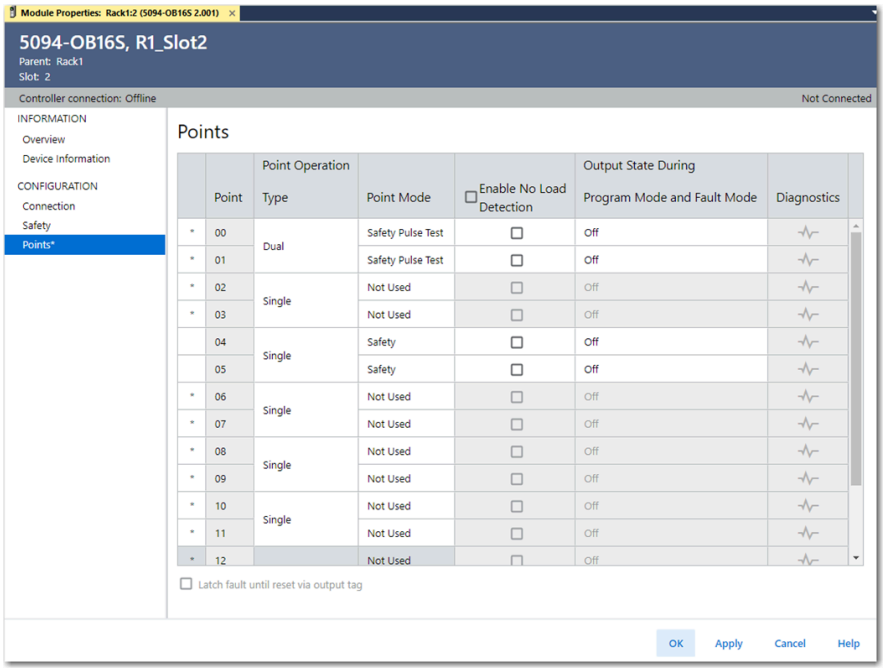
Output Operation

FLEX 5000® safety output modules can be configured with single or dual point operation types:

- A single point operation type allows the outputs to turn on and off individually and to fault independently.
- A dual point operation type verifies that safety task logic operates both outputs as a pair. If one output has a module fault, the other output goes to the safe state.

IMPORTANT: The point operation type affects the safety rating of the module.

Figure 17. Output Points Operation Types



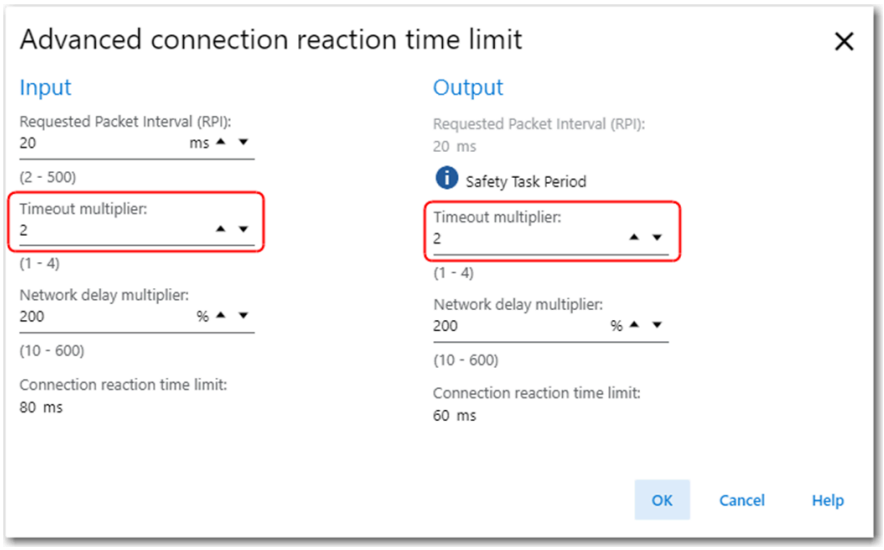
Recommended Safety I/O Settings

We recommend the following settings for safety I/O:

- Timeout multiplier—Use a minimum safety timeout multiplier of 2 on both input and output connections. This recommendation helps to make sure that timeouts do not occur during a loss of redundancy.
- Network delay multiplier—The network delay multiplier represents the transport time of a message across the communication network. You can tune this value to your application, but we recommend that you test to make sure that the I/O connections survive a loss of redundancy. Use a minimum value of 200, which means 200%.

IMPORTANT: To determine the appropriate values, analyze each safety channel. The default Timeout Multiplier of 2 and Network Delay Multiplier of 200 creates a worse-case input connection reaction time limit of 4 times the RPI, and an output connection reaction time limit of 3 times the RPI. Changes to these parameters must be approved only after a thorough review by a safety administrator.

Figure 18. Safety I/O Settings



Safety I/O Configuration Signature

The safety I/O configuration signature is a number that uniquely identifies the configuration of a device and verifies that the device is configured as expected by the safety application. The configuration signature consists of the following:

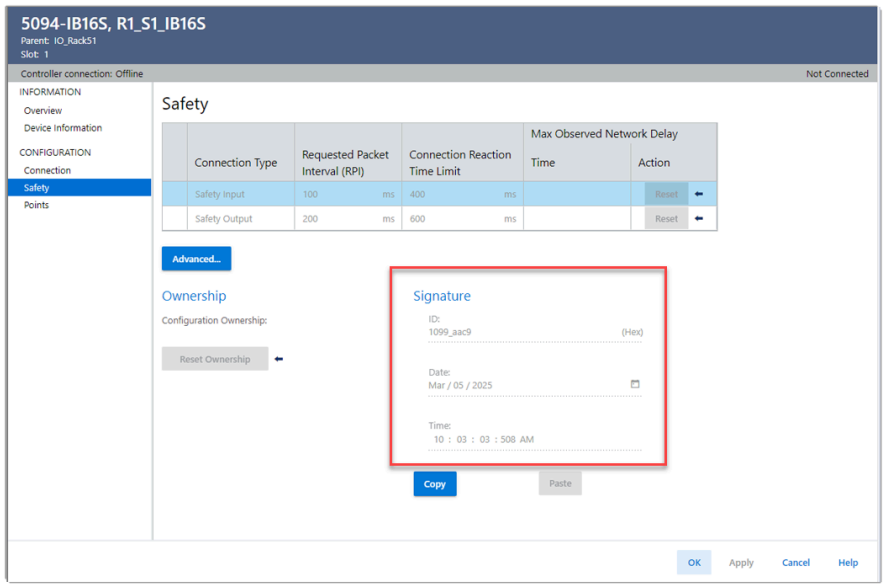
- A signature ID that represents the I/O module configuration
- The time and date that the module configuration was last applied

IMPORTANT:

The safety I/O configuration signature applies to individual safety modules.

The safety I/O configuration signature is different than the controller safety signature, which applies to the entire safety portion of the controller.

Figure 19. Safety I/O Configuration Signature



For a controller to establish a connection to a safety I/O module, the configuration signature in the controller must match the configuration signature in the safety I/O module. The process of synchronizing the configuration signatures requires these steps:

1. Create a safety I/O module in a Logix Designer application project.
2. Configure the I/O module in the module profile.
3. Download the project to the controller.

Online changes to the module configuration change the configuration signature. When online changes are applied, the controller downloads the configuration to the I/O module.

Offline changes to the I/O module configuration change the time and date. Once altered, the time and date remain changed even if the configuration is returned to the current running configuration. Offline changes to the time and date require one of these actions:

- Upload to keep the existing configuration.
- Download to push the new configuration to the I/O modules.

If a safety I/O module was previously configured in another location, the I/O module retains the configuration signature from the previous location. When a controller and a safety I/O module attempt to establish a safety connection, a mismatch of the configuration signatures can cause the connection to fail. To clear the safety I/O module configuration and enable the controller to download the module configuration to the safety I/O module, you must reset ownership.

The controller verifies that configuration signatures match, so there is no requirement to monitor or document the configuration signature. If the configuration signature changes unexpectedly, the safety connection between the controller and I/O module fails and causes the I/O module to enter its safe state.

When using a third-party module, if you connect to a safety I/O device without a configuration signature, you must verify that a valid configuration exists in the safety I/O device.

IMPORTANT: Safety I/O modules default to using a configuration signature and do not allow your system to run without a configuration signature.

Safety I/O Device Replacement

The process for making a connection to replacement safety I/O is dependent upon multiple factors including the following:

- Node address
- Electronic keying compatibility
- Whether the I/O module configuration is empty (out-of-box) or previously configured
- The Automatic Configuration setting for the safety controller

ATTENTION: During replacement or functional testing of a device, the safety of the system must not rely on any portion of the affected device.

The electronic keying configuration affects the process for replacing safety I/O modules. Carefully consider the implications of each of the following electronic keying options.

Table 6. Electronic Keying Settings

Keying Setting	Description	I/O Replacement Consideration
Compatible Module	Lets the installed device accept the key of the device that is defined in the project when the installed device can emulate the defined device. With Compatible Module, you can typically replace a device with another device that has the following characteristics: • Same catalog number • Same or higher major revision • Minor revision as follows: - If the major revision is the same, the minor revision must be the same or higher. - If the major revision is higher, the minor revision can be any number.	To maintain the safety signature, the replacement module must meet Compatible Module requirements.
Disable Keying	Indicates that the keying attributes are not considered when attempting to communicate with a device. With Disable Keying, communication can occur with a device other than the type specified in the project. **ATTENTION:** Be cautious when using Disable Keying. If used incorrectly, this option can lead to personal injury or death, property damage, or economic loss. We **strongly recommend** that you **do not use** Disable Keying. If you use Disable Keying, you must take full responsibility for understanding whether the device being used can fulfill the functional requirements of the application.	Many safety devices do not have a Disable Keying option. Disabled Keying is not recommended for safety applications.
Exact Match	Indicates that all keying attributes must match to establish communication. If any attribute does not match precisely, communication with the device does not occur.	• To maintain the safety signature, the replacement module must be Exact Match. • After a firmware change, keying in the safety application must be updated. Updating cannot

Table 6. Electronic Keying Settings (continued)

Keying Setting	Description	I/O Replacement Consideration
		be done without removing the controller safety signature. • Exact Match is often used to meet specific industry requirements.

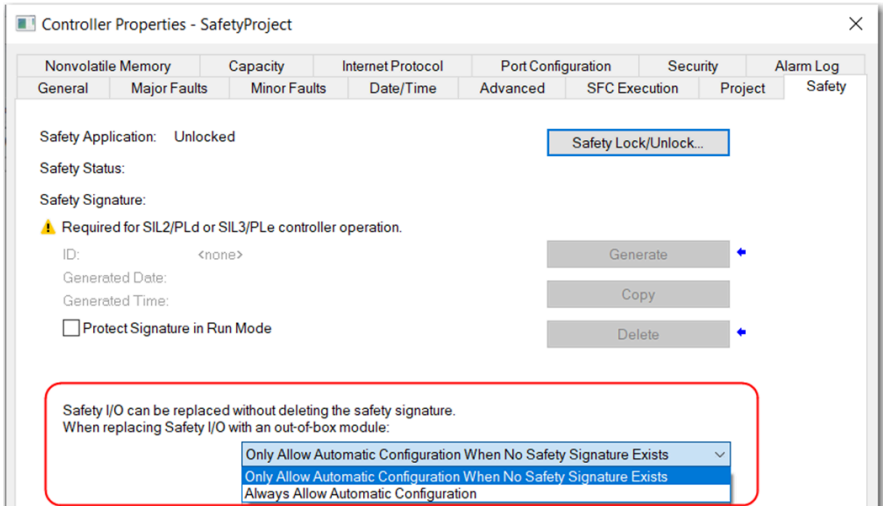
Automatic configuration enables the safety controller to establish a connection with replacement safety I/O without online user interaction. Automatic configuration is enabled when there is no controller safety signature. Allowing automatic configuration with no safety signature facilitates commissioning and maintenance activities when the safety controller is not being used for SIL-rated functions. Following I/O replacement, you must validate proper operation before using SIL-rated functions.

Two options for I/O device replacement are available on the Safety tab of the Controller Properties dialog box in the Studio 5000 Logix Designer® application:

- Only Allow Automatic Configuration When No Safety Signature Exists
- Always Allow Automatic Configuration

Choosing an automatic configuration setting requires an understanding of the safety network topology and the intended use of the safety system during I/O replacement.

Figure 20. Safety I/O Replacement Options



Only Allow Automatic Configuration When No Safety Signature Exists

This option instructs the safety controller to configure a safety device when the safety task does not have a safety signature, and the replacement device is in an out-of-box condition with no safety network number.

If the controller has a safety signature, the safety controller automatically configures the replacement safety I/O device if the following are true:

- The device already has the correct safety network number.
- The device electronic keying is correct.
- The node or IP address is correct.

To set the proper safety network number (SNN) when a controller safety signature exists, manual action is required to download the proper SNN:

1. In the Studio 5000 Logix Designer®, go online with the safety controller.
2. Open the Module Properties dialog box.
3. On the General tab, click Browse (...) next to the safety network number.
4. Click Set to write the SNN to the module manually.
5. Verify that the Network Status (NS) indicator is alternating red/green on the correct device.
6. Click Yes on the confirmation dialog box to set the SNN and accept the replacement device.
7. Follow your company-prescribed procedures to functionally test the replacement I/O device and system.

For more information, follow the safety I/O device replacement procedure in the controller user manual.

Always Allow Automatic Configuration

The controller attempts to configure a replacement safety I/O device automatically if the device is in an out-of-box condition. When a safety network number does not exist in the replacement safety device, and the node number and I/O device keying matches the configuration of the controller.

ATTENTION:

Select the Always Allow Automatic Configuration option only if the entire routable safety control system is not being relied on to maintain SIL 2 or SIL 3 behavior during the replacement and functional testing of a device.The routable safety control system includes any device that can have safety connections opened on it by the controller.

If other parts of the safety control system are being relied upon to maintain SIL 2 or SIL 3, make sure that the Always Allow Automatic Configuration option is not selected.

It is your responsibility to implement a process to make sure that proper safety functionality is maintained during device replacement.

ATTENTION: To place a device in the out-of-box condition on a safety network when the Always Allow Automatic Configuration option is selected, follow the device replacement procedure in the controller user manual.

Automatic Configuration Use Cases

Consider the following examples of when to use a particular automatic configuration setting.

Only Allow Automatic Configuration When No Safety Signature Exists	Always Allow Automatic Configuration

• Multi-zone safety system where I/O replacement is required in one zone, while other zones maintain SIL functionality.
• Multiple controllers with safety I/O on the same routable network.
	• The safety system provides no safety function during I/O replacement:
	  - Use of energy isolation or other application-specific procedure
	  - Other application-specific safeguards are in place
	• The network for safety I/O communication is isolated from external safety devices.

CIP Safety Systems and Safety Network Numbers

CIP Safety™ control systems are composed of CIP Safety™ devices that are interconnected via communication networks. These networks consist of devices, such as bridges, switches, routers, adapters, and redundancy modules, which may not be SIL 2 or SIL 3 certified. The secure transmission of safety-related data through standard infrastructure devices is known as black channel communication.

CIP Safety™ devices must be inherently protected from network delivery errors. The CIP Safety™ protocol is an end-node to end-node safety protocol. This configuration allows the routing of CIP Safety™ messages to and from CIP Safety™ devices through non-certified bridges, switches, routers, adapters, and redundancy modules.

For more information about CIP Safety™ functionality and related concepts, see the following:

- ODVA website at https://www.odva.org
- CIP Safety: Safety Networking for Today and Beyond White Paper, publication SAFETY-WP038

Unique Node Reference

A key element of the CIP Safety™ protocol is the concept of a Unique Node Reference (also called Unique Node ID or UNID). Every CIP Safety™ device must have a UNID value that is assigned to each CIP Safety™-capable port.

IMPORTANT: It is your responsibility to make sure that all UNIDs are unique within the scope of all devices that could possibly communicate with each other.

Safety Network Numbers (SNNs)

Communication within a control system travels over subnets that are interconnected with bridging or routing components. The following are examples of subnets:

- The backplane of a chassis
- A bank of I/O modules
- An Ethernet subnet within a LAN

Rather than creating a UNID directly for each CIP Safety™ device, which can be prone to error in a large system, each subnet has a unique safety network number (SNN), and the UNID is created from the SNN + the node address. In Logix SIS, an SNN applies to an entire redundant chassis pair.

How SNNs Get to Safety Devices

Most CIP Safety™ I/O modules in the factory default state accept an SNN that is assigned by the controller that owns that module. The SNN value that the programming software automatically adopts for the connection of that module is accepted when the controller opens the initial connection to the module.

IMPORTANT: CIP Safety™ I/O modules retain their UNID (SNN + node) once it has been assigned and must be reset before they can be reused with another value.

Some devices, such as another safety controller in the I/O tree, receive their SNN configuration from a programming workstation. For these devices, you must manually configure the connection to use the same SNN that has been programmed into that device if the programming software did not automatically assign the correct SNN.

SNN Formats

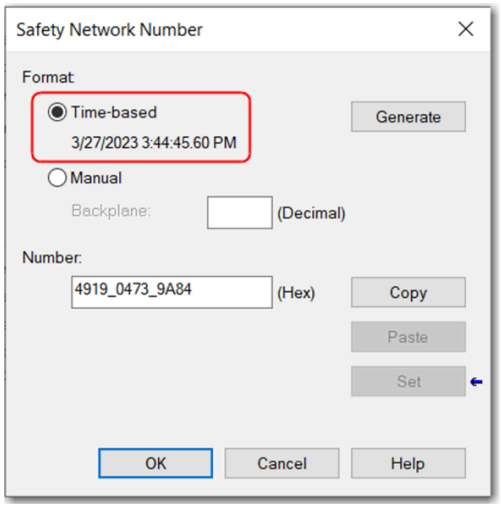
SNNs used by the system are 6-byte hexadecimal numbers. SNNs can be set and viewed in one of two formats:

- Time-based
- Manual

Time-based SNN Format and Assignment

When the time-based format is selected, the SNN represents a date and time based on the local clock of the computer that generated the SNN.

Figure 21. Time-based SNN Format



The assignment of time-based SNNs is automatic when you create a safety project or add EtherNet/IP™ by changing the IP mode or controller type. Time-based SNNs generated by the software are always unique to the project, whether generated by project creation or IP mode change. Devices that are created directly under the controller port default to having the same SNN as that port on the controller.

IMPORTANT: If you have a network diagram for your application, you must edit the SNNs of the controller to match your network diagram. We recommend that you edit the SNNs before you add devices to the I/O configuration in Controller Organizer.

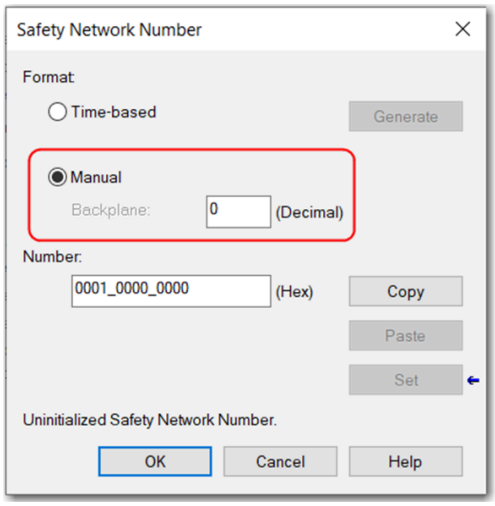
When you add CIP Safety™ I/O devices to ports under an adapter, as opposed to the controller, the following applies:

- If no other device under the port uses an SNN, a time-based SNN is automatically assigned.
- If another device under the port uses an SNN, the device is assigned the same SNN as the first device in address order.

Manual SNN Format and Assignment

When the manual format is selected, the SNN represents a network type and must have a decimal value from 1...9999.

Figure 22. Manual SNN Format



Manual manipulation of an SNN is required for the following reasons:

- To make sure that each safety controller port on the same subnet has the same SNN in all projects
- To copy safety projects

ATTENTION: If a safety project is copied into another project with different hardware or in another physical location, and the new project is within the same routable safety system, every SNN must be changed in the second system. SNN values cannot be repeated. For information about how to change the SNN, see the controller user manual.

IMPORTANT:

If you assign an SNN manually, make sure that system expansion does not result in a duplication of SNN and unique node reference combinations.

A warning appears if your project contains duplicate SNN and unique node reference combinations. You can still verify the project, but we recommend that you resolve the duplicate combinations.

However, there can be safety devices on the routable safety network that have the same SNN and node address and are not in the project. In this case, these safety devices are unknown to the Studio 5000 Logix Designer® application, and you may not see a warning.

If there are duplicate unique node references, as the system user, you are responsible for proving that an unsafe condition cannot result.

Safety Programming Considerations

Complete these programming tasks for safety applications:

- Define the location, ownership, and configuration of I/O devices and controllers.
- Create, test, and debug program logic. Only ladder diagram is supported in the safety task.

IMPORTANT: When the controller is in Run or Program mode and you have not validated the application program, you are responsible for maintaining safe conditions.

Programming Restrictions

The Logix Designer application limits the availability of some menu items and features, such as cut, paste, delete, and replace, to protect safety components from being modified whenever any of these are true:

- The controller is safety-locked
- A safety signature exists
- Safety faults are present

IMPORTANT: The maximum and last scan times of the safety task and safety programs can be reset when online.

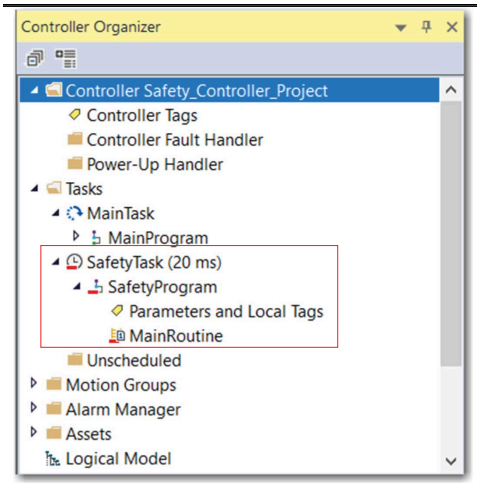
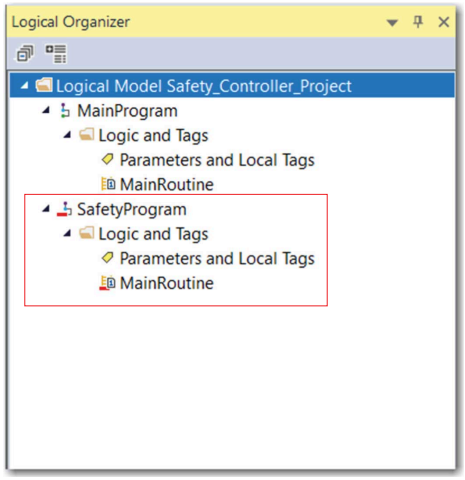
If even one of these conditions applies, you cannot do the following:

- Create or modify safety objects, including safety programs, safety routines, safety tags, safety Add-On Instructions, and safety I/O devices
- Apply forces to safety tags
- Create safety tag mappings
- Modify or delete tag mappings
- Modify or delete user-defined data types that are used by safety tags
- Modify the controller name, description, chassis type, slot, and safety network number
- Create, modify, or delete a safety connection

When the controller is safety-locked, you cannot modify or delete the safety signature.

Safety Task

A safety application includes a safety task with a safety program and main routine.

Safety Task in Controller Organizer	Safety Program in Logical Organizer
	

A safety task has these characteristics:

- The safety task is a periodic timed task. A periodic task is triggered at repetitive intervals. When triggered, the period task and its programs are executed. Data and outputs that the programs in the task establish retain their values until the next execution of the task or until another task manipulates them. Periodic tasks always interrupt the continuous task.
- The safety task must be the highest priority user task.
- The safety task function is temporarily muted during qualification and synchronization, loss of redundancy, and a lock for update. For more information, see Safety Function Muting on page 17.
- The safety task cannot be deleted.
- Within the safety task, you can use multiple safety programs that are composed of multiple safety routines.
- You cannot execute standard routines from within the safety task.
- There can be only one safety task for a controller.

Safety Task Parameters

You can configure the following safety task parameters:

- Period—The safety task period is the time interval between successive executions of the safety task. The safety task period directly affects system reaction time. You cannot edit the safety task period online.
- Priority—Logix SIS requires that the safety task is the highest priority task.
- Watchdog—The safety task watchdog is the maximum time that is allowed from the start of the safety task to its completion.

To configure the safety task, right-click the safety task and choose Properties.

IMPORTANT: The values that you define for the safety task parameters must meet the requirements that are defined in the following table.

Figure 23. Safety Task Parameters

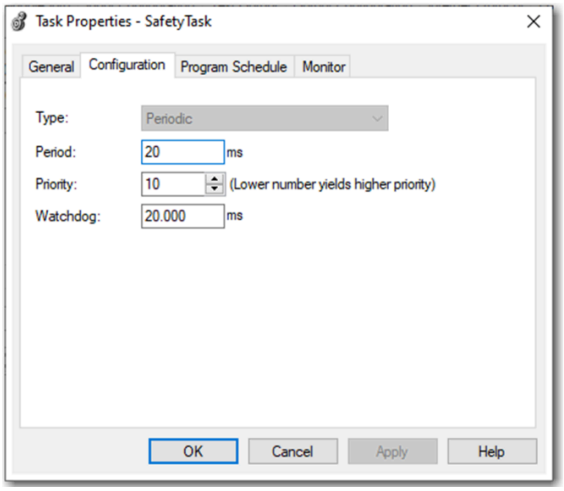


Table 7. Safety Task Requirements

Parameter	Requirement
Period	Be sure that the safety task has enough time to execute its logic before it restarts. Enter a value of 7...500 ms.
Priority	A value of 1 is enforced to make the safety task the highest priority task.
Watchdog	Enter a value of 5...500 ms. To avoid a fault that can shut down the safety task during a loss of redundancy, enter a value that is at least 5 ms higher than your safety task max scan time. For example, if your safety task max scan time is 5 ms, set your safety task watchdog to at least 10 ms. There is no fault to indicate that the safety task is consuming too much scan time.

IMPORTANT: When determining the safety task max scan time, we recommend that you perform several disqualifications and switchovers to get a more accurate time measurement that accounts for system dynamics.

Safety Task Execution Details

The safety task executes like standard periodic tasks with these exceptions:

- Safety input tags and safety-consumed tags are updated only at the beginning of safety task execution. This process means that even though the I/O RPI can be faster than the safety task period, the data in the Safety Input tag only updates once at the beginning of each safety task execution. Safety input and consumed packets that arrive after the start of the safety task are buffered until the next execution of the safety task.
- Time is frozen at the start of safety task execution. As a result, timer-related instructions, such as TON and TOF, are not updated during a safety-task execution. They keep accurate time from one task execution to another, but the accumulated time is not changed during safety task execution.

IMPORTANT: This behavior differs from standard Logix task execution.

- For standard tags that are mapped to safety tags, the standard tag values are copied to the safety tags at the start of the safety task:
  - The standard tag is free to continue changing.
  - Programming logic can change the safety tag within the safety task, but the change is not reflected back to the standard tag.

IMPORTANT: The addition of more mapped tags can increase the scan time.

- Safety output tag values can be changed during the safety task scan by the safety programming logic. The final value is transmitted to safety modules at the end of the safety task scan.

IMPORTANT:

While safety-unlocked and without a safety signature, the controller helps prevent simultaneous write access to safety memory from the safety task and communication commands. As a result, the safety task can be held off until a communication update completes. The time that is required for the update varies by tag size. Insufficient time can result in safety connection and safety watchdog timeouts.

To compensate for the hold-off time due to a communication update, you must increase the safety watchdog time.

Depending on the edit, a watchdog timeout can occur if there is insufficient time to complete the safety task operation.

When the controller is safety-locked or a safety signature exists, the scenarios that are described in this note cannot occur.

Safety Programs

A safety program has the attributes of a standard program, except that it can be scheduled only in the safety task:

- A safety program can also define program-scoped safety tags.
- A safety program can be scheduled or unscheduled.
- A safety program can contain only safety components.
- All routines in a safety program are safety routines. One safety routine must be designated as the main routine, and another safety routine can be designated as the fault routine.
- A safety program cannot contain standard routines or standard tags.
- For program parameters, a safety parameter cannot be connected with or bound to a standard parameter or controller-scoped tag.

Safety Routines

Safety routines have the same attributes of standard routines, except for the following:

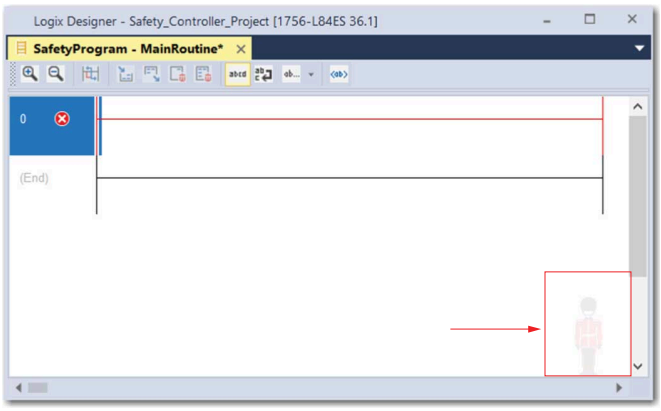
- Safety routines can exist only in safety programs.
- Safety routines cannot read or write standard tags.
- Safety routines can only be done in Ladder Logic.

One safety routine must be designated as the main routine in each safety program. Another safety routine can be designated as the fault routine for that safety program.

Only safety-certified instructions or Add-on Instructions composed from them are used in safety routines. For a list of instructions, see Safety Instructions on page 80.

A watermark visually distinguishes a safety routine from a standard routine.

Figure 24. Safety Routine Watermark



Safety Tags

The controller supports the use of both standard and safety tags in the same project. However, the programming software operationally differentiates standard tags from safety tags.

Safety tags have the same attributes as standard tags with the addition of mechanisms that are certified to provide SIL 2/PLd and SIL 3/PLe data integrity.

IMPORTANT: Logix SIS does not support produced and consumed safety tags.

When you create a tag, you assign the following properties:

- Name
- Description (optional)
- Tag type
- Data type
- Scope
- Class
- Style
- External Access
- If the tag value is a constant

The Studio 5000 Logix Designer® application helps prevent the direct creation of invalid tags in a safety program. If invalid tags are imported, they cannot be verified.

IMPORTANT: You cannot create a standard alias tag of a safety tag. Instead, standard tags can be mapped to safety tags using safety tag mapping. See Standard Tags in Safety Routines on page 46.

The Logix Designer application can write to safety tags directly via the Tag Monitor when the controller is safety-unlocked, does not have a safety signature, and is operating without safety faults.

The controller does not allow writes to safety tag data from external human machine interface (HMI) devices or via message instructions from peer controllers. HMI devices can have read-only access to safety tags depending on the External Access setting.

Valid Data Types

The data type defines the type of data that the tag stores, such as bit or integer.

Data types can be combined to form structures. A structure provides a unique data type that matches a specific need. Within a structure, each individual data type is called a member. Like

tags, members have a name and data type. You can create your own structures, such as arrays or user-defined data types.

Logix controllers contain predefined data types for use with specific instructions. Safety tags can be composed of the following:

- All primitive data types (for example, BOOL, SINT, INT, DINT, LINT, REAL)
- Predefined types used for safety application instructions
- User-defined data types or arrays that are composed of the two preceding types

Scope

The scope of a tag determines where you can access the tag data. When you create a tag, you define it as a controller tag (global data) or a program tag for a specific safety or standard program (local data).

Safety tags can be controller-scoped or safety program-scoped:

- Controller-scoped safety tags can be read by either standard or safety logic or external communication devices, but can be written by only safety logic.
- Program-scoped safety tags can be read by external communication devices, but only local safety routines within the safety program can write to them.

When you create program-scoped tags, the class is automatically specified, depending on whether you created the tag in a standard or a safety program. When you create controller-scoped tags, you must manually select the tag class.

When safety tags are controller-scoped, all programs have access to the safety data. Tags must be controller-scoped if they are used in the following ways:

- Multiple programs in the project
- In safety tag mapping

Controller-scoped safety tags can be read, but not written to, by standard routines.

IMPORTANT: Safety input tags are readable by any standard routine, but the update rate is based on the execution of the safety task. These tags are updated at the beginning of the safety task execution, which differs from standard tag behavior.

Safety Signature Elements

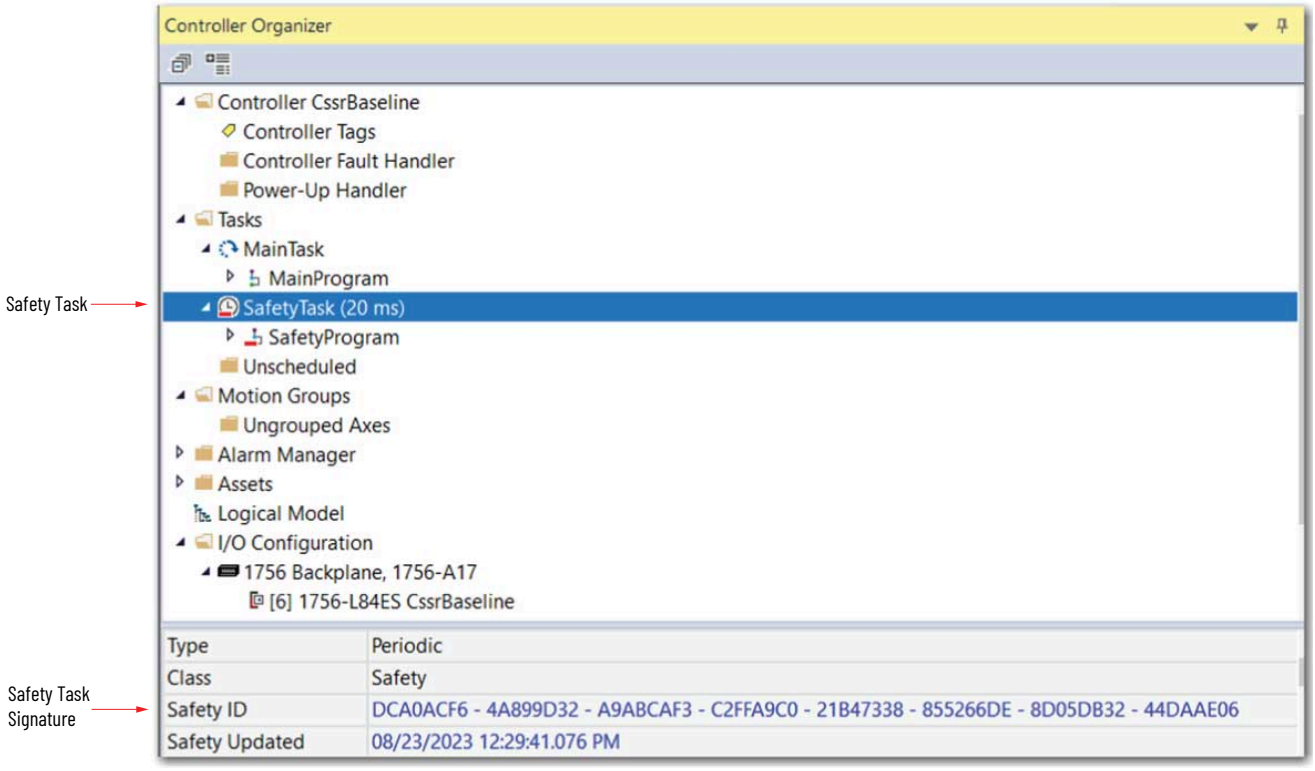
Safety signature elements include:

- Safety application
- Safety controller attributes
- Safety tags
- Tag mapping
- Safety task
- Safety programs
- Safety routines
- Safety add-on instructions
- Safety I/O device configuration

Each element has a safety signature. The signature changes when its associated element is modified and requires revalidation.

To view the safety signature for a safety tag, the safety task, or a safety program, select the element in the Controller Organizer. The safety signature ID and timestamp appear in the Quick View pane at the bottom of the Controller Organizer.

Figure 25. Safety Signature for Safety Task Element



Standard Tags in Safety Routines (Tag Mapping)

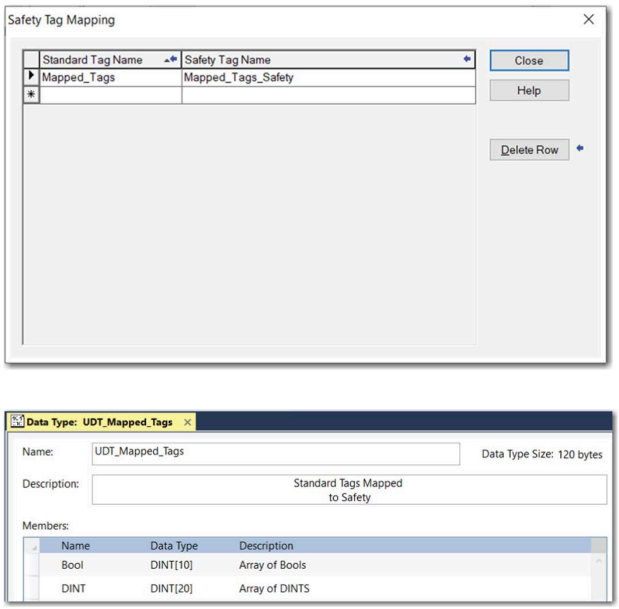
A safety routine cannot directly access standard tags. To allow standard tag data to be used within safety routines and synchronize standard and safety actions, safety controllers provide a safety tag-mapping feature that copies standard tag values into safety task memory.

Mapped tags are copied from the standard tags to their corresponding safety tags at the beginning of the safety task. The copying process can increase the safety task scan time.

Standard task routines can directly read safety tags.

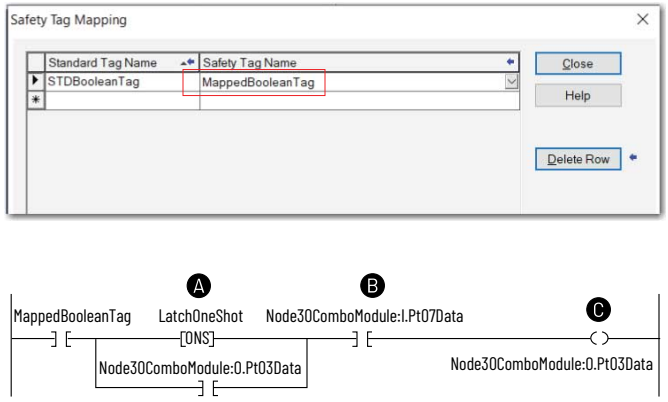
Because a download is required to change tag mapping, mapping a structure of information provides programming standardization and flexibility during commissioning.

Figure 26. Mapped Tag Structure Example



The following example shows how to qualify standard data with safety data.

Figure 27. Qualify Standard Data with Safety Data



Item	Description
A	Latch circuit to help prevent automatic restart if the standard input (MappedTag) is failed in a 'stuck at 1' state.
B	Safety input qualifier for mapped tag
C	Safety output

Restrictions

Safety tag mapping has these restrictions:

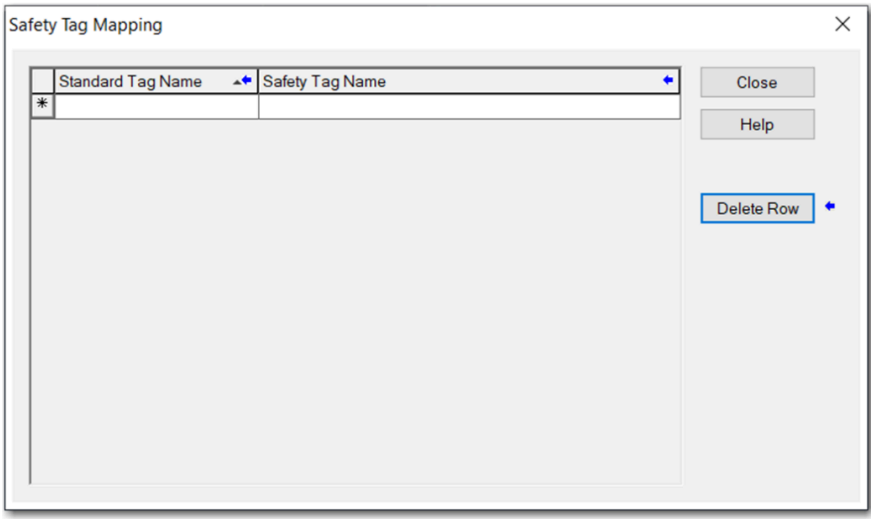
- The safety tag and standard tag pair must be controller-scoped.
- The data types of the safety and standard tag pair must match.
- Alias tags are not allowed.
- Mapping must take place at the whole tag level. For example, myTimer.pre is not allowed if myTimer is a TIMER tag.

- A mapping pair is one standard tag that is mapped to one safety tag.
- You cannot map a standard tag to a safety tag that has been designated as a constant.
- Tag mapping cannot be modified when any of the following are true:
  - The project is safety-locked.
  - A safety signature exists.
  - The controller switch is in RUN position.
  - A nonrecoverable safety fault exists.

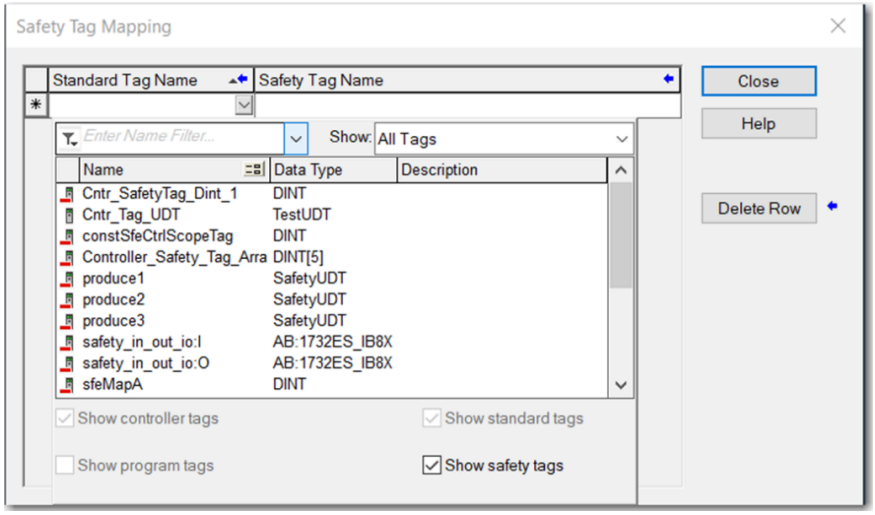
ATTENTION: If you use standard data in a safety routine, you must verify that the data is used in an appropriate manner. Standard data in a safety tag is not safety data. Do not directly control a SIL 2 or SIL 3 safety function by using standard data.

Create Tag Mapping Pairs

1. From the Logic menu, select Map Safety Tags.



2. In the Standard Tag Name or Safety Tag Name column, enter or select an existing tag:
   - To show only controller-scoped standard tags, click the arrow in the Standard Tag Name column.
   - To show only controller-scoped safety tags, click the arrow in the Safety Tag Name column.



3. In the Standard Tag Name or Safety Tag Name column, add a new tag:
   a. Right-click in the empty cell.
   b. Select New Tag.
   c. Enter the tag name into the field.
4. Right-click the field and select New tagname, where tagname is the text you entered in the field.

Custom Tag Initialization During Prescan

Only safety tags that are configured as constant value tags are captured as part of the safety signature.

IMPORTANT: When you use non-constant safety tag values for a safety critical operation, you must initialize the non-constant safety tags before Run mode.

Give special consideration to instructions that use pseudo-operands, such as the following:

- .PRE value for TON, TOF, RTO, CTD, and CTU
- .LEN value for FAL and FSC

Unless modified by the application, pseudo-operands are initialized only once when the application is downloaded. For details, see the 'Pseudo-operand Initialization' online Help topic. Before the controller is in Run mode, you must initialize the .PRE and .LEN values for the preceding instruction tags and other non-constant safety tags that are used in a safety critical operation. Initialize these values by using one of these methods:

- A first scan subroutine
- An Add-On Instruction prescan routine

For more information about how to perform a custom tag initialization during prescan, see the Logix 5000 Controllers Design Considerations Reference Manual, publication 1756-RM094.

You can use the SaveSnapshot routine to copy non-constant safety tag values to the safetyPrescanInitUDT backup, which consists of safety tag types, such as CTU preset, FAL length, TON preset, DINT array, REAL, and BOOL. For example, once the application is downloaded and configured, toggle the saveSnapshot tag to call the SaveSnapshot routine to initialize safetyPrescanInitUDT. Upon subsequent transitions to RUN mode, the prescan routine of safetyPrescanInitAOI reinitializes the non-constant safety tag values from the safetyPrescanInitUDT backup.

Figure 28. Main Routine

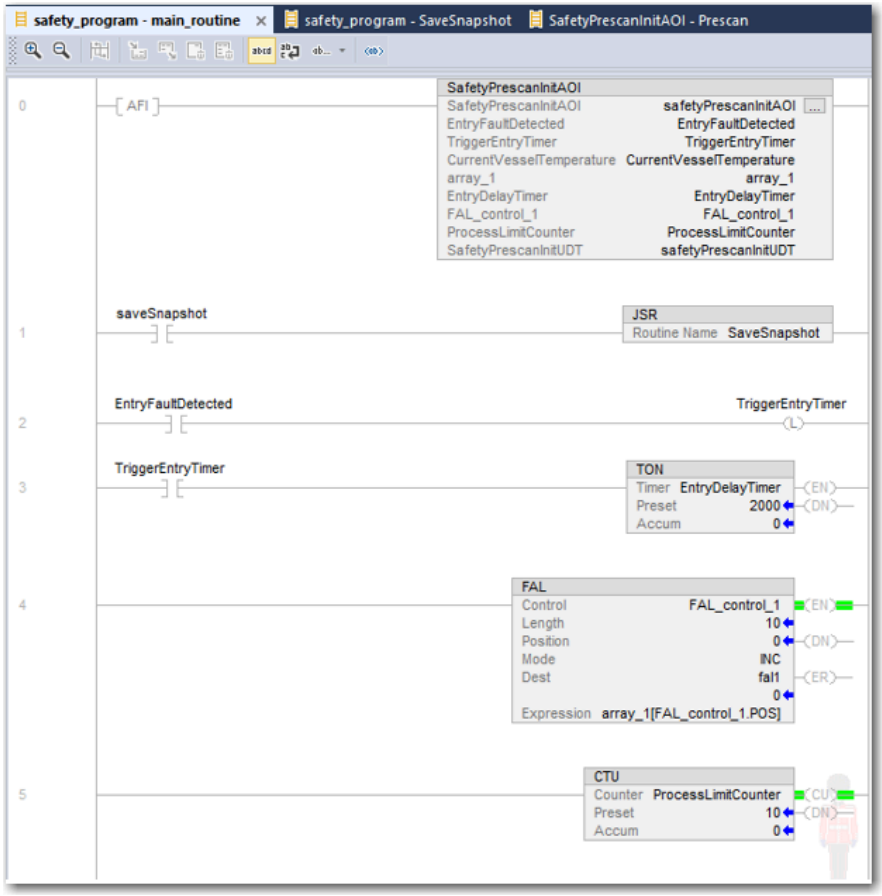


Figure 29. SaveSnapshot Routine

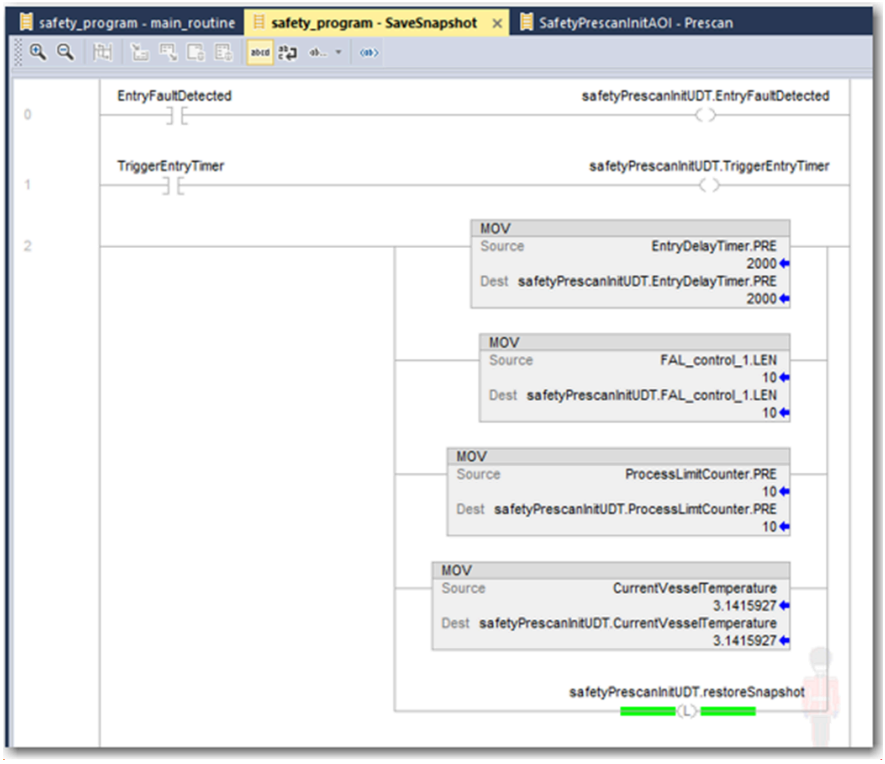
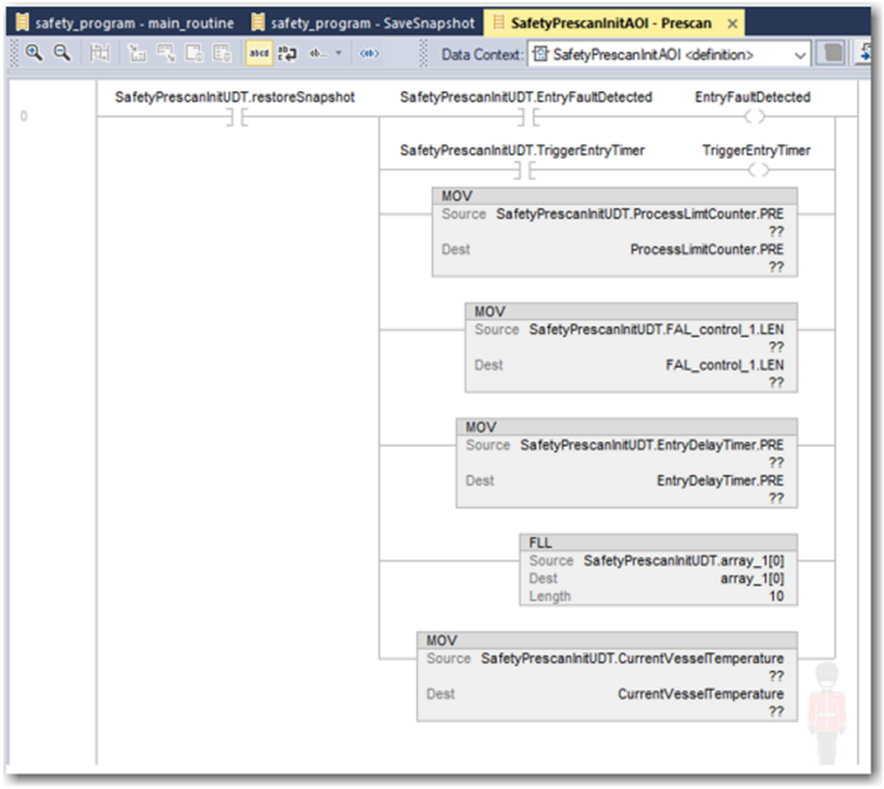


Figure 30. Add-On Instruction Prescan Initialization Routine



Safety Add-On Instructions

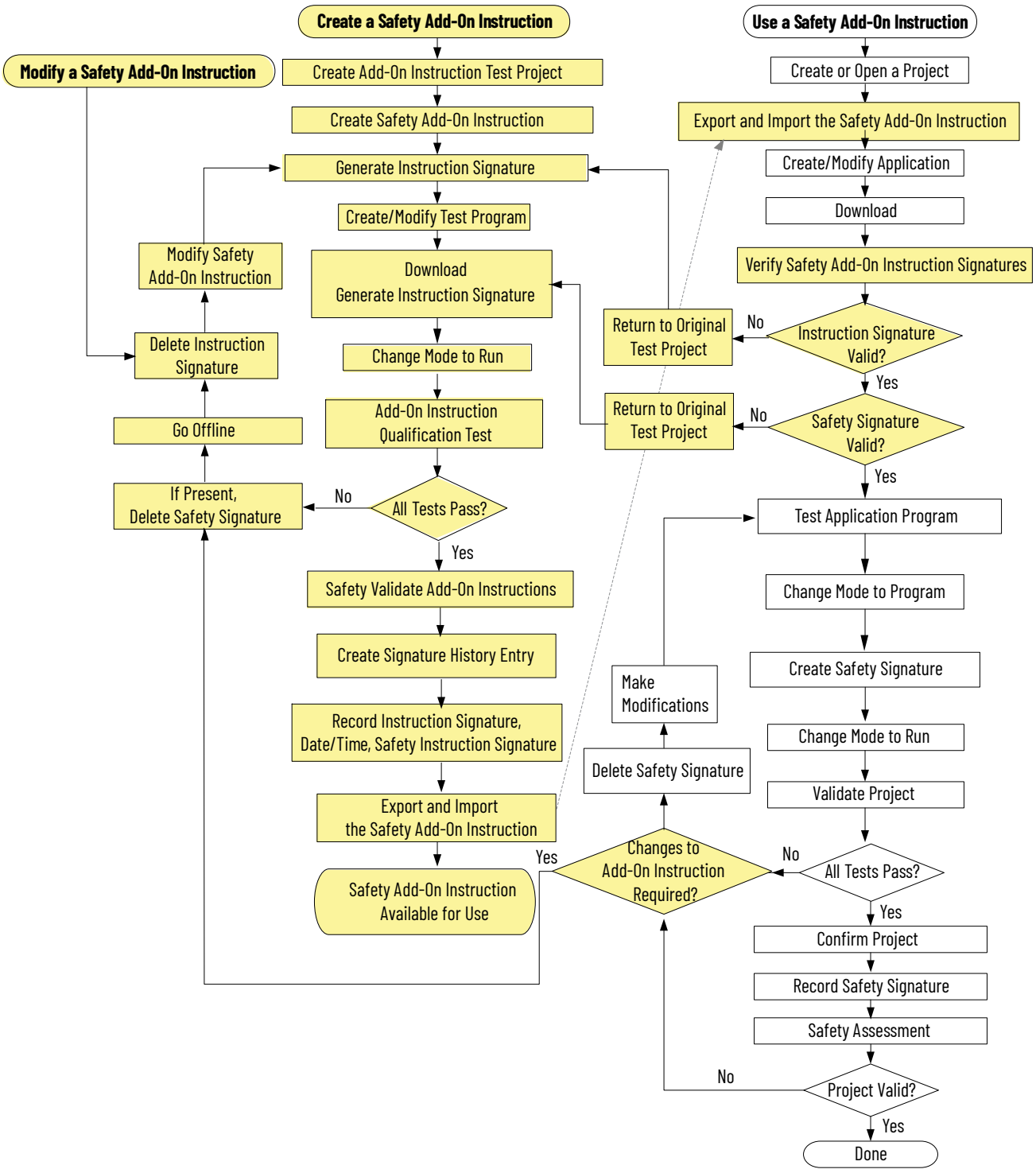
You can create safety Add-On Instructions to be used in safety applications. Safety Add-On Instructions enable you to put commonly used safety logic into one instruction, which makes it modular and easier to reuse. An Add-On Instruction is composed of parameters, local tags, logic routine, and optional scan-mode routines.

Safety Add-On Instructions feature a safety instruction signature for use in safety-related applications up to and including SIL 3-rated applications. The safety instruction signature is an ID number that identifies the execution characteristics of the safety Add-On Instruction. The signature is used to verify the integrity of the safety Add- On Instruction during downloads to the controller.

IMPORTANT: You must initialize safety critical Add-On Instruction tag values in the Add-On Instruction prescan logic.

The following figure shows the steps that are required to create a safety Add-On Instruction and then use that instruction in a safety application program. The shaded items are steps unique to Add-On Instructions.

Figure 31. Safety Add-On Instruction Workflow



Create an Add-On Instruction Test Project

You must create a unique test project to create and test the safety Add-On Instruction. This project must be a separate and dedicated project to minimize any unexpected influences. Follow the guidelines for projects that are described in Create the Project on page 60.

Create a Safety Add-On Instruction

For guidance in how to create Add-On Instructions, see the Logix 5000 Controllers Add-On Instruction Programming Manual, publication 1756-PM010.

Generate the Instruction Signature

The instruction signature consists of an ID number and time stamp that identifies the contents of the Add-On Instruction at a given point in time. The source of the date and time is from the local clock on the computer generating the signature.

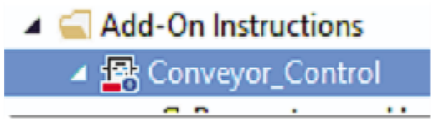
The instruction signature enables you to quickly determine if the instruction has been modified. Each Add-On Instruction can have its own signature. The instruction signature is required when an Add-On Instruction is used in safety-related functions, and can sometimes be required for regulated industries. Use it when your application calls for a higher level of integrity.

Once generated, the instruction signature seals the Add-On Instruction, which helps prevent it from being edited while the signature is in place. This restriction includes rung comments, tag descriptions, and any instruction documentation that was created. When the instruction is sealed, you can perform only these actions:

- Copy the instruction signature
- Create or copy a signature history entry
- Create instances of the Add-On Instruction
- Download the instruction
- Remove the instruction signature
- Print reports

When you generate an instruction signature, the Studio 5000 Logix Designer® application displays the instruction definition with the seal icon.

Figure 32. Add-On Instruction Icon



IMPORTANT: If you protect your Add-On Instruction with the source protection feature in the Studio 5000 Logix Designer© application, enable source protection before you generate the instruction signature.

When you generate a safety signature for a controller project, the Quick View Pane and Safety Signature Report show a safety signature element for Add-On Instructions even if the instruction signature is not generated. For more information about safety signature elements, see Safety Signature on page 19.

When a sealed safety Add-On Instruction is downloaded for the first time, a safety instruction signature is automatically generated.

IMPORTANT: Checking or clearing the Report Overflow Faults checkbox in the controller properties changes the safety instruction signature ID for safety Add-On Instructions that include math instructions.

Safety Add-On Instruction Qualification Tests

Safety Add-On Instruction tests must be performed in a separate, dedicated application to verify that unintended influences are minimized. You must follow a well-designed test plan and

perform a unit test of the safety Add-On Instruction that exercises all possible execution paths through the logic, including the valid and invalid ranges of all input parameters.

An independent, third-party review of the safety Add-On Instruction can be required before the instruction is approved for use. An independent, third-party validation may be required for functional safety certification.

Create Signature History Entry

The signature history provides a record for future reference. A signature history entry consists of the instruction signature, the name of the user, the time stamp value, and a user-defined description. Up to six history entries can be stored. You must be offline to create a signature history entry.

The Signature Listing report in the Studio 5000 Logix Designer® application prints the instruction signature, the time stamp, and the safety instruction signature. To print the report, right-click Add-On Instruction in the Controller Organizer and choose Print > Signature Listing.

Export and Import the Safety Add-On Instruction

When you export a safety Add-On Instruction, choose the option to include all referenced Add-On Instructions and user-defined data types in the same export file. By including referenced Add-On Instructions, you make it easier to preserve the signatures. When importing Add-On Instructions, consider these guidelines:

- You cannot import a safety Add-On Instruction into a standard controller project.
- You cannot import a safety Add-On Instruction into a safety controller project that has been safety-locked or one that has a safety signature.
- You cannot import a safety Add-On Instruction while online.
- If you import an Add-On Instruction with an instruction signature into a project where referenced Add-On Instructions or user-defined data types are not available, you may need to remove the signature.

For more information, see the Import/Export Project Components Programming Manual, publication 1756-PM019.

Qualification and Verification

After you download the application project that contains the imported safety Add-On Instruction, you must compare the instruction signature value, the date and time stamp, and the safety instruction signature values with the original values you recorded before you exported the safety Add-On Instruction. If they match, the safety Add-On Instruction is valid and you can continue with the validation of your application.

Next, test the application program. This step consists of any combination of Run and Program mode, online or offline program edits, upload and download, and informal testing that is required to get an application to run properly.

To validate the project, perform an engineering test of the application, including the safety system. For more information about requirements, see Validate the Project on page 62.

An independent, third-party review of the safety system can be required before the system is approved for operation. An independent, third-party validation may be required for functional safety certification.

Safety Applications

The safety concept assumes the following requirements:

- You are responsible to create, operate, and maintain the safety application.
- You are fully qualified, specially trained, and experienced in safety systems.
- You apply the logic correctly to detect programming errors through strict adherence to specifications, programming, and naming rules.
- You perform a critical analysis of the application and use all possible measures to detect a failure.
- You confirm all application downloads via a manual check of the safety signature.
- You perform a complete functional test of the entire system before the operational startup of a safety-related system. This test includes, but is not limited to, the following:
  - Validate that the overall functionality of the implemented safety functions, including I/O configuration via Add-On Profiles, beyond the limits of the individual devices (boundary testing).
  - Verify that the correct versions of software are used.

Table 8. Effect of Controller Modes on Safety Execution

Controller Mode	Controller Safety Execution
Program	• Safety input and output connections are established and maintained. Safety input tags are updated to reflect safety input values. • Safety mapped tags are updated to reflect the standard controller tag values. • Safety task logic is not being scanned.
Test	• Safety input and output connections are established and maintained. Safety input tags are updated to reflect safety input values. • Safety mapped tags are updated to reflect the standard controller tag values. • Safety task logic is being scanned.

Run	• Safety input and output connections are established and maintained:
	  - Safety input tags are updated to reflect safety input values.
	  - The controller sends 'run' safety output packets.
	• Safety mapped tags are updated to reflect the standard controller tag values.
	• Safety task logic is being scanned.
	• All safety task process logic, cross-compare logic outputs. Logic outputs are written to safety outputs.

Table 9. Safety Application Status

Safety Task Status	Safety Rating[2] (Up to and Including)	Controller Behavior
Unlocked no signature	Only for development purposes	• Safety I/O forces can be present. • Safety I/O forces can be modified. • Safety online editing is allowed. • Safety memory is isolated, but is unprotected (read/write). • Download allowed if a major firmware revision of the offline project matches the target controller.
Locked no signature	Only for development purposes	• Safety I/O forces are not allowed. Forces of safety I/O must be removed before locking is possible. • Online editing of the safety task is not allowed. • Safety memory is protected (read-only). • Download is not allowed.
Unlocked with signature	SIL 2/SIL 3 (per controller)	• Safety I/O forces are not allowed. Forces of safety I/O must be removed before locking is possible. • Online editing of the safety task is not allowed. • Safety memory is protected (read-only). • Safety signature is unprotected and anyone who has access to the controller can delete it. • Download allowed if a major firmware revision of the offline project matches the target controller.
Locked with signature	SIL 2/SIL 3 (per controller)	• Safety I/O forces are not allowed. • Online editing of the safety task is not allowed. • Safety memory is protected (read-only). • Safety signature is protected. You must enter the unlock password to unlock the controller before you can delete the safety signature. • Download is allowed if the major and minor firmware revision and signature of the offline project match the target controller, the project is safety-locked, and the safety task status of the controller is OK. **IMPORTANT:** If the controller is safety-locked and the safety-unlock password is lost and a download is needed, you must perform a Stage 1 reset of the controller.

Application Development and Testing

We recommend that a user who is trained and experienced in safety applications develop the application program for the intended SIL 2 or SIL 3 system. The developer must follow good design practices:

- Use functional specifications, including flowcharts, timing diagrams, and sequence charts.
- Perform a review of safety task logic.
- Perform application validation.

As you develop your safety application, consider the following.

2. To achieve this level, you must adhere to the safety requirements defined in this safety reference documentation.

IMPORTANT:

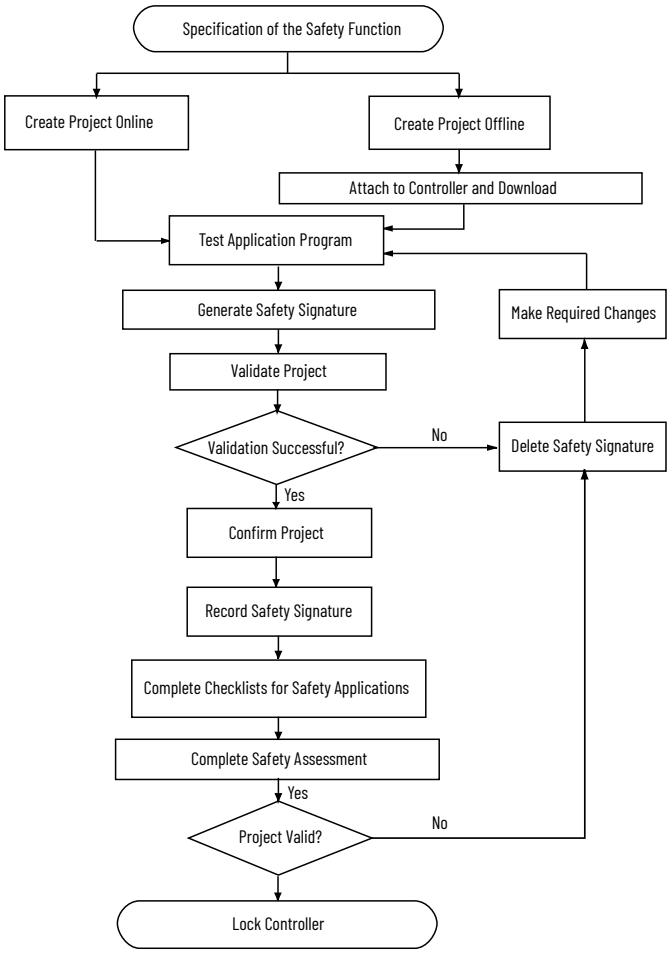
- The programming application has been certified to clause 7.4.4 of IEC 61508-3 Edition 2 and can be used during the coding lifecycle of controller-based applications and also as an aide in the module test, integration test, and validation test lifecycle phases. As a result, no additional justification for its use during those lifecycle phases is required. If, however, other tools are used, either on their own or with the application, additional justification for those other tools are required. It is your responsibility to verify that other offline tools that are used during all lifecycle phases are selected as a coherent part of the software development activities.
- It is your responsibility to conduct an assessment to determine the level of reliance that is placed on the programming application and the potential failure mechanisms that can affect the executable software when the application is used in a manner other than what is specified in the product documentation.
- You must verify that all programming and configuration information that is entered into the programming application, and downloaded to the controller, meets the requirements for your application. See Confirm the Project on page 63.
- As required by the safety integrity level, the software or design representation must match the characteristics of the application.
- As required by the safety integrity level, the software or design representation must be compatible with the features that are supported in the programming application and controllers. It is your responsibility to verify that the desired software and design representation are supported in the application and controllers. For example, if the design is represented in a flowchart format, it is your responsibility to convert that design to a ladder diagram.
- Use of third-party, or internally developed, tools to generate logic automatically to import into the programming application for compilation and download to a controller requires assessment of its suitability at the point in the development cycle where it is selected.

Commissioning Lifecycle

The flowchart shows the steps that are required for commissioning a safety system.

IMPORTANT: To achieve IEC-62443-4-2 SL 1 security certification, you must enable CIP Security™ before you download a safety application to the controller.

Figure 33. Commission a Safety System



Specification of the Safety Function

You must create a specification for your safety function. Use this specification to verify that program logic correctly and fully addresses the functional and safety control requirements of your application. In some applications, the specification can be presented in various formats. However, the specification must be a detailed description that includes the following (if applicable):

- Sequence of operations
- Flow and timing diagrams
- Sequence charts
- Program description
- Program printout
- Written descriptions of the steps with step conditions and actuators to be controlled, which includes the following:
  - Input definitions
  - Output definitions
  - I/O wiring diagrams and references
  - Theory of operation
- Matrix or table of stepped conditions and the actuators to be controlled, including the sequence and timing diagrams
- Definition of marginal conditions, for example, operating modes and emergency stop

The I/O portion of the specification must contain the analysis of field circuits:

- Sensors (digital or analog)
  - Signal in standard operation (dormant current principle for digital sensors, sensors OFF means no signal)
  - Determination of redundancies that are required for SIL levels
  - Discrepancy monitoring and visualization, including your diagnostic logic
- Actuators
  - Position and activation in standard operation (normally ON)
  - Safe reaction/positioning when switching OFF or power failure
  - Discrepancy monitoring and visualization, including your diagnostic logic

Create the Project

The logic and instructions that are used in programming the application must be the following:

- Easy to understand
- Easy to trace
- Easy to change
- Easy to test

Review and test all logic. Keep safety-related logic and standard logic separate.

Use these labels to identify the application program clearly:

- Name
- Date
- Revision
- Any other useful identification

Test the Application Program

This step consists of any combination of Run and Program modes, online or offline edits, upload and download, and informal testing that is required to get an application to run properly in preparation for the Project Validation test.

Generate the Safety Signature

The safety signature is composed of a safety signature ID and a time stamp. The safety signature ID applies to the entire safety portion of the controller and uniquely identifies each project, including its logic, constant data, and configuration.

ATTENTION: The safety signature is required for the controller to operate at a SIL 2 or SIL 3 rating. Controller operation without a safety signature is only suitable during development.

Before you generate the safety signature, make sure the following requirements are met:

- Your programming tests are complete, but verification testing has not begun.
- The programming software is online with the controller.
- With firmware revision 38 or later, the controller can be in Remote Run, Program/Remote Program, or Remote Test mode. If your controller supports firmware revision 37 or earlier, the controller must be in Program/Remote Program mode.

ATTENTION: When the controller is in Program mode, outputs are commanded to their Program mode state.

- The controller is safety-unlocked.
- The controller has no safety forces or pending online safety edits.
- If a constant safety tag value was changed online, the project must be saved with an upload of the online tag values.
- The safety task status is OK.

IMPORTANT:

When the safety application has been validated, there can be scenarios that require a redownload even though the safety application has not changed. For example, you make changes to the standard application, but not the safety application.

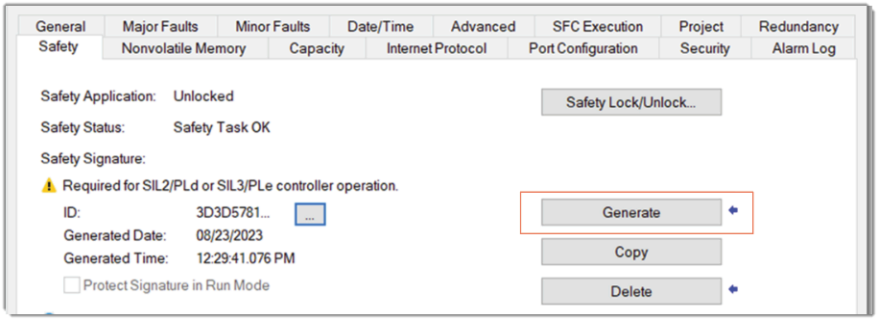
To verify that the correct safety application is downloaded, manually record the safety signature after initial creation and check the safety signature after every download to make sure that it matches the original sign.

You can generate the safety signature in three ways:

- From the Safety tab of the Controller Properties dialog box, click Generate, as shown in the figure below.
- From the Tools menu, select Safety > Generate Signature.
- From the Safety Status menu, select Generate Signature.

The programming application automatically uploads the safety signature after it is generated.

Figure 34. Generate Safety Signature



To view and copy the entire 64-character signature ID, click the Ellipse button next to the ID to open the Safety Signature ID dialog box.

Figure 35. Safety Signature ID

Safety signature creation and deletion are logged in the controller log. For more information on accessing the controller log, refer to Logix 5000 Controllers Information and Status Programming Manual, publication 1756-PM015.

When a safety signature exists, the following actions are not permitted in the safety portion of the application:

- Online/offline programming or editing, including safety Add-On Instructions
- Force safety I/O
- Change the inhibit state of safety I/O or producer controllers from module properties

- Safety data manipulation, except by safety routine logic
- Download a new safety application

With Logix SIS, you can update the firmware when a safety signature exists by using the Redundancy System Update (RSU) process.

Copy the Safety Signature

You can use the Copy button to create a record of the safety signature for use in safety project documentation, comparison, and validation. When you click Copy, the safety signature ID, date, and time components are saved to the Windows® clipboard.

Delete the Safety Signature

If safety-related changes are required to an application that has a safety signature, the safety signature must be deleted.

ATTENTION:

When you delete the safety signature, all safety functions are impacted and require additional safety measures while the safety signature is removed.

After you delete the safety signature, you must retest and revalidate your system at some level to meet SIL 2/PLd or SIL 3/PLe.

To delete the safety signature, click Delete. The safety signature cannot be deleted when the following is true:

- The controller is safety-locked.
- The controller is in Run mode with the switch set to RUN.
- The controller is in Run or Remote Run mode with Protect Signature in Run Mode enabled.

Validate the Project

To check your application program for adherence to the specification, you must generate a suitable set of test cases that cover the application. The set of test cases must be documented and retained as the test specification. To determine what to validate for your specific application, refer to IEC 61508 or your industry-specific safety standard.

You must include a set of tests to prove the validity of your application logic. These are tests within the defined value ranges, at the limits, or in invalid value ranges. The necessary number of test cases depends on safety application.

Active validation with field devices must also be included, as it is the only way to verify that the sensors and actuators in the system are wired correctly. Verify the operation of programmed functions by manipulating sensors and actuators manually.

You must also include tests to verify the reaction to wiring faults and network communication faults.

Project validation includes tests of fault routines, and input and output channels, to be sure that the safety system operates properly.

To perform a project validation test on the controller, you must perform a full test of your application. You must activate each sensor and actuator that is involved in every safety function. Be sure to test all shutdown functions, because these functions may not be exercised during normal operation.

Also, know that a project validation test is valid only for the specific application tested. If the safety application is moved to another installation, you must perform startup and project

validation on the safety application in the context of the new sensors, actuators, wiring, networks, and control system physical equipment.

Revalidation Considerations

The IEC 61508 functional safety standard requires an impact analysis before you upgrade or modify components in a certified, functional safety system. Reference the standard to make sure that you fulfill all requirements as they relate to your application. Consider the following high-level information for impact analysis of safety controller software, hardware, and firmware modification:

- All major and minor firmware releases for safety controller systems are certified for use in safety applications. As part of the certification process, Rockwell Automation tests the safety-related firmware functions, such as the CIP Safety™ communication subsystems, which are embedded safety instruction execution, and safety-related diagnostic functions. The firmware release notes identify changes to safety-related functions.
- Perform an impact analysis of the planned modifications.
  - Review the firmware release notes for changes in safety-related functionality.
  - Review the hardware and firmware compatibility in the Product Compatibility and Download Center (PCDC) to identify potential compatibility conflicts.
  - Plan, analyze, and document the impact of any modification, enhancement, or adaptation of your validated safety system.
  - As part of the upgrade process, remove and regenerate the safety signature.
- Based on the results of the safety impact analysis, choose the appropriate level of hardware and software revalidation. Use the Safety Signature report to determine which safety elements have been modified and require revalidation. If your validation plan does not require revalidation of unchanged elements, your certification effort can be reduced.

Confirm the Project

You must review the project and compare the uploaded safety I/O and controller configurations, safety data, and safety task program logic to make sure that the correct safety components were downloaded, tested, and retained in the safety application program.

If your application program contains a safety Add-On Instruction that has been sealed with an instruction signature, you must also compare the instruction signature, date/time, and safety instruction signature to the values you recorded when you sealed the Add-On Instruction.

The following steps illustrate one method for confirming the project.

1. While online with the controller, save the project.
2. Answer Yes to the Upload Tag Values prompt.
3. With the Studio 5000 Logix Designer® application offline, save the project with a new name, such as Offlineprojectname.acd, where 'projectname' is the name of your project.
   This file is the new tested primary project file.
4. Close the project.
5. Move the original project archive file out of its current directory.
   You can delete this file or store it in an archival location.
   This step is required because if the Studio 5000 Logix Designer® application finds the projectname.acd in this directory, it correlates it with the controller project and does not perform an upload.
6. Upload the project from the controller.

7. Save the uploaded project as Onlineprojectname.acd, where 'projectname' is the name of your project.
8. Answer Yes to the Upload Tag Values prompt.
9. Use the Studio 5000 Logix Designer® Program Compare utility to perform these comparisons:
   a. Compare all properties of the controller and CIP Safety™ I/O devices.
   b. Compare all properties of the safety task, safety programs, and safety routines.
   c. Compare all logic in the safety routines.
10. Verify that all controller and I/O configuration fulfills the requirements of your application specification.

Complete Checklists for Safety Applications

Complete the checklists located in the Checklists for Safety Applications Appendix.

Safety Assessment

An independent, third-party safety assessment can be required before the system is approved for operation. An independent, third-party certification can be required for IEC 61508 SIL 2 or SIL 3 levels.

Lock the Controller

ATTENTION: Safety-locking alone does not satisfy SIL 2/PLd or SIL 3/PLe requirements.

The default state of the controller is safety-unlocked. We recommend that you safety-lock the controller to help protect safety control components from modification and help prevent the safety signature from being deleted accidentally. However, safety-locking the controller is not a requirement for SIL 2 or SIL 3.

The safety-lock feature applies only to safety components, such as the safety task, safety programs, safety routines, safety tags, safety Add-On Instructions, safety I/O, and safety signature.

No aspect of safety can be modified while the controller is in the safety-locked state. When the controller is safety-locked, the following actions are not permitted in the safety task:

- Update the firmware without using the Redundancy System Update (RSU) process
- Online or offline programming or editing
- Forcing safety I/O
- Data manipulation of safety components except through routine logic
- Creating or editing safety Add-On Instructions
- Generating or deleting the safety signature

IMPORTANT: If a safety signature exists and the controller is safety-locked, only projects with a matching safety signature can be downloaded to the controller.

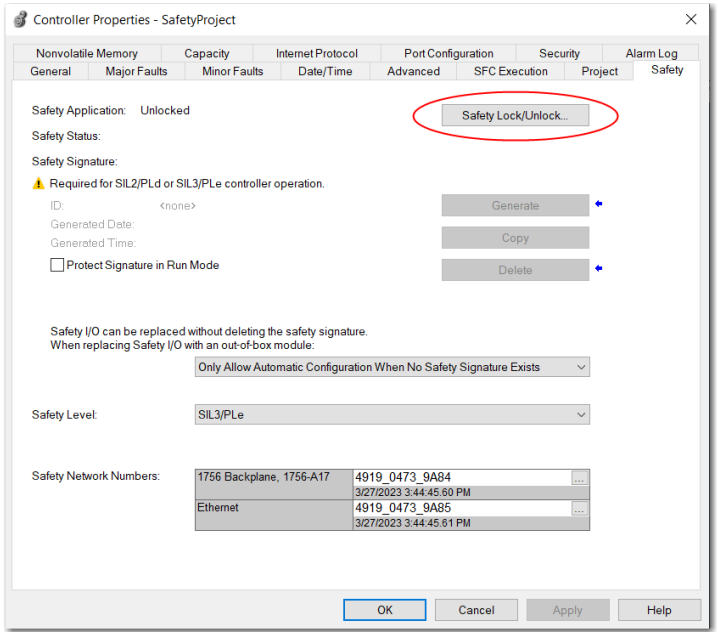
You can place the safety application in a safety-locked state regardless of whether you are online, offline, or you have the original program source. However, no safety forces or pending safety edits can be present. Safety-locked or -unlocked status cannot be modified when the keyswitch is in the RUN position.

Safety-lock and safety-unlock actions are logged in the controller log. For more information on accessing the controller log, see the Logix 5000 Controllers Information and Status Programming Manual, publication 1756-PM015.

You can change the safety-lock status in three ways:

- From the Safety tab of the Controller Properties dialog box.
- In the Studio 5000 Logix Designer® application, select Tools > Safety > Safety Lock/Unlock.
- From the Safety Status drop-down menu, select Safety Lock/Unlock.

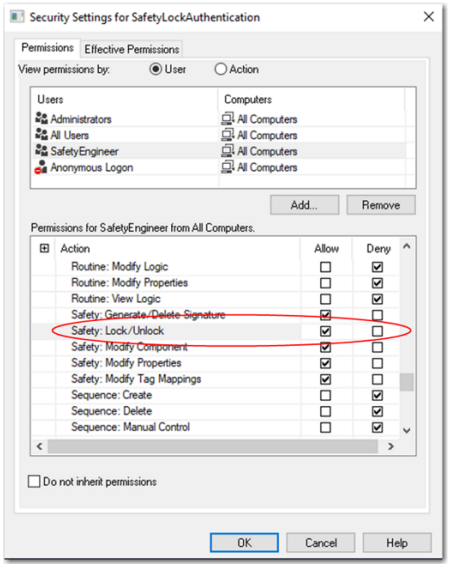
Figure 36. Safety Lock/Unlock Button



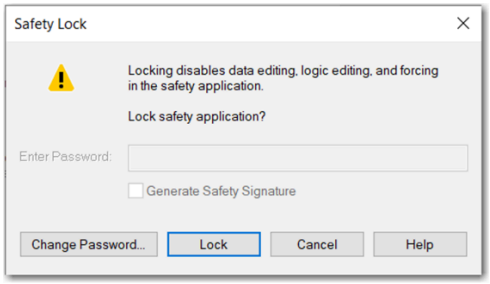
Restrict Access to Safety-lock and Safety-unlock Functionality

To provide additional layers of protection, you can use two methods to restrict access to safety-lock and safety-unlock functionality:

- FactoryTalk Security—To comply with IEC-62443-4-2 SL 1 security certification, you must use FactoryTalk Security to configure restricted access to safety-lock and safety-unlock actions. For information about setting these permissions, see the Configure System Security Features User Manual, publication SECURE-UM001.



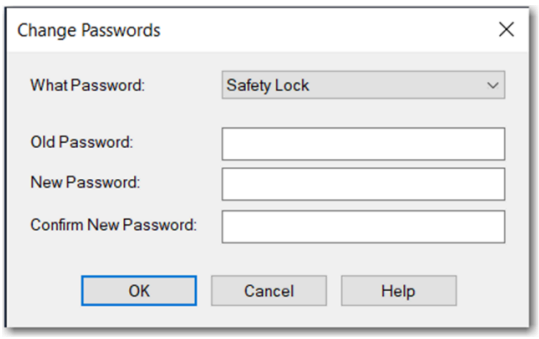
- Safety passwords—You can set separate passwords that are required to safety-lock or safety-unlock the controller. Passwords are optional and not required for IEC-62443-4-2 SL 1 security certification. For security certification requirements, see the ControlLogix 5580 and GuardLogix 5580 Controllers User Manual, publication 1756-UM543. For security certification requirements, see the CompactLogix 5380 and Compact GuardLogix 5380 Controllers User Manual, publication 1756-UM543.



IMPORTANT: For added security, enable CIP Security on the controller before you set password protection. For non-redundant controllers, we recommend that you enable CIP Security on the front Ethernet port of the controller. For more information about CIP Security, see CIP Security with Rockwell Automation Products, publication SECURE-AT001.

To set password protection for safety-lock or safety-unlock actions, follow these steps.

1. On the Logix Designer menu bar, click Tools > Safety > Change Passwords.
2. In the What Password field, select Safety Lock or Safety Unlock.



3. Enter the old password, if one exists.
4. Enter and confirm the new password.
5. Click OK.

Passwords can be from 1...40 characters in length and are not case-sensitive. Letters, numerals, and the following symbols can be used: ' ~ ! @ # $ % ^ & * ( ) _ + , - = { } | [ ] \ : ; ? / . To clear an existing password, enter a new password of zero length.

IMPORTANT: Rockwell Automation does not provide any form of password or security override services. When products and passwords are configured, Rockwell Automation encourages customers to follow good security practices and to plan accordingly for password management.

Download/Upload a Safety Application Program

Upon download, application testing is required unless a safety signature exists.

IMPORTANT: To verify that the correct safety application is downloaded or restored from a memory card, you must manually check that the safety signature matches the original signature in your safety documentation.

Downloads to a safety-locked controller are allowed only if the safety signature and the firmware revision of the offline project all match what is contained in the target controller and the safety task status of the controller is OK.

IMPORTANT: If the safety signature does not match and the controller is safety-locked, you must unlock the controller to download. In this case, downloading to the controller deletes the safety signature. As a result, you must revalidate the application.

IMPORTANT: To avoid project download failure, do not change the communication adapter name of a chassis containing safety I/O while the safety application has a safety signature.

If the controller contains a safety signature, the safety signature is uploaded in an online save of the project. The option to upload tag values includes both standard and safety tag values.

Store and Load a Project from a Memory Card

Safety controllers support firmware updates and user program storage and retrieval with a memory card.

When you store a safety project on a memory card, we recommend that you select Remote Program as the Load mode. The Load mode specifies the mode that the controller enters after a project load. Before system operation, operator intervention is required to start the controller.

You can initiate a load from a memory card only under these conditions:

- If the controller type specified by the project that is stored on the memory card matches your controller type.
- If the major and minor revisions of the project on the memory card match the major and minor revisions of your controller.

  IMPORTANT: A revision mismatch helps prevent only user-initiated loads. Controller-initiated loads overwrite the firmware on the controller with the contents of the memory card.

- If your controller is not in Run mode.

Loading a project to a safety-locked controller is allowed only when the safety signature of the project that is stored on the memory card matches the project on the controller. If the signatures do not match or the controller is safety-locked without a safety signature, you must first unlock the controller before attempting to update the controller via a memory card.

IMPORTANT: If you unlock the controller and initiate a load from the memory card, the safety-lock status, passwords, and safety signature are then set to the values contained on the memory card once the load is complete.

Force Data

All data that is contained in an I/O tag can be forced while the project is safety-unlocked and no safety signature exists. However, forces must be removed, not just disabled, on all safety tags before the safety project can be safety-locked or a safety signature can be generated. You cannot force safety tags while the project is safety-locked or when a safety signature exists.

You can install and remove forces on standard tags regardless of the safety-locked or safety-unlocked state.

Inhibit a Device

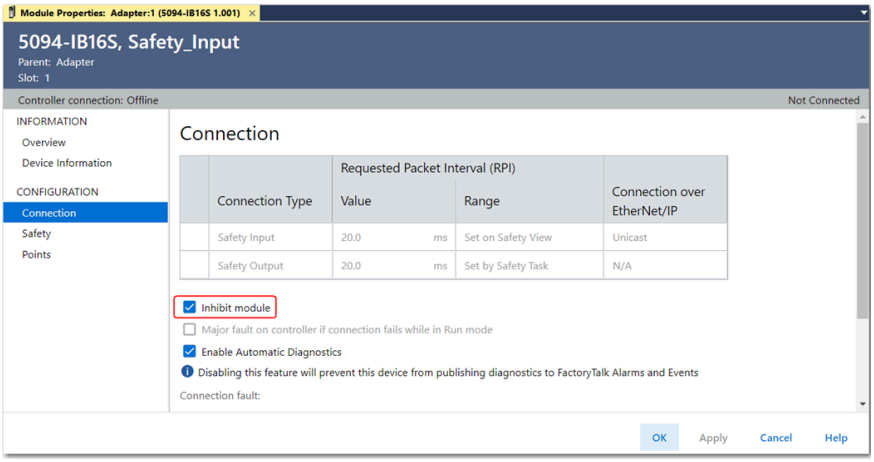
You cannot inhibit or uninhibit safety I/O devices from the module properties dialog under these conditions:

- The application program is safety-locked
- A safety signature exists

To inhibit a device, follow these steps.

1. Select the device and then select Properties.
2. In the navigation pane, select Connection.
3. On the Connection view, select Inhibit Module and click Apply.

The device is inhibited when the Inhibit Module checkbox is selected. If a communication device is inhibited, all downstream devices are also inhibited.



Regardless of safety signature or safety-locked status some I/O products allow the ability to programmatically inhibit and uninhibit with SSV from the standard task:

- Class Name: Module
- Attribute Name: Mode
- Source: Inhibit = 4; Uninhibit = 0

Edit a Safety Application

The following rules apply to changing your safety application program:

- Only authorized, specially trained personnel should make program edits. These personnel must use all supervisory methods available, for example, using the controller switch and software password protections.
- When authorized, specially trained personnel make program edits, they assume the safety responsibility while the changes are in progress. These personnel must also maintain safe application operation.
- When you edit online, you must use an alternate protection mechanism to maintain the safety of the system.
- You must sufficiently document all program edits, which include the following:
  - Authorization
  - Impact analysis
  - Execution
  - Test information
  - Revision information
- If online edits exist only in the standard routines, those edits are not required to be validated before returning to normal operation.
- You must make sure that changes to the standard routine, regarding timing and tag mapping, are acceptable to your safety application.
- You can edit the logic portion of your program while offline or online, as described in the following sections.

Offline Edits

When you make offline edits to only standard program elements and the safety signature matches following a download, you can resume operation.

When you make offline edits to the safety program, you must revalidate all affected elements of the application, as determined by the impact analysis, before you resume operation.

Online Edits

ATTENTION: Performing online edits to logic, data, or the configuration can affect the safety functions of the system if the edits are performed while the application is running. Online edits should only be done if necessary. If the edits are not performed correctly, they can stop the application. You must use alternative safety measures and constraints during online edits.

Online edits in standard routines are unaffected by the safety-locked or safety-unlocked state.

The following requirements apply to online edits of safety logic:

- The controller must be safety-unlocked and unsigned. If the controller is locked with safety edits, you must unlock the controller to assemble or cancel the edits.
  You assemble edits to make online edits change the controller program.

  You cancel edits to reject and delete any unassembled online edits.
- For safety routines, the controller cannot be locked when there is a pending edit, but it can be locked when there is a test edit.
  A pending edit is a change to a routine that has been made in the Studio 5000 Logix Designer® application, but has not yet been communicated to the controller by accepting the edit.

  A test edit is an online edit that has been accepted and causes the controller to execute the new, edited version of logic. The original, unedited version of logic is still in controller memory, but is not executed.

  IMPORTANT: Certain parameters are only evaluated when the instruction is first evaluated. You must transition the controller to Program mode and back to Run mode before the changes take effect. For affected operands, see the Logix 5000® Controller Safety Application Instruction Set Reference Manual, publication 1756-RM095.

You cannot edit standard or safety Add-On Instructions when the controller is online.

When you make online edits to the safety program, you must revalidate all affected elements of the application, as determined by the impact analysis, before you resume operation.

Limit online edits to minor program modifications, such as setpoint changes or minor logic additions, deletions, and modifications.

The safety-lock and safety signature features of the controller affect online edits. For more information, see Lock the Controller on page 64 and Generate the Safety Signature on page 60.

Modification Impact Test

Any modification, enhancement, or adaptation of your validated software must be planned and analyzed for any impact to the functional safety system. All appropriate phases of the software safety lifecycle must be conducted as indicated by the impact analysis. If your validation plan requires cold start or warm start testing of the modification, this can be achieved by transitioning from Program mode to Run mode. Special attention should be paid to the impact

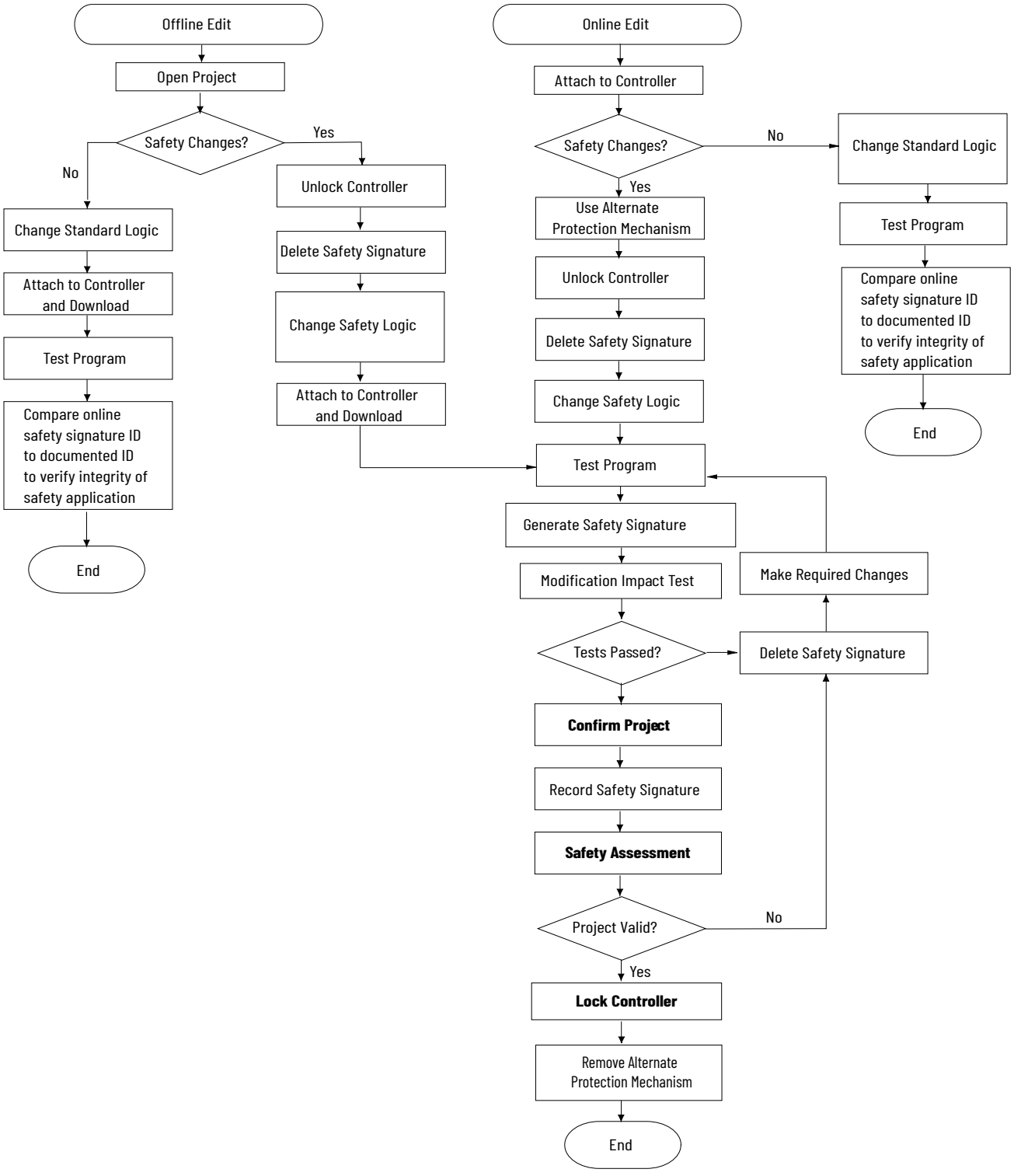
of the modification on safety task initialization as described in Custom Tag Initialization During Prescan on page 49.

At a minimum, you must perform these actions:

- Perform functional tests of all impacted software.
- Document all modifications to your software specifications.
- Document all test results.

For detailed information, see IEC 61508-3, Section 7.8 Software Modification.

Figure 37. Online and Offline Edit Process



Monitor Safety Status and Handle Faults

There are multiple methods to detect and react to faults in the system. The first way that you can handle faults is to verify that you have completed the checklists for your application as described in Checklists for Safety Applications on page 99.

For details about the operation of status indicators, see the user manual for the controller.

IMPORTANT: Status indicators do not provide detailed diagnostics for safety functions. Use them only for general diagnostics during commissioning or troubleshooting. Do not attempt to use status indicators to determine operational status.

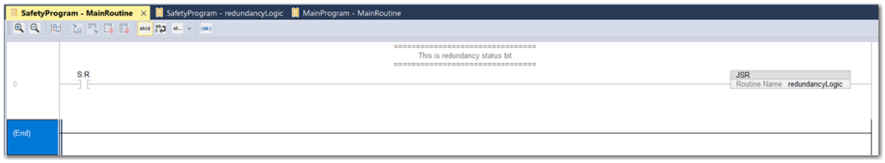
Redundancy Status

You can determine if your system is operating with redundancy by using the Redundancy Status bit (S:R) in standard or safety task logic. The bit can monitor the system while the controller is in Run or Test mode.

S:R Bit Status	Redundancy Status
On (1)	The system is operating with redundancy.
Off (0)	The system operating without redundancy.

IMPORTANT: When the S:R bit turns Off in a safety task that supports a SIL 3 safety function, you must repair the system within your specified mean repair time (MRT). If the system is not repaired within the MRT, you must take a specified action to maintain or achieve a safe state.

Figure 38. Redundancy Status Bit



System Status

You can monitor the system status with the following diagnostic indicators.

IMPORTANT: It is your responsibility to determine what data is most appropriate to initiate a shutdown sequence.

Safety I/O Module Diagnostics

Safety I/O modules provide pulse test and monitoring capabilities. If the module detects a failure, it sets the offending input or output to its safe state and reports the failure to the controller. The failure indication is made via input or output status and is maintained for a configurable amount of time after the failure is repaired.

I/O Device Connection Status

The CIP Safety™ protocol allows the recipients of I/O data to determine the status of that data:

- The controller detects input connection failures and then sets all input data to the safe state and the associated input status to faulted.
- The output device detects output connection failures and then de-energizes its outputs.
- Generally, the safety controller also has input connections from output devices. The safety controller determines the status of these input connections, but the input connection status is not the primary mechanism to de-energize outputs.

IMPORTANT: You are responsible for providing application logic to latch I/O failures and to verify that the system restarts properly.

De-energize to Trip System

Safety controllers are part of a de-energize to trip system, which means that off is the safe state. Some, but not all, safety I/O device faults cause all device inputs or outputs to be set to safe state. Faults that are associated to a specific input channel result in that specific channel being set to a safe state. For example, a pulse test fault that is specific to channel 0 results in channel 0 input data being set to the safe state. If a fault is general to the device and not to a specific channel, the combined status bit displays the fault status and all device data is set to the safe state.

Get System Value (GSV) and Set System Value (SSV) Instructions

The GSV and SSV instructions let you get (GSV) and set (SSV) controller system data that is stored in device objects. When you enter a GSV/SSV instruction, the programming software displays the valid object classes, object names, and attribute names for each instruction. Restrictions exist for using the GSV and SSV instructions with safety components.

IMPORTANT:

With firmware revision 37, even when your system functions at a SIL 3 level, the SafetySILConfiguration attribute always shows a SIL 2 value. This value is expected because it reflects the Safety Level setting in the controller properties. A SIL2/PLd safety level is the required configuration for safety controllers that are enabled for redundancy.

With firmware revision 38, even when your system functions at a SIL 2 level, the SafetySILConfiguration attribute always shows a SIL 3 value.

IMPORTANT:

The safety task cannot perform GSV or SSV operations on standard attributes.

The standard task can GSV diagnostic information from safety objects and SSV can inhibit some safety I/O devices.

For more information about GSV and SSV instructions, see the Logix 5000 Controllers General Instructions Reference Manual, publication 1756-RM003.

Safety Status

You can use the following to monitor safety status:

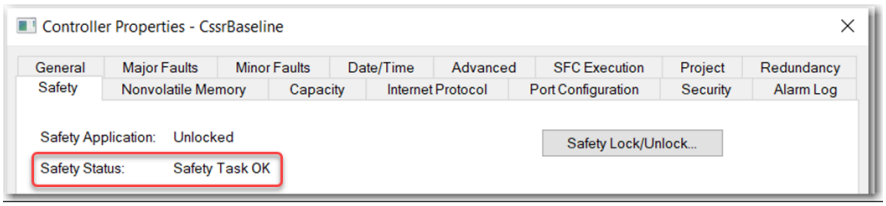
- Safety task status in the controller properties
- Safety I/O status

Safety Task Status

The safety task status appears on the Safety tab of the controller properties and is one of the following:

- Safety Task Inoperable
- Safety Task OK

Figure 39. Safety Task Status



Safety I/O Status

Connection Status (.ConnectionFaulted) is the status of the safety connection between the controller and the safety I/O module:

- When the connection is operating properly, this bit is LO (0).
- When the connection is not operating properly, this bit is HI (1). All other module-defined tags are LO and considered invalid data.

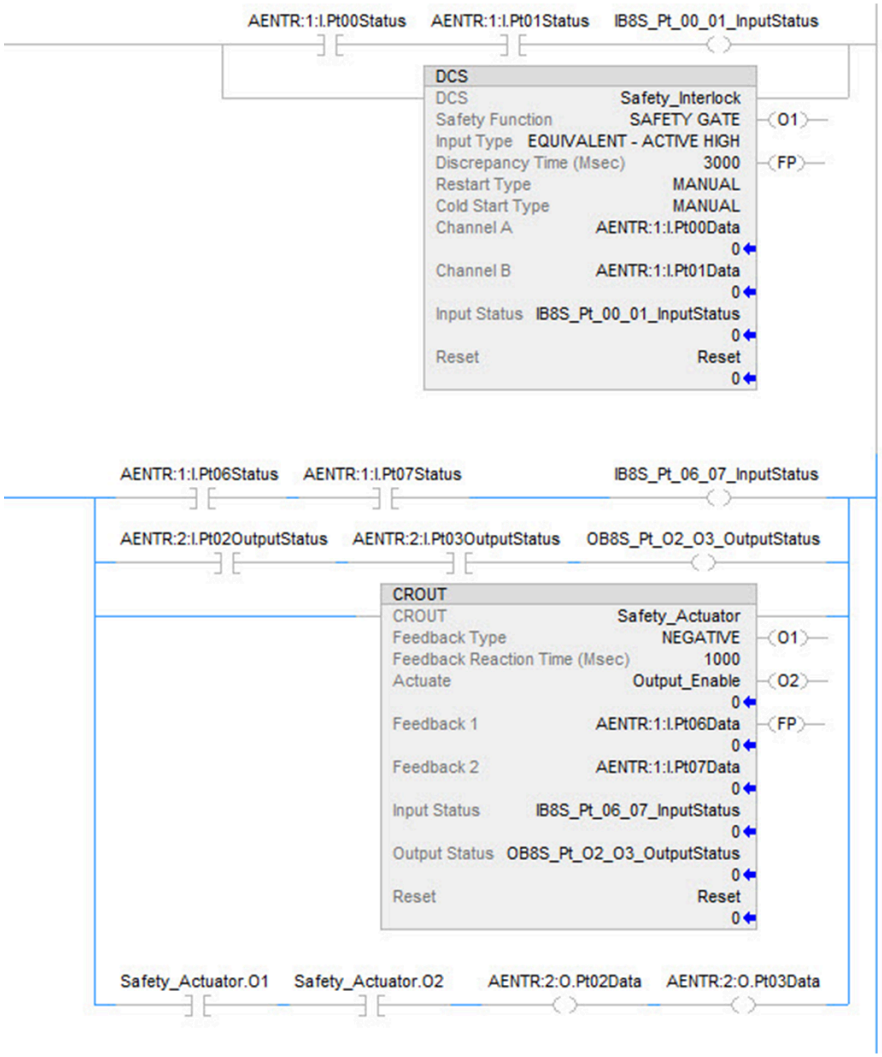
Point Status or Channel Status is available for safety inputs and safety outputs. When a status tag is HI (1), it indicates the following:

- An individual channel is functioning and wired correctly.
- The safety connection between the controller and the safety I/O module on which the channel resides is operating properly.

The safety instructions have built-in safety I/O status monitoring. Input Status and Output Status are parameters for the safety input and output instructions. The DCS instruction and other dual-channel safety instructions have Input Status for input channels A and B. The CROUT instruction has Input Status for Feedbacks 1 and 2 and Output Status for the output channels that are driven by the CROUT outputs O1 and O2. The status tags in these instructions must be HI (1) for the safety instruction output tags (O1 for input instructions and O1/O2 for CROUT) to be energized.

For proper safety instruction operation, it is important to drive the input status and output status tags BEFORE/ABOVE the safety instruction as shown in the following example.

Figure 40. Instruction Examples



When you use instructions, such as XIC and OTE, to directly reference safety I/O data points without the use of a safety-rated instruction, you are responsible for interrogating the safety I/O status:

- Before you use a safety input channel as an interlock, verify that the safety input channel status is HI (1).
- Before you energize a safety output channel, verify that the safety output channel status is HI (1).

Logix SIS Safety Faults

Logix SIS can experience the following types of faults.

Major Nonrecoverable Faults

A major nonrecoverable fault typically indicates that one controller has a hardware or program fault that prevents operation. For example, if a safety diagnostic detects a problem in the hardware, a major nonrecoverable fault occurs. In this case, the system attempts to execute the safety task on the other controller while the faulted controller shuts down.

Safety Program Faults

Because the safety task executes concurrently on the primary and secondary controllers, most safety program faults occur on both controllers in the same scan, but at slightly different times. Examples of safety program faults include programming issues or safety task watchdog faults.

A safety program fault on both controllers has one of the following results:

- If the fault first occurs on the secondary controller, the redundant chassis pair becomes disqualified.
- If the fault first occurs on the primary controller, then a switchover occurs.

Cross-compare Faults

During synchronized operation between a primary and secondary controller, the controllers cross-check the results of the safety task. If the cross-check fails, a fault occurs and both controllers shut down.

Loss-of-safety Faults

If a loss of redundancy occurs, the safety function is temporarily muted while the system determines the controller to become the lone primary. If the safety task cannot determine the new primary within a certain amount of time, a fault occurs.

For more information about safety function muting, see Safety Function Muting on page 17.

View Faults

The Recent Faults dialog box on the Major Faults tab of the Controller Properties dialog box contains two subtabs, one for standard faults and one for safety faults.

The status display on the controller also shows fault codes with a brief status message. For more information about status indicators, see the controller user manual.

Fault Codes

Safety controllers show fault codes on the Major Faults tab of the Controller Properties dialog box and in the PROGRAM object, MAJORFAULTRECORD or MINORFAULTRECORD attribute.

IMPORTANT: This manual links to Logix 5000 Controller and I/O Fault Codes and Syslog Messages, 1756-RD001; the file automatically downloads when you click the link.

Override Safety Faults

To enable standards tasks to continue after a safety fault, you can override the safety fault in the controller-scoped fault handler. Before you override a safety fault in Logix SIS, consider the following:

- Standard task impact—The controller-scoped fault handler runs only on the primary controller. If you override a safety fault from the controller-scoped fault handler, standard tasks continue only on the primary controller. Standard tasks continue on the primary controller, and the redundant chassis pair becomes disqualified.
- Safety task impact—If you override a safety fault from the controller-scoped fault handler, the safety task stops on both controllers.

ATTENTION:
Overriding a safety fault does not clear the fault. If you override a safety fault, it is your responsibility to prove that operation of your system is still safe.

You must provide proof to your certifying agency that your system can continue to operate safely after an override of a safety fault.

You must clear the safety fault to start the safety function again, and you must requalify the system to restore safety task execution on both controllers to meet SIL 3 requirements.

Fault Routine for Safety Applications

If a fault condition occurs that is severe enough for the controller to shut down, the controller generates a major fault and stops the execution of logic.

Some applications do not want all safety faults to shut down the entire system. In those situations, use a fault routine to clear a specific fault and let the standard control portion of your system continue to operate or configure some outputs to remain ON.

ATTENTION:
You must provide proof to your certifying agency that your system can continue to operate safely after an override of a safety fault.

The occurrence of recoverable faults is an indication that the application code is not protecting itself from invalid data values or conditions. Consider modifying the application to eliminate these faults, rather than handling them at runtime.

The controller supports two levels for handling major faults in a safety application:

- Safety Program Fault Routine
- Controller Fault Handler

Both routines can use the GSV and SSV instructions.

Each safety program can have its own fault routine. The controller executes the program's fault routine when an instruction fault occurs. If the program's fault routine does not clear the fault, or if a program fault routine does not exist, the safety task faults and shuts down.

When the safety task faults, a standard major recoverable fault is also logged, and the controller proceeds to execute the controller fault handler, if one exists. If the controller fault handler handles this fault, then the standard tasks continue to run, even though the safety task remains faulted.

The controller fault handler is an optional component that executes when the program fault routine cannot clear the fault or does not exist.

You can create one program for the controller fault handler. After you create that program, you must configure a routine as the main routine.

The Logix 5000 Controllers Major, Minor, and I/O Faults Programming Manual, publication 1756-PM014, provides details on creating and testing a fault routine.

GSV/SSV Instructions in a Safety Application

For standard tasks, you can use the GSV instruction to get values for the available attributes. When using the SSV instruction, the software displays only the attributes that you can set.

For the safety task, the GSV and SSV instructions are more restricted. The SSV instructions in safety and standard tasks cannot set bit 0 (major fault on error) in the mode attribute of a safety I/O device.

ATTENTION: Use the SSV instruction carefully. Making changes to objects can cause unexpected controller operation or injury to personnel.

FaultRecord Attributes

Create a user-defined structure to simplify access to the MajorFaultRecord and SafetyTaskFaultRecord attributes.

Table 10. Parameters for Accessing FaultRecord Attributes

Name	Data Type	Style	Description
TimeLow	DINT	Decimal	Lower 32 bits of the fault time stamp value
TimeHigh	DINT	Decimal	Upper 32 bits of the fault time stamp value
Type	INT	Decimal	Fault type (program, I/O, or other)
Code	INT	Decimal	Unique code for this fault (dependent on fault type)
Info	DINT[8]	Hexadecimal	Fault-specific information (dependent on fault type and code)

Fault Information

The SafetyStatus and SafetyTaskFaultRecord attributes can capture information about nonrecoverable faults. Use a GSV instruction in the controller fault handler to capture and store fault information. The GSV instruction can be used in a standard task with a controller fault handler routine that clears the fault and lets the standard tasks continue executing.

For more information on using the GSV and SSV instructions in safety applications, refer to the Input/Output Instructions chapter of the Logix 5000 Controllers General Instructions Reference Manual, publication 1756-RM003.

Safety Application Instructions

Safety application instructions are certified for use in SIL 2 or SIL 3 applications.

ATTENTION: Safety application instructions are the only instructions that can be used in the safety task for SIL 2 or SIL 3 applications.

For more information about using instructions, see the following:

- Logix 5000 Safety Application Instruction Set Reference Manual, publication 1756-RM095
- Logix 5000 Controllers General Instructions Reference Manual, publication 1756-RM003

Safety Instructions

Mnemonic	Name	Description
CROUT	Configurable Redundant Output	Controls and monitors redundant outputs.
DCA	Dual Channel Input - Analog (integer version)	Monitors two analog values for deviation and range tolerance.
DCAF	Dual Channel Input - Analog (floating point version)	Monitors two analog values for deviation and range tolerance.
DCM	Dual Channel Input - Monitor	Monitors dual-input safety devices.
DCS	Dual Channel Input - Stop	Monitors dual-input safety devices whose main purpose is to provide a stop function, such as an E-stop, light curtain, or gate switch.
DCSRT	Dual Channel Input - Start	Energizes dual-input safety devices whose main function is to start a machine safely, for example an enable pendant.
DCST	Dual Channel Input - Stop With Test	Monitors dual-input safety devices whose main purpose is to provide a stop function, such as an E-stop, light curtain, or gate switch. It includes the added capability to initiate a functional test of the stop device.
DCSTL	Dual Channel Input - Stop With Test and Lock	Monitors dual-input safety devices whose main purpose is to provide a stop function, such as an E-stop, light curtain, or gate switch. It includes the added capability to initiate a functional test of the stop device. It can monitor a feedback

Mnemonic	Name	Description
		signal from a safety device and issue a lock request to a safety device.
DCSTM	Dual Channel Input - Stop With Test and Mute	Monitors dual-input safety devices whose main purpose is to provide a stop function, such as an E-stop, light curtain, or gate switch. It includes the added capability to initiate a functional test of the stop device and the ability to mute the safety device.
FSBM	Four Sensor Bi-directional Muting	Automatically disables the protective function of a light curtain temporarily, by using four sensors that are arranged sequentially before and after the sensing field of the light curtain.
SMAT	Safety Mat	Indicates whether the safety mat is occupied.
THRSe	Two-Hand Run Station – Enhanced	Monitors two diverse safety inputs, one from a right-hand push button and one from a left-hand push button, to control one output. Features configurable channel-to-channel discrepancy time and enhanced capability for bypassing a two-hand run station.
TSAM	Two Sensor Asymmetrical Muting	Automatically disables the protective function of a light curtain temporarily, by using two muting sensors that are arranged asymmetrically.
TSSM	Two Sensor Symmetrical Muting	Automatically disables the protective function of a light curtain temporarily, by using two muting sensors that are arranged symmetrically.

Metal Form Instructions

AVC	Name	Description
AVC	Auxiliary Valve Control	Controls an auxiliary valve that is used with a main valve.

AVC	Name	Description
CBIM	Clutch Brake Inch Mode	Used for press applications where minor slide adjustments are required, such as press setup.
CBCM	Clutch Brake Continuous Mode	Used for press applications where continuous operation is desired.
CBSSM	Clutch Brake Single Stoke Mode	Used in single-cycle press applications.
CPM	Crankshaft Position Monitor	Used to determine the slide position of the press.
CSM	Camshaft Monitor	Monitors motion for the start, stop, and run operations of a camshaft.
EPMS	Eight-position Mode Selector	Monitors eight safety inputs to control one of the eight outputs that correspond to the active input.
MMVC	Maintenance Manual Valve Control	Used to drive a valve manually during maintenance operations.
MVC	Main Valve Control	Controls and monitors a main valve.

Ladder Diagram Safety Instructions

Routines in the safety task can use the following types of ladder diagram safety instructions.

When instructions that result in values with decimal place are used with REAL or LREAL operands, Rockwell Automation has verified the precision to 6 decimal places.

Process

Mnemonic	Name	Description	Logix Designer Application
FGEN	Function Generator	Converts an input based on a piece-wise linear function.	Version 38 or later

Advanced Math

Rockwell Automation has not done independent mathematical analysis of the numerical algorithms used in any of the following advanced math instructions. If you require a specific degree of accuracy, you must functionally test this instruction over your expected input domain.

Mnemonic	Name	Description	Logix Designer Application
EXPT	X to the Power of Y	Returns the value of X to the power of Y.	Version 36 or later
XPY	X to the Power of Y	Returns the value of X to the power of Y.	Version 35 or earlier
LN	Natural Log	Compute the natural log of a number.	Version 36 or later
LOG	Log Base 10	Compute the log base 10 of a number.	All versions

Array (File)

Mnemonic	Name	Description	Logix Designer Application
AVE	File Average	Calculate the mean/average over a set of values.	All versions
COP	Copy File	Copy binary data from one tag to another (no type conversion). When you use the COP instruction in a safety routine, you must verify that the length is a constant and that the source and destination length are the same.	All versions
FAL	File Arithmetic and Logic	Perform copy, arithmetic, logic, and function operations on data that is stored in an array.	All versions
FLL	File Fill	Fill the elements of an array with the source value, while leaving the source value unchanged.	All versions
FSC	File Search and Compare	Compare the values in an array, element by element.	All versions
SIZE	Size in Elements	Find the size of a dimension of an array.	All versions

Mnemonic	Name	Description	Logix Designer Application
STD	File Standard Deviation	Calculate the standard deviation over a set of values.	All versions
BSL	Bit Shift Left	Shifts the specified number of bits to the left.	All versions
BSR	Bit Shift Right	Shifts the specified number of bits to the right.	All versions
FFL	FIFO Load	Store a value into an array by using first-in/first-out semanti cs.	All versions
FFU	FIFO Unload	Retrieve a value from an array by using last-in/first-out semantics.	All versions
LFL	LIFO Load	Store a value into an array by using last-in/first-out semantics.	All versions
LFU	LIFO Unload	Retrieve a value from an array by using last-in/first-out semantics.	All versions

Bit

Mnemonic	Name	Description	Logix Designer Application
ONS	One Shot	Allows an event to occur one time.	All versions
OSR	One Shot Falling	Sets an output bit for one scan on the true-to-false (falling) edge of rung state.	All versions
OSF	One Shot Rising	Sets an output bit for one scan on the false-to-true (rising) edge of rung state.	All versions
OTE	Output Energize	Controls a bit (it performs both Set and Clear operations based on rung state).	All versions
OTL	Output Latch	Set a bit (retentive).	All versions

Mnemonic	Name	Description	Logix Designer Application
OTU	Output Unlatch	Clear bit (retentive).	All versions
XIC	Examine if Closed	Examines the data bit to set or clear the rung condition.	All versions
XIO	Examine if Open	Examines the data bit to set or clear the rung condition.	All versions

Time/Counter

Mnemonic	Name	Description	Logix Designer Application
CTD	Count Down	Count down.	All versions
CTU	Count Up	Count up.	All versions
RES	Reset	Reset a timer or counter.	All versions
RTO	Retentive Timer On	Accumulate time.	All versions
TOF	Off-delay Timer	Time how long a timer is disabled.	All versions
TON	On-delay Timer	Time how long a timer is enabled.	All versions

Compare

Mnemonic	Name	Description	Logix Designer Application
EQ	Equal To	Test whether two values are equal.	Version 36 or later
EQU	Equal To	Test whether two values are equal.	Version 35 or earlier
CMP	Compare	Perform a comparison on the arithmetic operations that you specify in the expression.	All versions
GE	Greater Than Or Equal To	Test whether one value is greater than or equal to a second value.	Version 36 or later

Mnemonic	Name	Description	Logix Designer Application
GEQ	Greater Than Or Equal To	Test whether one value is greater than or equal to a second value.	Version 35 or earlier
GT	Greater Than	Test whether one value is greater than a second value.	Version 36 or later
GRT	Greater Than	Test whether one value is greater than a second value.	Version 35 or earlier
IsINF	Is Infinity	Check if floating point value is +/- infinity.	All versions
IsNAN	Is Not a Number	Check if floating point value is Not-a-Number.	All versions
LE	Less Than or Equal To	Test whether one value is less than or equal to a second value.	Version 36 or later
LEQ	Less Than or Equal To	Test whether one value is less than or equal to a second value.	Version 35 or earlier
LT	Less Than	Test whether one value is less than a second value.	Version 36 or later
LES	Less Than	Test whether one value is less than a second value.	Version 35 or earlier
LIMIT	Limit	Test whether a value falls within a specified range.	Version 36 or later
LIM	Limit	Test whether a value falls within a specified range.	Version 35 or earlier
MEQ	Mask Equal To	Pass source and compare values through a mask and test whether they are equal.	All versions
NE	Not Equal	Test whether one value is not equal to a second value.	Version 36 or later
NEQ	Not Equal	Test whether one value is not equal to a second value.	Version 35 or earlier

Move/Logical

Mnemonic	Name	Description	Logix Designer Application
AND	Bitwise AND	Perform bitwise AND operation.	All versions
CLR	Clear	Clear a value.	All versions
MOVE	Move	Copy a value.	Version 36 or later
MOV	Move	Copy a value.	Version 35 or earlier
MVM	Masked Move	Copy a specific part of an integer.	All versions
NOT	Bitwise Not	Perform bitwise NOT operation.	All versions
OR	Bitwise Or	Perform bitwise OR operation.	All versions
SWPB	Swap Byte	Rearrange the bytes of a value.	All versions
XOR	Bitwise Exclusive Or	Perform bitwise exclusive OR operation.	All versions

Program Control

Mnemonic	Name	Description	Logix Designer Application
AFI	Always False Instruction	Forces a rung to false (rung continues to execute).	All versions
EVENT	Trigger Event Task	Triggers one execution of an event task. The event instruction triggers a scan of the standard task.	All versions
JMP	Jump To Label	Scan of logic jumps to a labeled location within the same routine.	All versions
JSR	Jump to Subroutine	Jump to a separate routine.	All versions
LBL	Label	Identifies a target location for a JMP instruction.	All versions

Mnemonic	Name	Description	Logix Designer Application
MCR	Master Control Reset	Forces every rung in a section of logic to execute in the False state.	All versions
NOP	No Operation	Insert a placeholder in the logic.	All versions
RET	Return	Return the results of a subroutine.	All versions
SBR	Subroutine	Accept data that is passed to a subroutine by the JSR instruction.	All versions
TND	Temporary End	Mark a temporary end that halts routine execution.	All versions

Compute/Math

Rockwell Automation has done no independent mathematical analysis of the numerical algorithms used in the SQRT or SQR instructions. If you require a specific degree of accuracy, you must functionally test this instruction over your expected input domain.

Mnemonic	Name	Description	Logix Designer Application
ABS	Absolute Value	Take the absolute value of a value.	All versions
ADD	Add	Add two values.	All versions
CPT	Compute	Perform the arithmetic operation that is defined in the expression.	All versions
DIV	Divide	Divide two values.	All versions
MOD	Modulo	Determine the remainder after one value is divided by a second value.	All versions
MUL	Multiply	Multiply two values.	All versions
NEG	Negate	Take the opposite sign of a value.	All versions
SQRT	Square Root	Calculate the square root of a value.	Version 36 or later
SQR	Square Root	Calculate the square root of a value.	Version 35 or earlier

Mnemonic	Name	Description	Logix Designer Application
SUB	Subtract	Subtract two values.	All versions

Trigonometric

Rockwell Automation has not done independent mathematical analysis of the numerical algorithms in any of the following trigonometric instructions. If you require a specific degree of accuracy, you must functionally test this instruction over your expected input domain.

Mnemonic	Name	Description	Logix Designer Application
ACOS	Arc Cosine	Compute the arc-cosine of a number.	Version 36 or later
ACS	Arc Cosine	Compute the arc-cosine of a number.	Version 35 or earlier
ASIN	Arc Sine	Compute the arc-sine of a number.	Version 36 or later
ASN	Arc Sine	Compute the arc-sine of a number.	Version 35 or earlier
ATAN	Arc Tangent	Compute the arc tangent in radians of y/x based on the sign of a value to determine the correct quadrant.	Version 36 or later
ATN	Arc Tangent	Compute the arc tangent in radians of y/x based on the sign of a value to determine the correct quadrant.	Version 35 or earlier
ATAN2	Arc Tangent 2	Compute the arc tangent in radians of y/x based on the signs of both values to determine the correct quadrant.	All versions
COS	Cosine	Compute the cosine of a number.	All versions
SIN	Sine	Compute the sin of a number.	All versions
TAN	Tangent	Compute the tangent of a number.	All versions

I/O

For special considerations when using the GSV and SSV instructions, see the controller user manual.

Mnemonic	Name	Description	Logix Designer Application
GSV	Get System Value	Get controller status information.	All versions
SSV	Set System Value	Set controller status information.	All versions

Math Conversion

Mnemonic	Name	Description	Logix Designer Application
DEG	Degrees	Convert radians into degrees.	All versions
BCD_TO	Convert to Integer	Convert the BCD value to an integer value.	Version 36 or later
FRD	Convert to Integer	Convert the BCD value to an integer value.	Version 35 or earlier
RAD	Radian	Convert degrees into radians.	All versions
TO_BCD	Convert to BCD	Convert the integer value to a BCD value.	Version 36 or later
TOD	Convert to BCD	Convert the integer value to a BCD value.	Version 35 or earlier
TRUNC	Truncate	Remove the fractional part of a value.	Version 36 or later
TRN	Truncate	Remove the fractional part of a value.	Version 35 or earlier

Safety Reaction Times

The input reaction time is the time from when the signal changes on an input terminal to when safety data is sent to the controller.

The output reaction time is the time from when safety data is received from the controller to when the output terminal changes state.

For information on how to determine the input and output reaction times, see the product documentation for your specific safety I/O device.

Connection Reaction Time Limit

The Connection Reaction Time Limit (CRTL) is the maximum age of safety packets on the associated connection. If the age of the data that is used by the consuming device exceeds the CRTL, a connection fault occurs.

The CRTL is defined by the values in the following table.

Table 11. Connection Reaction Time Limit Values

Value	Default	Description
Requested packet interval (RPI)	10 ms (input RPI)	The RPI determines how often the input and output packets are placed on the network.
Timeout multiplier	2	The timeout multiplier determines the number of retries before a timeout occurs. **IMPORTANT:** For any safety connection, do not set the timeout multiplier lower than 2. If the timeout multiplier is lower than 2, you risk the controller not meeting time coordination requirements during a loss of redundancy.
Network delay multiplier	200%	The network delay multiplier accounts for known delays on the network. When known delays occur, timeouts can be avoided with this parameter.

If you adjust these values, then you can adjust the CRTL. If a valid packet is not received within the CRTL, the safety connection times out, and the input and output data is placed in the safe state (Off).

IMPORTANT: The default values generate an input connection reaction time limit of 40 ms. If no edits are made to the defaults, verify this connection reaction time limit is used in the safety reaction time calculations.

IMPORTANT: For applications with safety I/O, the default connection reaction time limit can result in connection loss to the safety I/O modules. In some cases, it is necessary to increase the values from their defaults. Make sure that you use the new connection reaction time limit in the safety reaction time calculations.

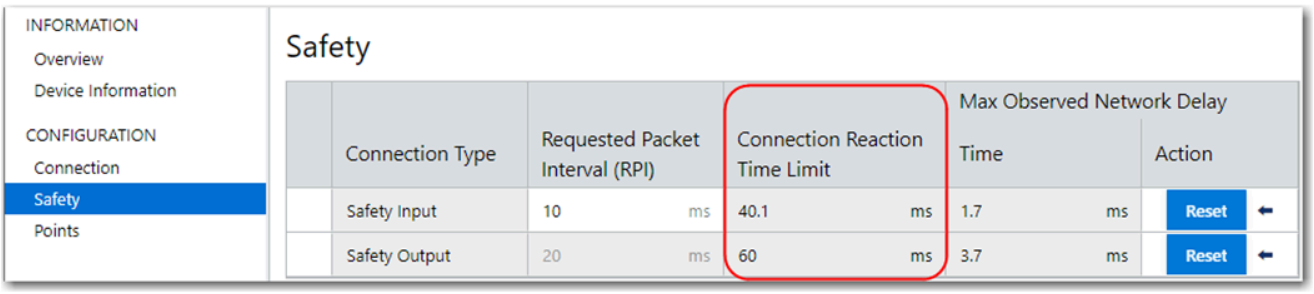
The following equations determine the CRTL:

Input Connection Reaction Time Limit = Input RPI x [Timeout Multiplier + Network Delay Multiplier]

Output Connection Reaction Time Limit = Safety Task Period x [Timeout Multiplier + Network Delay Multiplier - 1]

The CRTL is shown on the Safety view of the Module Properties dialog box.

Figure 41. Connection Reaction Time Limit



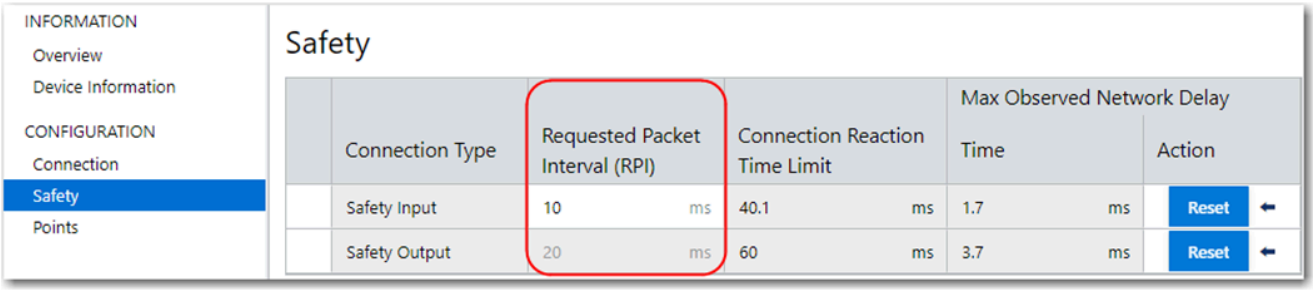
Specify the Requested Packet Interval (RPI)

The RPI specifies the period that data updates over a connection. For example, an input module produces data at the RPI that you assign.

For safety input connections, you can set the RPI on the Safety view of the Module Properties dialog box. The RPI is entered in 1 ms increments.

The CRTL is adjusted immediately when you change the RPI value.

Figure 42. Requested Packet Interval



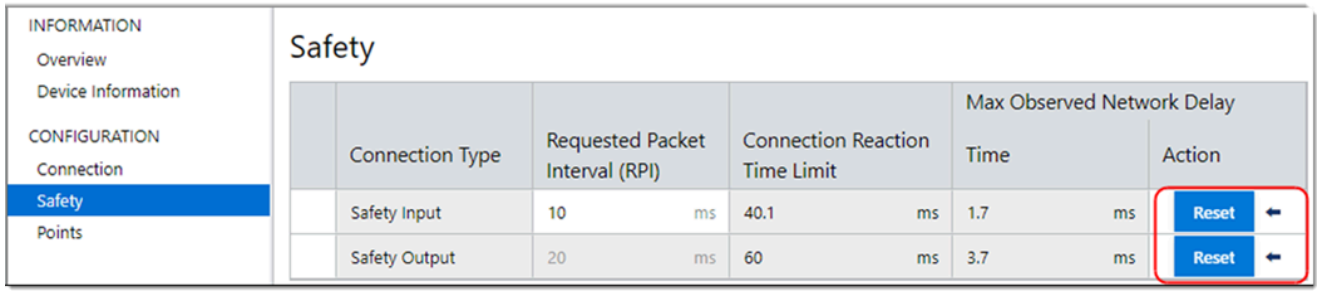
For safety output connections, the RPI is fixed at the safety task period. If the corresponding CTRL is not satisfactory, you can adjust the safety task period on the Safety Task Properties dialog box.

For typical applications, the default CRTL for input connections of 4 x RPI and the default CRTL for output connections of 3 x RPI is usually sufficient. For more complex requirements, use the Advanced button to modify the CTRL parameters.

View the Maximum Observed Network Delay

The maximum observed network delay is shown on the Safety view of the Module Properties dialog box. When online, click Reset to reset the maximum observed network delay.

Figure 43. Reset the Max Observed Network Delay

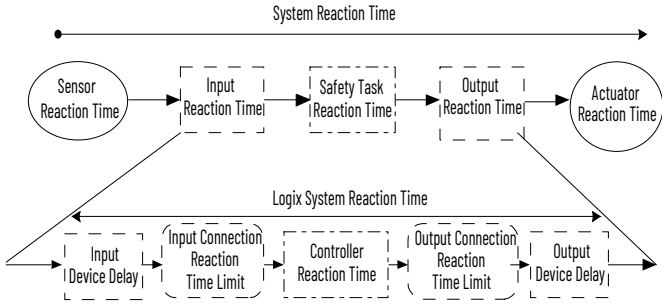


System Reaction Time

To determine the system reaction time of any control chain, add the reaction times of all the components of the safety chain:

System Reaction Time = Sensor Reaction Time + Logix System Reaction Time + Actuator Reaction Time

Figure 44. System Reaction Time



Safety Task Reaction Time

The safety task reaction time is the worst-case delay from any input change that is presented to the controller until the output producer sets the processed output. To determine the safety task reaction time, use the following equation:

Safety task reaction time = (safety task period + safety task watchdog) × 1.01

The multiplier is for potential clock drift.

IMPORTANT: For high demand safety functions, you must include safety function muting time in your safety reaction time calculations. To determine the muting time to add to your calculations, see High Demand Considerations on page 18.

Safety Task Period and Safety Task Watchdog

The safety task period is the interval at which the safety task executes.

The safety task watchdog time is the maximum permissible time for safety task processing. If the time to process a safety task exceeds the safety task watchdog time, a nonrecoverable safety fault occurs in the controller, which results in a transition to the safe state (Off).

You define the safety task watchdog time, which must be less than or equal to the safety task period. You can set the safety task watchdog time in the safety task properties. This value can be modified online, regardless of controller mode, but it cannot be changed when the controller is safety-locked or once a safety signature is created.

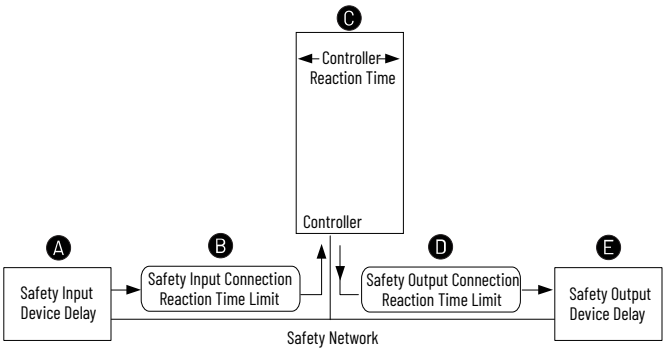
Logix System Reaction Time

The following sections provide information on how to calculate the Logix system reaction time.

Simple Input-Logic-Output Chain

This section describes the Logix system reaction time for any simple input-logic-output chain.

Figure 45. Worst Case Reaction Time



The Logix system reaction time for any simple input to logic to output chain consists of the following components.

Item	Description
A	Safety input device reaction time plus input delay time, if applicable.
B	Safety input connection reaction time limit. Read from the Module Properties dialog box in the Logix Designer application, this value is a multiple of the safety input device connection RPI.
C	Controller reaction time. See Safety Task Reaction Time on page 93.
D	Safety output connection reaction time limit. Read from the Module Properties dialog box in the Logix Designer application, this value is a multiple of the safety task period.
E	Safety output device reaction time.

Factors That Affect Logix Reaction-time Components

A number of factors can influence the Logix system reaction time components.

Network traffic and EMC create a lower limit for the values that you can successfully use for the timeout multiplier and network delay multiplier.

Table 12. System Reaction Time Components

Reaction Time Component	Factors
Safety task reaction time	Safety function muting during qualification, loss of redundancy, or a lock for update impacts the safety task reaction time in high demand applications.

Table 12. System Reaction Time Components (continued)

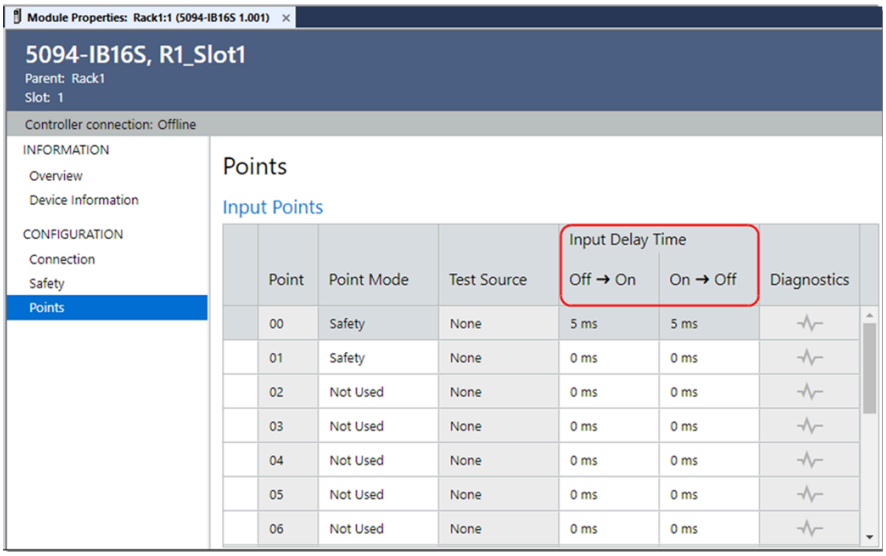
Reaction Time Component	Factors
	See Safety Function Muting on page 17.
Input device delay	• Input device reaction time • On-Off and Off-On delay settings for each input channel
Safety input connection reaction time limit	• Input device settings: - Requested packet interval (RPI) - Timeout multiplier - Network delay multiplier • The amount of network communication traffic. • The EMC environment of the system
Safety task period and safety task watchdog	• Safety task period setting • Safety task watchdog setting • The number and execution time of instructions in the safety task The instructions in your safety task create a lower limit for the values that you can successfully use for the safety task period and safety task watchdog.
Output connection reaction time limit	• Safety task period setting • Output device settings: - Timeout multiplier - Network delay multiplier • The amount of network communication traffic • The EMC environment of the system
Output module delay	Output module reaction time

Configure the Safety Input Module Delay Time

To configure input module delay time in the Logix Designer application, follow these steps.

1. In the configuration tree, right-click the safety input module and select Properties.
2. In the navigation pane, select Points.
3. Adjust the input delay time as required for your application.

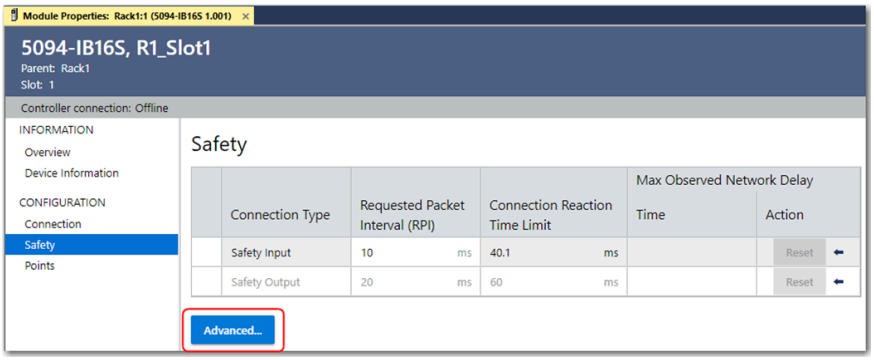
Figure 46. Input Delay Time



Configure the Input and Output Safety Connection Reaction Time Limits

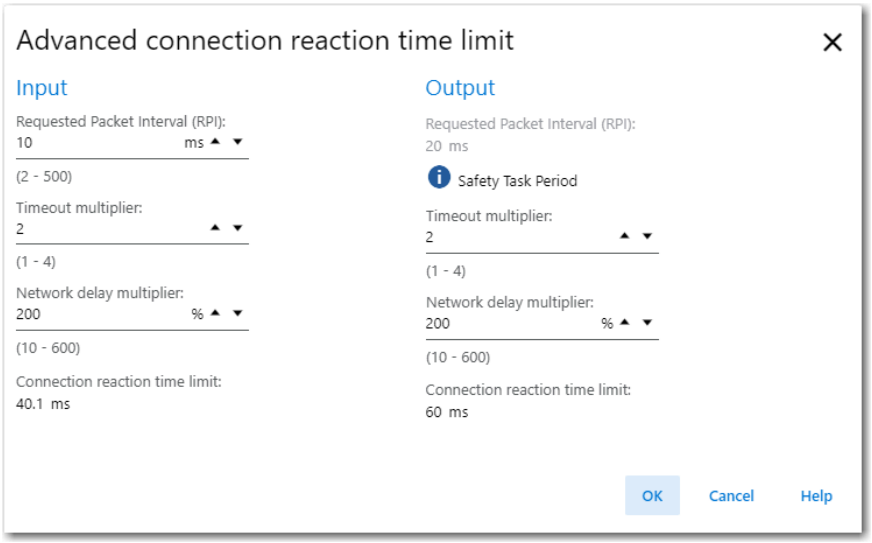
To configure connection reaction time limit settings, follow these steps.

1. In the configuration tree, right-click your safety I/O module and select Properties.
2. In navigation pane, select Safety.
3. On the Safety view, click Advanced.



4. On the Advanced Connection Reaction Time Limit dialog box, configure the parameters and click OK.

IMPORTANT: The values you define for the timeout multiplier and the network delay multiplier provide resilience for variations in network reliability and performance. Use caution when reducing the values as this increases the likelihood of false trips.



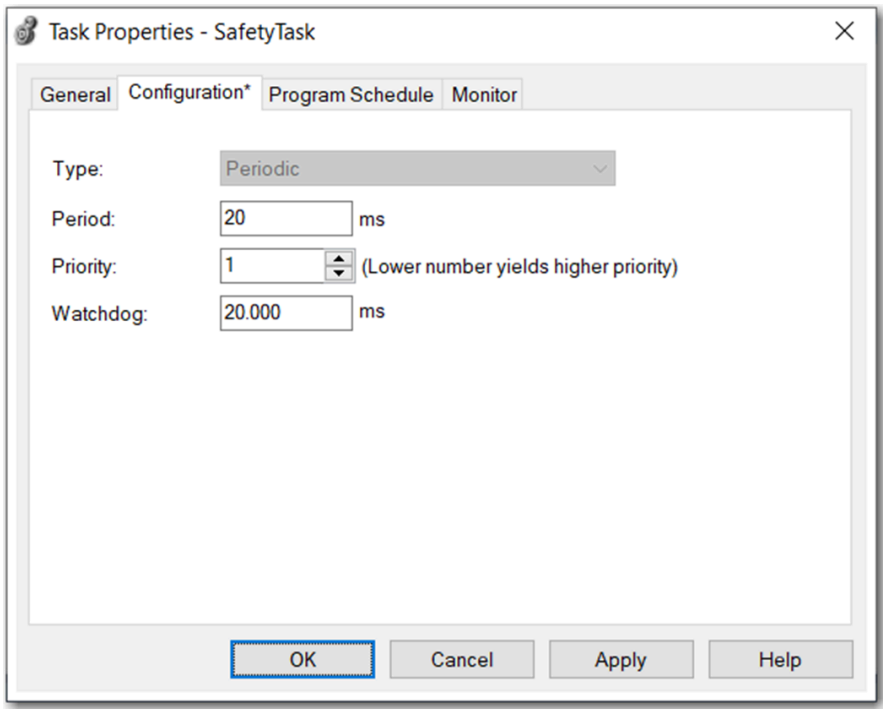
Configure the Safety Task Period and Watchdog

The safety task is a periodic timed task.

To configure the safety task period and watchdog, follow these steps.

1. In the configuration tree, right-click the safety task.
2. On the Task Properties dialog box, select the Configuration tab.
3. Configure the parameters as described in Safety Task on page 40.

Figure 47. Safety Task Configuration



Checklists for Logix SIS Applications

Checklists for safety applications are required to plan, program, and start a safety application. Use the checklists as planning guides and during project validation testing. When you use the checklists as planning guides, they can be saved as a record of the plan.

The checklists provide a sample of safety considerations and are not intended to be a complete list of items to verify. Your particular safety application can have additional safety requirements.

Make copies of the checklists to keep for future use.

Checklist for the Controller System

✓	System Requirements
	Are you using only the certified components for your SIL level, with the corresponding firmware release, as listed at https://rok.auto/certifications?
	Have you calculated the safety response time of the system for each safety function?
	Does the response time of the system include both the user-defined safety-task program watchdog (software watchdog) time and the safety task rate/period?
	Is the system response time in proper relation to the process safety time?
	Have probability (PFD/PFH) values been calculated for each safety function?
	Have you performed all appropriate project validation tests?
	If necessary, have you created a prescan routine to initialize safety critical data?
	Have you determined how your system can handle faults?
	Does each network in the safety system have a unique safety network number?
	Is each safety device configured with the correct safety network number?
	Have you generated a safety signature?
	Have you uploaded and recorded the safety signature for future comparison?
	After a download, have you verified that the safety signature in the controller matches the recorded safety signature?
	Do you have an alternate mechanism in place to preserve the safety integrity of the system when making online edits?

✓	System Requirements
	Have you considered the checklists for using SIL inputs and outputs?
	Have you considered safety function muting time in your process safety time?

Checklist for Safety Inputs

For programming or startup, an individual checklist can be completed for every safety input in the system. This method is the only way to make sure that the requirements are fully and clearly implemented. This checklist can also be used as documentation on the connection of external wiring to the application program.

✓	Input Device Requirements
	Have you followed installation instructions and precautions to conform to applicable safety standards?
	Have you performed project validation tests on the system and devices?
	Are control, diagnostics, and alarm functions performed in sequence in application logic?
	Have you uploaded and compared the configuration of each device to the configuration sent by the configuration tool?
	Are devices wired in compliance with the target standard and required safety level?
	Have you verified that the electrical specifications of the sensor and input are compatible?

Checklist for Safety Outputs

For programming or startup, an individual requirement checklist must be completed for every safety output in the system. This method is the only way to make sure that the requirements are fully and clearly implemented. This checklist can also be used as documentation on the connection of external wiring to the application program.

✓	Output Device Requirements
	Have you followed installation instructions and precautions to conform to applicable safety standards?
	Have you performed project validation tests on the system and devices?
	Have you uploaded and compared the configuration of each device to the configuration sent by the configuration tool?
	Have you verified that test outputs are not used as safety outputs?

✓	Output Device Requirements
	Are devices wired in compliance with the target standard and required safety level?
	Have you verified that the electrical specifications of the output and the actuator are compatible?

Checklist to Develop a Safety Application Program

Use the following checklist to help maintain safety when you create or modify a safety application program.

✓	System Requirements
	Were the programming guidelines followed during the creation of the safety application program?
	Does the safety application program contain only a ladder diagram?
	Does the safety application program contain only safety instructions?
	Does the safety application program clearly differentiate between safety and standard tags?
	Are only safety tags used for safety routines?
	Have you verified that safety routines do not attempt to read from or write to standard tags?
	Have you verified that no safety tags are aliased to standard tags and vice versa?
	Is each safety output tag correctly configured and connected to a physical output channel?
	Have you verified that all mapped tags have been conditioned in safety application logic?
	Have you defined the process parameters that the fault routines monitor?
	Have you sealed any safety Add-On Instructions with an instruction signature and recorded the safety instruction signature? Optional for one time use Add-On Instructions. Required Add-On Instructions are reused on different applications.
	Did you measure potential alarm bursts during system commissioning and change the project if measured scan times are not acceptable?
	Has an independent safety reviewer reviewed the program, if necessary?
	Has the review been documented and signed?

✓	System Requirements
	If your application is SIL 3 or high-demand SIL 2, did you use the Redundancy Status bit (S:R) in program logic to monitor redundant system operation?

Rockwell Automation Support

Use these resources to access support information.

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

Documentation Feedback

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

Waste Electrical and Electronic Equipment (WEEE)

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

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

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.

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

1756-RM015C-EN-P December 2025
Supercedes 1756-RM015B-EN-P September 2025