Logix 5000 Controllers
Information and Status

1756 ControlLogix, 1756 GuardLogix, 1769 CompactLogix,
1769 Compact GuardLogix, 1789 SoftLogix, 5069
CompactLogix, 5069 Compact GuardLogix, Studio 5000
Logix Emulate
Important User Information

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

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

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

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

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

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

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

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

---

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

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

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

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

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

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

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

This manual includes new and updated information. Use these reference tables to locate changed information.

Grammatical and editorial style changes are not included in this summary.

Global changes

<table>
<thead>
<tr>
<th>Change</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updated branding.</td>
<td>Throughout</td>
</tr>
</tbody>
</table>

New or enhanced features

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

<table>
<thead>
<tr>
<th>Topic Name</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legal notices on page 8</td>
<td>Updated Legal notices</td>
</tr>
<tr>
<td>Controller log on page 23</td>
<td>Updated the controller log information for the Compact GuardLogix 5370, CompactLogix 5370, ControlLogix 5570, GuardLogix 5570 Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, and GuardLogix 5580 controllers.</td>
</tr>
<tr>
<td>Controller log buffer on page 26</td>
<td>Updated the controller log buffer information for the Compact GuardLogix 5370, CompactLogix 5370, ControlLogix 5570, GuardLogix 5570 Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, and GuardLogix 5580 controllers.</td>
</tr>
<tr>
<td>Controller log events on page 36</td>
<td>Added the information logged for USB cable connected and USB cable disconnected entries.</td>
</tr>
<tr>
<td>Entries captured in the controller log on page 25</td>
<td>Added USB cable connected and disconnected to the list.</td>
</tr>
</tbody>
</table>
# Summary of Changes

In this manual ................................................................. 7

# Preface

Studio 5000 environment .................................................... 7
Additional Resources .......................................................... 7
Legal notices ........................................................................... 8

## Connections

Introduction .............................................................................. 9
Inhibit a Connection .................................................................. 9
Manage a connection failure ................................................... 11
  Configure a major fault to occur ........................................ 12
  Monitor the health of a module ........................................... 13

## Determine controller memory information

Introduction .............................................................................. 15
View data usage ......................................................................... 15
  Capacity tab for controllers that divide memory .................. 16
  Capacity tab for controllers with common memory area ....... 16
  Estimate memory information offline for controllers with divided memory ............................................. 17
  View run-time memory information for controllers with divided memory ..................................................... 18
  View the maximum and used Ethernet nodes for a controller ................................................................. 19
Write logic to get memory information ................................... 20
  Get memory information from the controller ...................... 20
  Choose the memory information ......................................... 21
  Convert INTs to a DINT ......................................................... 22

## Controller logging

Introduction .............................................................................. 23
Controller log ........................................................................... 23
  Controller log header .......................................................... 24
  Controller log entry ............................................................. 24
  Entries captured in the controller log ................................. 25
  Controller log buffer ........................................................... 26
  Controller log files and the removable media .................... 26
  Writing the controller log to the SD card ............................ 26
  Automatic save ..................................................................... 27
  Save on demand ................................................................. 28
Controller logging counters ................................................... 29
  Total entry count ............................................................... 29
Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsaved entry count</td>
<td>30</td>
</tr>
<tr>
<td>Execution modification count</td>
<td>30</td>
</tr>
<tr>
<td>Log file storage</td>
<td>33</td>
</tr>
<tr>
<td>Log file format</td>
<td>34</td>
</tr>
<tr>
<td>Create Custom Log Entries</td>
<td>35</td>
</tr>
<tr>
<td>Sample Ladder Logic File</td>
<td>36</td>
</tr>
<tr>
<td>Controller Log Events</td>
<td>36</td>
</tr>
</tbody>
</table>

**Chapter 4**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>45</td>
</tr>
<tr>
<td>Controller change detection</td>
<td>45</td>
</tr>
<tr>
<td>ChangesToDetect</td>
<td>45</td>
</tr>
<tr>
<td>AuditValue</td>
<td>47</td>
</tr>
<tr>
<td>ChangesToDetect format</td>
<td>47</td>
</tr>
<tr>
<td>Change detection in the Logix Designer application</td>
<td>49</td>
</tr>
<tr>
<td>Component tracking</td>
<td>49</td>
</tr>
<tr>
<td>Track the state of components and constant tags</td>
<td>51</td>
</tr>
<tr>
<td>View tracked components</td>
<td>51</td>
</tr>
<tr>
<td>Retrieve the tracked state value using a Message instruction</td>
<td>51</td>
</tr>
</tbody>
</table>

**Chapter 5**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>53</td>
</tr>
<tr>
<td>Status of S:FS when the project has an SFC</td>
<td>53</td>
</tr>
<tr>
<td>Get and set system data</td>
<td>54</td>
</tr>
</tbody>
</table>

**Index**
This manual describes how Logix 5000 controllers use connections with other devices. This manual also describes status keywords and how to get controller information, such as memory resources. This manual is one of a set of related manuals that show common procedures for programming and operating Logix 5000 controllers.

For a complete list of common procedures manuals, refer to the Logix 5000 Controllers Common Procedures Programming Manual, publication 1756-PM001.

The term Logix 5000 controller refers to any controller based on the Logix 5000 operating system.

The Studio 5000 Automation Engineering & Design Environment® combines engineering and design elements into a common environment. The first element is the Studio 5000 Logix Designer® application. The Logix Designer application is the rebranding of RSLogix 5000® software and will continue to be the product to program Logix 5000™ controllers for discrete, process, batch, motion, safety, and drive-based solutions.

The Studio 5000® environment is the foundation for the future of Rockwell Automation® engineering design tools and capabilities. The Studio 5000 environment is the one place for design engineers to develop all elements of their control system.

These documents contain additional information concerning related Rockwell Automation products.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Industrial Automation Wiring and Grounding Guidelines, publication, 1770-4.1</td>
<td>Provides general guidelines for installing a Rockwell Automation industrial system.</td>
</tr>
</tbody>
</table>

View or download publications at http://www.rockwellautomation.com/literature. To order paper copies of
Legal notices

technical documentation, contact a local Rockwell Automation distributor or sales representative.

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

End User License Agreement (EULA)

You can view the Rockwell Automation End User License Agreement (EULA) by opening the license.rtf file located in your product's install folder on your hard drive.

The default location of this file is:

C:\Program Files (x86)\Common Files\Rockwell\license.rtf.

Open Source Software Licenses

The software included in this product contains copyrighted software that is licensed under one or more open source licenses.

You can view a full list of all open source software used in this product and their corresponding licenses by opening the oss_license.txt file located your product's OPENSOURCE folder on your hard drive. This file is divided into these sections:

- Components
  Includes the name of the open source component, its version number, and the type of license.
- Copyright Text
  Includes the name of the open source component, its version number, and the copyright declaration.
- Licenses
  Includes the name of the license, the list of open source components citing the license, and the terms of the license.

The default location of this file is:

C:\Program Files (x86)\Common Files\Rockwell\Help\<product name>\Release Notes\OPENSOURCE\oss_licenses.txt.

You may obtain Corresponding Source code for open source packages included in this product from their respective project web site(s). Alternatively, you may obtain complete Corresponding Source code by contacting Rockwell Automation via the Contact form on the Rockwell Automation website: http://www.rockwellautomation.com/global/about-us/contact/contact.page. Please include “Open Source” as part of the request text.
Connections

Introduction

A Logix 5000 controller uses connections for most, but not all, of its communication with other devices.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
</table>
| Connection                | A communication link between two devices, such as between a controller and an I/O module, PanelView terminal, or another controller. Connections are allocations of resources that provide more reliable communication between devices than unconnected messages. The number of connections that a single controller can have is limited. Configuring the controller to communicate with other devices in the system indirectly determines the number of connections the controller uses. These communication types use these connections:  
  • I/O modules  
  • Produced and consumed tags  
  • Program parameters  
  • Certain types of Message (MSG) instructions (not all types use a connection) |
| Requested packet interval (RPI) | The RPI specifies the period at which data updates over a connection. For example, an input module sends data to a controller at the RPI assigned to the module.  
  • Typically, an RPI is configured in milliseconds (ms). The range is 0.2 ms (200 microseconds)...750 ms.  
  • If a ControlNet network connects the devices, the RPI reserves a slot in the stream of data flowing across the ControlNet network. The timing of this slot may not coincide with the exact value of the RPI, but the control system guarantees that the data transfers at least as often as the RPI. |
| Path                      | The path describes the route that a connection takes to get to the destination. Typically, the path for a connection is defined when a device is added to the I/O Configuration folder of the controller. |

Inhibit a Connection

In some situations, such as when initially commissioning a system, it is useful to disable portions of a control system and enable them as they are connected to the control system. Inhibiting individual modules or groups of modules prevents the controller from trying to communicate with the modules.

ATTENTION: Inhibiting a module breaks the connection to the module and prevents communication of I/O data.
An installed I/O module defaults to not inhibited. To inhibit a module, change an individual module’s properties.

![Module Properties Report](image)

<table>
<thead>
<tr>
<th>To</th>
<th>Then</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communicate with the module</td>
<td>Do not inhibit the module. Clear the Inhibit Module check box.</td>
</tr>
<tr>
<td>Prevent communication with the module</td>
<td>Inhibit the module. Select the Inhibit Module check box.</td>
</tr>
</tbody>
</table>

When a communication bridge module is inhibited, the controller shuts down the connections to the bridge module and to all the modules that depend on that bridge module. Inhibit a communication bridge module to disable an entire branch of the I/O network.

When the module is inhibited, the Controller Organizer displays a yellow attention symbol over the module.

<table>
<thead>
<tr>
<th>If</th>
<th>And you</th>
<th>And</th>
<th>Then</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline</td>
<td></td>
<td>------&gt;</td>
<td>The inhibit status is stored in the project. When the project downloads to the controller, the module still is inhibited.</td>
</tr>
<tr>
<td>Online</td>
<td>Inhibit a module while connected to the module</td>
<td>------&gt;</td>
<td>The connection to the module is closed. The module’s outputs go to the last configured Program mode.</td>
</tr>
<tr>
<td>Inhibit a module but a connection to the module was not established (perhaps due to an error condition or fault)</td>
<td>------&gt;</td>
<td>The module is inhibited. The module status information changes to indicate that the module is inhibited and not faulted.</td>
<td></td>
</tr>
<tr>
<td>Uninhibit a module (clear the check box)</td>
<td>No fault occurs</td>
<td></td>
<td>A connection is made to the module and the module is dynamically reconfigured (if the controller is the owner-controller) with the configuration created for that module. If the controller is configured for listen-only, it cannot reconfigure the module.</td>
</tr>
<tr>
<td>Fault occurs</td>
<td></td>
<td></td>
<td>A connection is not made to the module. The module status information changes to indicate the fault condition.</td>
</tr>
</tbody>
</table>

**To inhibit or uninhibit a connection**

1. Use a Get System Value (GSV) instruction to read the Mode attribute for the module.
2. To inhibit the module, set bit 2. To uninhibit the module, clear bit 2.
3. Use a Set System Value (SSV) instruction to write the Mode attribute back to the module.
EXAMPLE  Inhibit a connection
If Module_1.Inhibit = 1, then inhibit the operation of the I/O module named Module_1.

1. The GSV instruction sets Module_1.Mode = value of the Mode attribute for the module.
2. The OTE instruction sets bit 2 of Module_1.Mode = 1. This means inhibit the connection.
3. The SSV instruction sets the Mode attribute for the module = Module_1.Mode.

ATTENTION: Outputs respond to the last, non-faulted state of the controlling inputs. To avoid potential injury and damage to machinery, make sure this does not create an unsafe operation. Configure critical I/O modules to generate a controller major fault when they lose their connections to the controller, or monitor the status of I/O modules.

EXAMPLE  Loss of Communication
Controller B requires data from controller A. If communication fails between the controllers, controller B continues to act on the last data that it received from controller A.
If communication with a device in the I/O configuration of the controller does not occur for 100 ms, the communication times out. If this occurs, choose between these options.

<table>
<thead>
<tr>
<th>If the controller should</th>
<th>Then</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault (major fault)</td>
<td>Configure a major fault to occur</td>
</tr>
<tr>
<td>Continue operating</td>
<td>Monitor the health of a module</td>
</tr>
</tbody>
</table>

**See also**

- [Configure a major fault to occur](#) on page 12
- [Monitor the health of a module](#) on page 13

**Configure a major fault to occur**

To force a module to generate a major fault in the controller if it loses connection with the controller, configure a major fault to occur. This interrupts the execution of logic and runs the Controller Fault Handler. If the Controller Fault Handler does not clear the fault, the controller shuts down.
When selecting **Major Fault On Controller...Run Mode**, the controller:

- Must be connected to the module during the Program transition to Run mode. During the Program to Run mode transition, there can be a 20-second delay.

  During this delay, the controller makes one attempt to connect to a module. If **Major Fault On Controller...Run Mode** is selected and no connection occurs during the 20-second delay, a fault occurs because at least one required connection is not established before going to Run mode. This is a 3/23 type fault code. This fault can occur in large systems with networked I/O.

- Registers a fault if the connection is dropped while in Run mode. A required I/O module connection failed, creating a 3/16 type fault.

For fault codes, see the *Logix 5000 Controllers Major and Minor Faults Programming Manual*, publication no. 1756-PM014.

### See also

*Logix 5000 Controllers Major and Minor Faults Programming Manual*, publication no. 1756-PM014

**Monitor the health of a module**

Monitor the module status if the module is not configured to generate a major fault. If a module loses its connection to the controller, outputs go to their configured faulted state. The controller and other I/O modules continue to operate based on old data from the module.

If communication with a module times out, the controller produces the following warnings.

- The I/O status indicator on the front of the controller flashes green.
- A Warning (⚠️) icon shows over the I/O configuration folder and over the device that has timed out.
- A module fault code is produced, which is accessible through:
  - Module Properties window for the module.
  - GSV instruction.

To monitor the health of connections, use a Get System Value (GSV) instruction to monitor the Module object for either the controller or a specific module.

<table>
<thead>
<tr>
<th>To determine if communication has timed out with any device</th>
<th>Get this attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEDStatus</td>
<td>INT</td>
<td>For efficiency, use a DINT as the destination data type.</td>
<td>Current state of the I/O status indicator on the front of the controller. No need to enter an instance name with this attribute. This attribute applies to the entire collection of modules.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Status Indicator off. No Module objects are configured for the controller (there are no modules in the I/O Configuration section of the controller organizer).</td>
</tr>
</tbody>
</table>
### Chapter 1  Connections

<table>
<thead>
<tr>
<th>To</th>
<th>Get this attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 Flashing red. None of the Module objects are Running.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 Flashing green. At least one Module object is not Running.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 Solid green. All the Module objects are Running.</td>
</tr>
</tbody>
</table>

Determine if communication has timed out with a specific device

| FaultCode | INT       | For efficiency, use a DINT as the destination data type. | A number that identifies a module fault, if one occurs. |

In the **Instance Name**, choose the device with a connection to monitor. Make sure to assign a name to the device in the I/O Configuration folder of the project.

If **Module_Status** is any value other than 4, the controller is not communicating with the module. See the following example.

**EXAMPLE** This rung checks the status of an I/O connection. The controller checks the entry status of the connection; any value other than 4 indicates that the connection is not functioning correctly. When the controller detects an error, the error code and information is trapped, and the controller tries to re-establish the connection.

![Diagram](image-url)
## Determine controller memory information

### Introduction

Depending on the type of controller, the memory of the controller is divided into several areas.

<table>
<thead>
<tr>
<th>This controller</th>
<th>Stores this</th>
<th>In this memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>ControlLogix 5570</td>
<td>I/O tags</td>
<td>I/O memory</td>
</tr>
<tr>
<td>GuardLogix 5570S</td>
<td>Produced / Consumed tags</td>
<td></td>
</tr>
<tr>
<td>CompactLogix 5370</td>
<td>Communication via Message (MSG) instructions</td>
<td></td>
</tr>
<tr>
<td>Compact GuardLogix 5370S</td>
<td>Communication with workstations</td>
<td></td>
</tr>
<tr>
<td>ControlLogix 1756-L5x, L6x</td>
<td>Communication with polled (OPC/DDE) tags that use RSLinx software¹</td>
<td></td>
</tr>
<tr>
<td>GuardLogix 1756-L6xS</td>
<td>Tags other than I/O, produced, or consumed tags</td>
<td>Data and logic memory²</td>
</tr>
<tr>
<td></td>
<td>Logic routines</td>
<td></td>
</tr>
<tr>
<td>CompactLogix 1769-L2x, L3x</td>
<td>Communication with polled (OPC/DDE) tags that use RSLinx software²</td>
<td></td>
</tr>
<tr>
<td>SoftLogix5800</td>
<td>These controllers do not divide their memory. They store all elements in one common memory area.</td>
<td></td>
</tr>
<tr>
<td>CompactLogix 5380</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CompactGuardLogix 5380</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CompactLogix 5480</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ControlLogix 5580</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GuardLogix5Logix5580</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(1) To communicate with polled tags, the controller uses both I/O data and logic memory.

(2) 1756-L55M16 controllers have an additional memory section for logic.

In the Logix Designer application, the **Capacity** tab (formerly the **Memory** tab) on the **Controller Properties** dialog box shows data usage in the controller. The data displayed on the **Capacity** tab depends on the controller.

**Tip:** Refer to the **Memory information table** to see whether a controller divides its memory or has one common memory area.

### View data usage

See also

[Memory information table](#) on page 15
Chapter 2  Determine controller memory information

Capacity tab for controllers that divide memory

The Capacity tab for controllers that divide memory shows I/O memory and Data and Logic memory separately. For each area of controller memory, the tab shows the consumed and available memory in bytes.

The tab provides an estimation tool to estimate the amount of controller memory a project requires for I/O, Data, and Logic. It also includes a Max Used field for each type of memory to show peak memory usage as communication occurs.

![Capacity tab for controllers that divide memory](image)

Capacity tab for controllers with common memory area

The Capacity tab for controllers that store elements in one common area differs slightly from the Capacity tab for other controllers.

Instead of describing consumed memory in bytes, the Capacity tab shows logical program blocks which represent units of executable code. The Capacity tab shows project size in blocks compared to the specified product capacity in blocks to determine if the project can be downloaded to the controller. If the size of the project exceeds the controller capacity available, a message appears that states the overage size in blocks. When the project size exceeds the memory available, the project does not download to the controller.

Block units are only used for application source code sizes. Data structures and messages are still described in bytes.

IMPORTANT Although block sizes are not physically equivalent to the bytes measured in controllers with divided memory controllers, in general, applications that fit in a divided memory controller should fit in the same controller with one common memory area.
For controllers with divided memory, the applications in the project have a close correlation between the size of the source code and the physical memory in the controller. To estimate how much controller memory a project requires, use the **Estimate** button on the **Capacity** tab. It estimates the number of bytes of:

- Free (unused) memory.
- Used memory.
- Largest free contiguous block of memory.

**Tip:** This section only applies to controllers with divided memory. Refer to the **Memory information table** for a list of controllers with divided memory. Refer to the **Capacity tab** controllers with common memory area for information about the other controllers.
To estimate memory information offline for controllers with divided memory

1. On the Online toolbar, click the Controller Properties icon.

2. On the Controller Properties dialog box, click the Capacity tab.
3. In the Estimated Data and Logic Memory area, view the memory information since the last estimate.
4. Click Estimate to re-estimate the amount of controller memory.
5. Click OK.

See also

Memory information table on page 15
Capacity tab for controllers with common memory area on page 16

View run-time memory information for controllers with divided memory

When online with a controller with divided memory, the Capacity tab shows the actual memory usage of the controller. While the controller is running, it uses additional memory for communication. The amount of memory the controller needs varies depending on the state of the communication.

The Capacity tab of the controller includes a Max Used entry for each type of memory. The Max Used values show the peak memory usage as communication occurs.

Tip: This section only applies to controllers with divided memory. Refer to the Memory information table for a list of controllers with divided memory. Refer to Capacity tab controllers with common memory area for information about the other controllers.
To view run-time memory information for controllers with divided memory

1. On the Online toolbar, click the Controller Properties icon.

2. On the Controller Properties dialog box, click the Capacity tab.
3. Click Reset All Max to reset values.
4. Click OK.

See also

- Memory information table on page 15
- Capacity tab for controllers with common memory area on page 16

View the maximum and used Ethernet nodes for a controller

Some controllers have limits on the number of Ethernet nodes they support in the I/O Configuration tree. When a device is added directly to the Ethernet I/O configuration, it is counted toward the node limitation of the controller.

The Capacity tab of the Controller Properties dialog box shows the current number of used Ethernet nodes in the I/O Configuration tree and the maximum number of Ethernet nodes supported by the controller.

When the controller is online, the controller enforces the node count, and the node count exceeds the limit:

- An error dialog box appears.
- The I/O module cannot be added to the I/O Configuration tree.

When the controller is offline, the controller enforces the node count, and the node count exceeds the limit when the project is verified or the controller attempts to go online:

- An error message is generated on the Error tab of the Output window.
- The project cannot be downloaded to the controller.

Tip: The 1756-L05E controllers do not enforce the node count, meaning the system does not prevent connecting additional nodes, and it may be possible to download the project. The system provides a warning when the recommended node limit has been exceeded.

To view the maximum and used Ethernet nodes for a controller:

1. On the Online toolbar, click the Controller Properties icon.
2. On the **Controller Properties** dialog box, click the **Capacity** tab.

3. In the **Ethernet Node** group, view the information about the Ethernet nodes.

- **Maximum/Recommended Maximum** - Number of Ethernet nodes supported by the controller, or the recommended maximum number of Ethernet nodes for the controller.

- **Used** - Current number of Ethernet nodes in the I/O Configuration tree

If the controller is offline, 🚭 appears next to the **Used** parameter when the node count exceeds the limit.

For controllers that do not limit the number of Ethernet nodes, ⚠ appears next to the **Used** parameter when the node count exceeds the recommended limit and this message appears:

*The recommended node limit has been exceeded. It may be possible to connect additional nodes depending on the node type and system architecture.*

### Write logic to get memory information

Use a Message (MSG) instruction to get memory information from the controller.

**Tip:** This procedure applies to controllers with divided memory. The Logix Designer application determines capacity information for controllers that store elements in one common area.

### Get memory information from the controller

To get memory information from the controller, execute a Message (MSG) instruction. This table lists configuration information for the instruction.

<table>
<thead>
<tr>
<th>On this tab</th>
<th>For this item</th>
<th>Type or select</th>
<th>Which means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
<td>Message Type</td>
<td>CIP Generic</td>
<td>Execute a Control and Information Protocol command.</td>
</tr>
<tr>
<td>Service Type</td>
<td>Custom</td>
<td>Create a CIP Generic message that is not available in the list.</td>
<td></td>
</tr>
<tr>
<td>Service Code</td>
<td>3</td>
<td>Read specific information about the controller (GetAttributeList service).</td>
<td></td>
</tr>
<tr>
<td>Class</td>
<td>72</td>
<td>Get information from the user memory object.</td>
<td></td>
</tr>
<tr>
<td>Instance</td>
<td>1</td>
<td>This object contains one instance.</td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>0</td>
<td>Null value.</td>
<td></td>
</tr>
</tbody>
</table>

$source\_array$ of type **SINT[12]**

<table>
<thead>
<tr>
<th>In this element</th>
<th>Enter</th>
<th>Which means</th>
</tr>
</thead>
<tbody>
<tr>
<td>$source_array[0]$</td>
<td>5</td>
<td>Get 5 attributes.</td>
</tr>
<tr>
<td>$source_array[1]$</td>
<td>0</td>
<td>Null value.</td>
</tr>
<tr>
<td>$source_array[3]$</td>
<td>0</td>
<td>Null value.</td>
</tr>
<tr>
<td>$source_array[5]$</td>
<td>0</td>
<td>Null value.</td>
</tr>
<tr>
<td>$source_array[6]$</td>
<td>5</td>
<td>Get largest contiguous block of additional free logic memory.</td>
</tr>
<tr>
<td>$source_array[7]$</td>
<td>0</td>
<td>Null value.</td>
</tr>
<tr>
<td>$source_array[9]$</td>
<td>0</td>
<td>Null value.</td>
</tr>
</tbody>
</table>
Choose the memory information

This table lists the information that the MSG instruction returns to INT_array (destination tag of the MSG).

Tip: This procedure applies to controllers with divided memory. The Logix Designer application determines capacity information for controllers that store elements in one common area.

<table>
<thead>
<tr>
<th>To find this information</th>
<th>Then copy these array elements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount of free I/O memory (32-bit words)</td>
<td>INT_array[3]</td>
<td>Lower 16 bits of the 32 bit value</td>
</tr>
<tr>
<td></td>
<td>INT_array[4]</td>
<td>Upper 16 bits of the 32 bit value</td>
</tr>
<tr>
<td>Amount of free data and logic memory (32-bit words)</td>
<td>INT_array[5]</td>
<td>Lower 16 bits of the 32 bit value</td>
</tr>
<tr>
<td></td>
<td>INT_array[6]</td>
<td>Upper 16 bits of the 32 bit value</td>
</tr>
<tr>
<td>1756-L55M16 controllers only—amount of additional free logic memory (32-bit words)</td>
<td>INT_array[7]</td>
<td>Lower 16 bits of the 32 bit value</td>
</tr>
<tr>
<td></td>
<td>INT_array[8]</td>
<td>Upper 16 bits of the 32 bit value</td>
</tr>
<tr>
<td>Total size of I/O memory (32-bit words)</td>
<td>INT_array[11]</td>
<td>Lower 16 bits of the 32 bit value</td>
</tr>
<tr>
<td></td>
<td>INT_array[12]</td>
<td>Upper 16 bits of the 32 bit value</td>
</tr>
<tr>
<td>Total size of data and logic memory (32-bit words)</td>
<td>INT_array[13]</td>
<td>Lower 16 bits of the 32 bit value</td>
</tr>
<tr>
<td></td>
<td>INT_array[14]</td>
<td>Upper 16 bits of the 32 bit value</td>
</tr>
<tr>
<td>1756-L55M16 controllers only—additional logic memory (32-bit words)</td>
<td>INT_array[15]</td>
<td>Lower 16 bits of the 32 bit value</td>
</tr>
<tr>
<td></td>
<td>INT_array[16]</td>
<td>Upper 16 bits of the 32 bit value</td>
</tr>
<tr>
<td>1756-L55M16 controllers only—largest contiguous block of additional free logic memory (32-bit words)</td>
<td>INT_array[19]</td>
<td>Lower 16 bits of the 32 bit value</td>
</tr>
<tr>
<td></td>
<td>INT_array[20]</td>
<td>Upper 16 bits of the 32 bit value</td>
</tr>
<tr>
<td>Largest contiguous block of free I/O memory (32-bit words)</td>
<td>INT_array[23]</td>
<td>Lower 16 bits of the 32 bit value</td>
</tr>
<tr>
<td></td>
<td>INT_array[24]</td>
<td>Upper 16 bits of the 32 bit value</td>
</tr>
<tr>
<td>Largest contiguous block of free data and logic memory (32-bit words)</td>
<td>INT_array[27]</td>
<td>Lower 16 bits of the 32 bit value</td>
</tr>
<tr>
<td></td>
<td>INT_array[28]</td>
<td>Upper 16 bits of the 32 bit value</td>
</tr>
</tbody>
</table>
Convert INTs to a DINT

The MSG instruction returns each memory value as two separate INTs.

- The first INT represents the lower 16 bits of the value.
- The second INT represents the upper 16 bits of the value.

To convert the separate INTs into one usable value, use a Copy (COP) instruction.

<table>
<thead>
<tr>
<th>In this operand</th>
<th>Specify</th>
<th>Which means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>First INT of the 2 element pair (lower 16 bits)</td>
<td>Start with the lower 16 bits.</td>
</tr>
<tr>
<td>Destination</td>
<td>DINT tag in which to store the 32-bit value</td>
<td>Copy the value to the DINT tag.</td>
</tr>
<tr>
<td>Length</td>
<td>1</td>
<td>Copy 1 times the number of bytes in the Destination data type. In this case, the instruction copies 4 bytes (32 bits), which combines the lower and upper 16 bits into one 32-bit value.</td>
</tr>
</tbody>
</table>

In this example, the COP instruction produces the 32-bit value that represents the amount of free I/O memory, in 32-bit words.

**EXAMPLE** Convert INTs to a DINT.

- Elements 3 of INT_array is the lower 16 bits of the amount of free I/O memory. Element 4 is the upper 16 bits.
- Memory_IO_Free is a DINT tag (32 bits) in which to store the value for the amount of free I/O memory.
- To copy all 32 bits, specify a Length of 1. This tells the instruction to copy 1 times the size of the Destination (32 bits). This copies both element 3 (16 bits) and element 4 (16 bits) and places the 32-bit result in Memory_IO_Free.

```
COP
  Copy File
Source INT_array[3]
Dest Memory_IO_Free
Length 1
```
Chapter 3

Controller logging

Introduction

Use the controller logging feature to detect and log changes made to Logix5000 controllers without adding any auditing software. With controller logging, the controllers:

- Detect changes and create logs entries containing information about the changes.
- Store the log entries to removable media for later review.
- Provide programmatic access to log entry counters to provide change detection information remotely.

Note these logging considerations:

- The 1769-L3x and 1769-L4x CompactLogix Controllers do not support storing log entries to removable media, and the audit value is not populated.
- The Audit Value is not supported in versions 19 and earlier.

Controller log

A controller log is a record of interactions that occurred in the controller due to physical conditions such as keyswitches, changes to removable media, fault conditions, and programming changes configured in the application.

- For the Compact GuardLogix 5370, CompactLogix 5370, ControlLogix 5570, and GuardLogix 5570 controllers, up to 100 log entries are buffered in the controller memory. The controller can save these buffered entries to removable media using a message instruction. Additionally, the controller can be configured to write buffered entries to the removable media after accumulating 80 entries.
- For the Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, and GuardLogix 5580 controllers, up to 500 log entries are buffered in the controller memory. The controller can be configured to write buffered entries to the removable media after accumulating 400 entries.

In the example table below, record numbers 4 through 6 are related to Change Detection.

<table>
<thead>
<tr>
<th>Record Number</th>
<th>Time</th>
<th>Entry Description</th>
<th>User Name</th>
<th>Workstation Name</th>
<th>FactoryTalk ID</th>
<th>Extended Information</th>
<th>Change Detection Audit Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12-Feb 03:39:34</td>
<td>Project download</td>
<td>John Doe</td>
<td>Laptop</td>
<td>FT\JDoe</td>
<td>Project L11</td>
<td>16#FD60_CB89_029F_3500</td>
</tr>
<tr>
<td>2</td>
<td>12-Feb 04:05:12</td>
<td>Forces Enabled</td>
<td>Jones</td>
<td>USMAYLT</td>
<td>FT\Jones</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>12-Feb 04:22:03</td>
<td>Online edits modified controller program</td>
<td>John Doe</td>
<td>Laptop</td>
<td>FT\JDoe</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 3  Controller logging

<table>
<thead>
<tr>
<th>Record Number</th>
<th>Time</th>
<th>Entry Description</th>
<th>User Name</th>
<th>Workstation Name</th>
<th>FactoryTalk ID</th>
<th>Extended Information</th>
<th>Change Detection Audit Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>12-Feb 04:42:12</td>
<td>Change Log entry added</td>
<td></td>
<td></td>
<td>FT\JDoe</td>
<td>16#FD60_CB89_029F</td>
<td>3521</td>
</tr>
<tr>
<td>5</td>
<td>12-Feb 04:50:43</td>
<td>Change detection mask modified</td>
<td>None</td>
<td>None</td>
<td>16#FFFF_FFFF_FFF_FFFF, New mask 16#FFFF_FFFF_FFC_FFFF</td>
<td>16#FD60_CB89_029F</td>
<td>3556</td>
</tr>
<tr>
<td>6</td>
<td>12-Feb 04:58:29</td>
<td>Change Log entry added</td>
<td>None</td>
<td>None</td>
<td></td>
<td>16#FD60_CB89_029F</td>
<td>3558F</td>
</tr>
</tbody>
</table>

**Controller log header**

When the controller creates a log file on the removable media, it includes some header information. The header information includes these items:

- Date the log file was created.
- Controller model number.
- Controller serial number.
- Version of firmware running on the controller.

**Controller log entry**

Each entry in the log can include this information.

- Record Number
- Time of Occurrence (UTC - 24 hour clock)
- Entry Description
- Windows User Name
- Workstation Name
- FactoryTalk User ID (if available)
- Extended Information
- Change Detection Audit Value

This example shows controller log entries in a spreadsheet. Controller log files are stored in plain text files in TSV format, so no special tools are required to...
These entries are detected and logged by the controller. These events are described in more detail in Controller Log Events.

Tip: In Logix Designer versions 30 and later, audit log entries also appear in the controller log. Audit log entries describe events such as changes to component properties, edits in a routine, and changes to tag values. The audit log is used for auditing by the FactoryTalk® AssetCentre.

- Project downloaded
- Project loaded from removable media
- Project stored to removable media
- Online edits modified controller program
- Partial import online completed or Transaction committed
- I/O forces enabled, disabled, removed, or modified
- SFC forces enabled, disabled, removed, or modified
- Firmware update
- Constant tag data changed
- Multiple constant tag data changed
- Change to constant tag configuration reset
- Mode change
- Major fault, major fault cleared
- Program properties modified
- Task properties modified
- Controller timeslice modified
- Removable media inserted or removed
- Safety signature created or deleted
- Safety locked or unlocked
- Custom entry: User-defined logic to create a log entry, with user-defined entry description and extended information
• Safety signature delete inhibited in Run mode
• Safety signature delete allowed in Run mode
• The Changes To Detect value has changed
• Log Collected Data Cleared
• Program Connection Modified
• USB cable connected and disconnected

See also

See Controller log events on page 36

Controller log buffer

For the Compact GuardLogix 5370, CompactLogix 5370, ControlLogix 5570, and GuardLogix 5570 controllers, the controller keeps up to 100 log entries buffered in its internal memory.

For the Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, and GuardLogix 5580 controllers, the controller keeps up to 500 log entries buffered in its internal memory.

Configure the controller to write the buffered entries to the removable media when its internal buffer is 80% full, or command the controller to write the buffered entries to the removable media with a message instruction. This procedure is detailed below. Once a log entry is written to the removable media, it is removed from the buffer.

If the removable media is not present, is full, or if the controller is not configured to automatically write buffered entries to the removable media, and the internal buffer becomes full, entries are saved in the buffer in a circular fashion. As new entries are stored, the oldest entries are discarded.

When written to the removable media, controller logs are stored in plain text files in the Tab Separated Value (TSV) format. Each time the controller writes entries to the CompactFlash card, the entries are appended to the text file until the file reaches 1 MB in size. At this point, the controller creates a new text file.

The controller does not attempt to write log entries to a full removable media card. If the removable media becomes full, the system behaves as if the removable media is not present.

Controller log files are stored in plain text files in the TSV format, so no special tools are required to read them. Open them in any text editor, or in a spreadsheet application such as Microsoft Excel.

Writing the controller log to the SD card

The controller log can be written to the Secure Digital (SD) card either automatically or on demand.

Tip: Some Logix 5000 controllers support additional types of removable media that can be used to write the controller log entries. Refer to the Logix 5000 controller documentation for information regarding the type of removable media the controller supports.
Configure the controller to automatically write buffered log entries to the SD card when the controller’s internal log entry buffer becomes 80% full. Alternately, configure the controller to write the buffered entries before a firmware update. To write buffered entries before a firmware update, send a message instruction to the controller using a CIP Generic message type and a service type of Controller Log Automatic Write Set. Sending a value of 0 turns off automatic writes, and sending a value of 1 turns on automatic writes. By default, entries are not automatically written.

A rung of logic that performs this configuration and the configuration dialog box of the message instruction are shown in these examples.

**EXAMPLE** Automatic Save Set ladder instruction and configuration dialog box

Retrieve the current state of the automatic write setting by using a CIP Generic message type and a service type of Controller Log Automatic Write Get. Use this rung of ladder logic and message instruction configuration to get this value.

**EXAMPLE** Automatic Save Get ladder instruction and configuration dialog box
Save on demand

Command the controller to write buffered entries to the SD card by using a message instruction with a CIP Generic message type and a service type of Controller Log Write To Media.

A rung of ladder logic that sends this message and the configuration of the message instruction are shown in these examples.

EXAMPLE  Write Buffered Entries to the SD Card ladder instruction and configuration dialog box
Controller logging counters

Three counters, listed in the following table, provide real-time statistics about modifications to the controller.

<table>
<thead>
<tr>
<th>Counter Name</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Entry Count</td>
<td>Number of entries added to the log since the last firmware update.</td>
<td>GSV/SSV</td>
</tr>
<tr>
<td>Unsaved Entry Count</td>
<td>Number of entries in controller RAM not yet written to the CompactFlash card.</td>
<td>GSV</td>
</tr>
<tr>
<td>Modify Execution Count</td>
<td>Count that specifically tracks modifications that can change behavior of a running controller. A subset of entries increment this count.</td>
<td>GSV/SSV</td>
</tr>
</tbody>
</table>

Total entry count

Total Entry Count is the number of controller entries that have been added since the last firmware update. This counter increments after any entry is added to the log, and it is written to the log in the Record Number field. Use a Set System Value (SSV) instruction to set it to a known value. This can be useful, for example, for monitoring system changes during a production run.

This rung of ladder logic shows how to retrieve the Total Entry Count by using a Get System Value (GSV) instruction.

EXAMPLE  Total Entry Count by using a Get System Value (GSV) ladder instruction
This rung of ladder logic shows how to set the Total Entry Count to a known value (in this example, 0) by using an SSV instruction.

**EXAMPLE** Set the Total Entry Count to a Known Value ladder instruction

The Unsaved Entry Count is the number of log entries that are in controller memory but have not yet been stored to the CompactFlash card.

This counter value is available via a GSV instruction, and can range from 0...100, the maximum number of entries that the controller can buffer.

This rung of ladder logic shows how to retrieve the Unsaved Entry Count by using a Get System Value (GSV) instruction.

**EXAMPLE** Retrieve the Unsaved Entry Count using a Get System Value (GSV) ladder instruction

The Execution Modification Count tracks the number of changes that occur that can change the behavior of a running controller. Configure the counter to include or exclude force changes.

These events cause the Execution Modification Count to increment:

- Online edits tested or assembled
- Forces enabled or disabled (if so configured)
- Program properties modified
- Task properties modified
- Controller timeslice modified

Set this counter to a known value using an SSV instruction.

This example rung of ladder logic shows how to retrieve the Execution Modification Count by using a GSV instruction.

**EXAMPLE** Retrieve the Execution Modification Count by using a GSV ladder instruction

```
Controller Object Execution Modification Count
Number of log entries generated from the following modifications:
  Program properties change
  Task properties change
  Online edit
  Controller timeslice changes
  Optionally force changes
  *This is configured by the execution count configure mask

GSV
Get System Value
Class Name
Instance Name
Attribute Name
Dest
ControllerLogExecutionModificationCount
ExecModificationCount
12
```

This rung of ladder logic shows how to set the Execution Modification Count to a known value.

**EXAMPLE** Ladder instruction to set the Execution Modification Count to a known value

```
Set the Execution Modification Count
SSV
Set System Value
Class Name
Instance Name
Attribute Name
Source
ControllerLogExecutionModificationCount
ExecModificationCount
SetCount
```

To configure whether the Execution Modification Count includes forces, use a message instruction of message type CIP Generic and a service type of Controller Log Config Execution Set.

If it is sent a value of 1, forces are included in the counter. If it is sent a value of 0, forces are not included.

This rung of ladder logic shows how to send the message instruction, followed by the configuration dialog box of the message instruction.

**EXAMPLE** Controller Log Config Execution Set ladder instruction and configuration dialog box
The Source Element should be of data type DINT.

Use a message instruction to retrieve the current value of this configuration. This message uses a message type of CIP Generic and a service type of Controller Log Config Execution Get.

This rung of ladder logic shows how to send the message instruction, followed by the configuration dialog box of the message instruction.

**EXAMPLE** Controller Log Config Execution Get ladder instruction and configuration dialog box
Log file storage

The Destination tag should be of type DINT.

When a log file is written to the CompactFlash card, it is stored at `\Logix\XXXXXXXX\Logs\VYY_ZZ`, where XXXXXXXX is the eight-digit serial number of the controller and YY_ZZ is the revision number of the firmware (major_minor revision).

Log file location
The file is named ControllerLog_yyy.txt, where yyy is a sequential number from 000…999. The controller adds to the log file until it reaches a size greater than 1 MB. At that point, the next write of the controller log causes a new file to be created with the next sequence number.

When there are 1000 files larger than 1 MB, no more logs are created. However, the controller searches for the file name with the smallest possible sequence number that it can create or write to. For example, if a user deletes files 001…100 but leaves the rest, the controller starts creating logs again starting at a sequence number of 001. If there are already 1000 log files and a user deletes log entries out of file 005, the controller writes the next log entries to that file. The controller starts at 000 and looks for the first file that does not exist or is less than 1 MB in size.

Each time the controller opens a log file for writing, it creates a back-up file that is a copy of the log file before the write. This file is called Backup.txt. The controller overwrites the backup every time it opens a log file is opened for writing.

Rockwell Automation recommends periodically clearing space on the card for new log files. The controller does not delete any files off the card to create more space for new log files.

This table lists the information that is contained in the controller log file.

<table>
<thead>
<tr>
<th>Content</th>
<th>Description</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Controller's GMT time</td>
<td>MMM-DD-YY HH:MM:SS 24-hour time</td>
</tr>
<tr>
<td>Entry</td>
<td>Entry Description (defined in Entry List section)</td>
<td></td>
</tr>
<tr>
<td>User Name</td>
<td>User’s login ID</td>
<td>Windows domain name with display name if available</td>
</tr>
<tr>
<td>Workstation Name</td>
<td>User’s computer name</td>
<td>Computer Name</td>
</tr>
<tr>
<td>FactoryTalk ID</td>
<td>User’s FactoryTalk login ID</td>
<td>Alphanumeric characters</td>
</tr>
<tr>
<td>Extended Information</td>
<td>Entry specific information (defined in Entry List section)</td>
<td></td>
</tr>
<tr>
<td>Change Detection Audit Value(^1)</td>
<td>Changes that the controller detected in the Audit value</td>
<td></td>
</tr>
</tbody>
</table>

(1) Version 20 or later. See Change detection.

The log file is formatted in UTF-16, and has a file extension of .txt. Double-clicking the file opens it in Notepad on most systems. However, since it is formatted as Tab Separated Values (TSV), it also opens correctly in a spreadsheet application, like Microsoft Excel.

This screen is an example of the log file in Notepad.

Tip: For version 20 and later, a column for the Audit Value changes is included in the log file.
Create Custom Log Entries

Use a message instruction to add custom entries to the controller log. The message instruction uses a CIP Generic message type and a service type of Controller Log Add Entry.

The source element of this message should be a tag of a user-defined data type. The user-defined data type should contain two string members. The controller writes the first string to the log entry's Description field. The controller writes the second string to the log entry's Extended Information field.

This rung of ladder logic shows how to send the message instruction, followed by the Configuration dialog box of the message instruction and the definition of the user-defined data type used for the source element.

EXAMPLE  Send the Message ladder instruction, Controller Log Add Entry configuration dialog box, and Data Type dialog box.

See also

Change detection on page 48
The Logix Designer application contains a controller logging sample ladder file. If the sample files were installed during the installation, open the sample files from the Help menu. On the Help menu, select Vendor Sample Projects.

This table describes the events that the controller stores in the controller log.

Tip: In Logix Designer versions 30 and later, audit log entries also appear in the controller log. Audit log entries describe events such as changes to component properties, edits in a routine, and changes to tag values. The audit log is used for auditing by the FactoryTalk® AssetCentre.
<table>
<thead>
<tr>
<th>Entry</th>
<th>Information Logged</th>
</tr>
</thead>
</table>
| Project download | • Time Stamp  
• Entry Description: Project download  
• UserName  
• Workstation Name  
• FactoryTalk Login Id  
• Extended Information: Project |
| Load from removable media | • Time Stamp  
• Entry Description: Project load  
• UserName  
• Workstation Name  
• FactoryTalk Login Id  
• Extended Information: Project |
| Load from removable media auto-initiated | • Time Stamp  
• Entry Description: Project auto load  
• UserName: Local  
• Workstation Name: None  
• FactoryTalk Login Id: None  
• Extended Information: Project |
| Store to removable media | • Time Stamp  
• Entry Description: Project store  
• UserName  
• Workstation Name  
• FactoryTalk Login Id  
• Extended Information: Project |
| Online edits tested or assembled | • Time Stamp  
• Entry Description: Online edits modified controller program  
• UserName  
• Workstation Name  
• FactoryTalk Login Id  
• Extended Information: None  
• Edits logged are:  
  • Test Program Edits  
  • UnTest Program Edits  
  • Assemble Program Edits  
  • Accept Program Edits  
  • Accept Pending Rung Edits |
| Partial import online completed | • Time Stamp  
• Entry Description: Partial import online modified controller  
• UserName  
• Workstation Name  
• FactoryTalk Login Id  
• Extended Information: None |
| I/O forces enabled | • Time Stamp  
• Entry Description: I/O forces enabled  
• UserName  
• Workstation Name  
• FactoryTalk Login Id  
• Extended Information: None |
### I/O forces disabled
- **Time Stamp**
- **Entry Description:** I/O forces disabled
- **UserName**
- **Workstation Name**
- **FactoryTalk Login Id**
- **Extended Information:** None

### I/O forces removed
- **Time Stamp**
- **Entry Description:** I/O forces removed
- **UserName**
- **Workstation Name**
- **FactoryTalk Login Id**
- **Extended Information:** None

### I/O forces modified
- **Time Stamp**
- **Entry Description:** I/O force value changed
- **UserName**
- **Workstation Name**
- **FactoryTalk Login Id**
- **Extended Information:** Tag

### SFC forces enabled
- **Time Stamp**
- **Entry Description:** SFC forces enabled
- **UserName**
- **Workstation Name**
- **FactoryTalk Login Id**
- **Extended Information:** None

### SFC forces disabled
- **Time Stamp**
- **Entry Description:** SFC forces disabled
- **UserName**
- **Workstation Name**
- **FactoryTalk Login Id**
- **Extended Information:** None

### SFC forces removed
- **Time Stamp**
- **Entry Description:** SFC forces removed
- **UserName**
- **Workstation Name**
- **FactoryTalk Login Id**
- **Extended Information:** None

### SFC forces modified
- **Time Stamp**
- **Entry Description:** SFC element force value changed
- **UserName**
- **Workstation Name**
- **FactoryTalk Login Id**
- **Extended Information:** Routine

### Firmware update from workstation
- **Time Stamp**
- **Entry Description:** Firmware update attempted
- **UserName:** None
- **Workstation:** None
- **FactoryTalk Login Id:** None
- **Extended Information:** Old revision `<major.<minor>`, New revision `<major.<minor>`, where the major and minor revision numbers are each two digits.
<table>
<thead>
<tr>
<th>Event Description</th>
<th>Time Stamp</th>
<th>UserName</th>
<th>Workstation Name</th>
<th>FactoryTalk Login Id</th>
<th>Extended Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firmware update from removable media</td>
<td>• Time Stamp</td>
<td>• Entry Description: Firmware update from removable media attempted</td>
<td>• UserName: Local</td>
<td>• Workstation: None</td>
<td>• Extended Information: Old revision &lt;major.&lt;minor&gt;, New revision &lt;major.&lt;minor&gt;, where the major and minor revision numbers are each two digits.</td>
</tr>
<tr>
<td>Mode changed through Logix Designer</td>
<td>• Mode change started Time Stamp</td>
<td>• Entry Description: Remote mode change</td>
<td>• UserName</td>
<td>• Workstation Name</td>
<td>• FactoryTalk Login Id: None</td>
</tr>
<tr>
<td>Mode changed through keyswitch</td>
<td>• Time Stamp</td>
<td>• Entry Description: Keyswitch mode change</td>
<td>• UserName: Local</td>
<td>• Workstation Name: None</td>
<td>• FactoryTalk Login Id: None</td>
</tr>
<tr>
<td>Major fault</td>
<td>• Time Stamp</td>
<td>• Entry Description: A major fault occurred</td>
<td>• UserName: None</td>
<td>• Workstation Name: None</td>
<td>• FactoryTalk Login Id: None</td>
</tr>
<tr>
<td>Major faults cleared</td>
<td>• Time Stamp</td>
<td>• Entry Description: All major faults cleared</td>
<td>• UserName</td>
<td>• Workstation Name</td>
<td>• FactoryTalk Login Id</td>
</tr>
<tr>
<td>Event Description</td>
<td>Time Stamp</td>
<td>Entry Description</td>
<td>UserName</td>
<td>Workstation Name</td>
<td>FactoryTalk Login Id</td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td>------------</td>
<td>--------------------------------------------------------</td>
<td>----------</td>
<td>------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Major faults cleared through key switch</td>
<td></td>
<td>All major faults cleared</td>
<td>Local</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Program properties modified</td>
<td></td>
<td>Program properties modified</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task properties modified</td>
<td></td>
<td>Task properties modified</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Controller time slice modified</td>
<td></td>
<td>Controller timeslice modified</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Removable media removed</td>
<td></td>
<td>Removable media removed</td>
<td>Local</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Event Type</td>
<td>Information Provided</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Removable media inserted</td>
<td>- Time Stamp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Entry Description: Removable media inserted</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- UserName: Local</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Workstation Name: None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- FactoryTalk Login Id: None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Extended Information: None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Safety signature create</td>
<td>- Time Stamp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Entry Description: Safety signature create</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- UserName</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Workstation Name</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- FactoryTalk Login Id</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Extended Information:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Eight-character signature number on Compact GuardLogix 5370 and GuardLogix 5570 controllers: 0xYYYYYYYY (hex format)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- 64-character signature number on Compact GuardLogix 5380 and GuardLogix 5580 controllers: 0xYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY (hex format)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Safety signature delete</td>
<td>- Time Stamp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Entry Description: Safety signature delete</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- UserName</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Workstation Name</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- FactoryTalk Login Id</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Extended Information:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Eight-character signature number on Compact GuardLogix 5370 and GuardLogix 5570 controllers: 0xYYYYYYYY (hex format)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- 64-character signature number on Compact GuardLogix 5380 and GuardLogix 5580 controllers: 0xYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY (hex format)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Safety lock</td>
<td>- Time Stamp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Entry Description: Safety lock</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- UserName</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Workstation Name</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- FactoryTalk Login Id</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Extended Information: None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Safety unlocked</td>
<td>- Time Stamp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Entry Description: Safety unlock</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- UserName</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Workstation Name</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- FactoryTalk Login Id</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Extended Information: None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Custom entry</td>
<td>- Time Stamp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Entry Description: &lt;User supplied string&gt; maximum 40 characters</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- UserName</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Workstation Name</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- FactoryTalk Login Id</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Extended Information: &lt;User supplied info&gt;, maximum 82 characters</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Description</td>
<td>Time Stamp</td>
<td>Entry Description</td>
<td>UserName</td>
<td>Workstation Name</td>
<td>FactoryTalk Login Id</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------------------------</td>
<td>----------</td>
<td>-----------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Constant tag data changed</td>
<td>Time Stamp</td>
<td>Entry Description: Constant tag data changed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>UserName</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Workstation Name</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FactoryTalk Login Id</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended Information: Tag: <code>&lt;tag name&gt;</code>&lt;old value&gt; to <code>&lt;new value&gt;</code></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiple constant tag data changed</td>
<td>Time Stamp</td>
<td>Entry Description: Multiple constant tag data changed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>UserName</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Workstation Name</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FactoryTalk Login Id</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended Information: Tag: <code>&lt;tag name&gt;</code></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change to constant tag configuration reset</td>
<td>Time Stamp</td>
<td>Entry Description: Constant tag configuration reset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>UserName</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Workstation Name</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FactoryTalk Login Id</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended Information: Tag: <code>&lt;tag name&gt;</code></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Safety signature delete inhibited in Run mode</td>
<td>Time Stamp</td>
<td>Entry Description: Safety signature delete inhibited in Run mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>UserName</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Workstation Name</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FactoryTalk Login Id</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended Information: None</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Safety signature delete allowed in Run mode</td>
<td>Time Stamp</td>
<td>Entry Description: Safety signature delete allowed in Run mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>UserName</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Workstation Name</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FactoryTalk Login Id</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended Information: None</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Audit Value Mask Modified</td>
<td>Time Stamp</td>
<td>Entry Description: Change detection mask modified</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>UserName</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Workstation Name</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FactoryTalk Login Id</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended Information: Old mask: <code>0xFFFF_FFFF_FFFF_FFFF</code> New mask: <code>0xFFFF_FFFF_FFFF_DFFF</code></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Log Collected Data Cleared</td>
<td>Time Stamp</td>
<td>Entry Description: Log Collected Data Cleared</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>UserName</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Workstation Name</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FactoryTalk Login Id</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended Information: Log: <code>&lt;log name&gt;</code></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Program Connection Modified
- Time Stamp
- Entry Description: Program connection modified
- UserName
- Workstation Name
- FactoryTalk Login Id
- Extended Information: Program, <connection type>

### USB Cable Connected\(^3\)
- Time Stamp
- Entry Description: USB connected
- UserName: None
- Workstation Name: None
- FactoryTalk Login Id: None
- Extended Information: None

### USB Cable Disconnected\(^3\)
- Time Stamp
- Entry Description: USB disconnected
- UserName: None
- Workstation Name: None
- FactoryTalk Login Id: None
- Extended Information: None

---

1. In versions 20 and later, the Change Detection Audit Value column is included in the controller log. This column is used to record the Audit Value for Change Detection.

2. In versions 20 and later, this value in the controller log equates to Transaction committed in the Change Detection mask.

3. This feature only applies for the Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, and GuardLogix 5580 controllers.
Chapter 4

Change detection

Introduction

Use the controller change detection feature as an additional means of detecting changes made to Logix 5000 controllers.

- A unique audit value is generated when a project is downloaded to the controller.
- When a change is detected in the controller, a new audit value is generated.
- ChangesToDetect mask allows programmatic configuration of the events to monitor for changes.
- Controller change detection is integrated into the Logix Designer application.

**IMPORTANT** The change detection feature is not supported in version 19 or earlier.

Keep these considerations in mind when using change detection:

- Change detection is not available on the Studio 5000® Logix Emulate™ Controller and the SoftLogix5860 Controller, and the audit value is not populated.
- The 1769-L3x and 1769-L4x CompactLogix controllers do not support storing log entries to removable media.
- Change detection is integrated in FactoryTalk AssetCentre version 4.1 and later. FactoryTalk AssetCentre can be configured to detect changes in the controller and read the controller’s Controller Log.
- Change detection is not integrated with RSMACC utilities.

Controller change detection

Two controller attributes support the Change Detection feature in version 20 and later.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuditValue</td>
<td>A unique value that is generated when a project is downloaded to the controller or loaded from removable storage. When a change is detected this value is updated. To specify which changes are monitored, use the ChangesToDetect attribute.</td>
<td>GSV</td>
</tr>
<tr>
<td>ChangesToDetect</td>
<td>Used to specify which changes are monitored. When a monitored change occurs, the Audit Value is updated.</td>
<td>GSV/SSV</td>
</tr>
</tbody>
</table>

ChangesToDetect

The ChangesToDetect mask is a 64-bit value. Each bit of the ChangesToDetect mask corresponds to a particular event that could cause the Audit Value to change. See ChangeToDetect format.
IMPORTANT: Change detection is unavailable on the Studio 5000 Logix Emulate Controller and the SoftLogix5860 Controller.

If a bit is set to a value of 1, the corresponding event is monitored for changes and when a change for that event occurs the AuditValue updates. If a bit is set to a value of 2, the corresponding event is not monitored and does not affect the Audit Value. The default Audit Value of: 0xFFFFFFFFFFFFFFFF indicates that all events are monitored.

Tip: The Audit Value updates when the controller is online.

Some events always update the audit value when they occur. For example, the audit value changes when a project is downloaded to the controller, or when ChangesToDetect is reconfigured. These types of events are not included in the ChangesToDetect mask.

Configure the ChangesToDetect mask programmatically using SSV, GSV and MSG instructions.

Tip: Use the Security tab in the Controller Properties dialog box to configure Change Detection on version 20 or later of the Logix Designer application.

Use the Set System Value (SSV) instruction to write to the ChangesTo Detect attribute and the Get System Value (GSV) instruction to read the ChangesTo Detect attribute programmatically.

Tip: Rockwell recommends using the DINT[2] data type to avoid limitations when working with LINT data types in Rockwell Automation controllers.

Use the Message Configuration dialog box to read or write to the ChangesToDetect attribute through CIP Generic Messages. For complete information on how to access data using an MSG instruction, refer to the <LOGIX> Controllers Message Programming Manual, publication 1756-PM012.

Use these settings to configure the ChangesToDetect attribute using the Message Configuration dialog box.

<table>
<thead>
<tr>
<th>If you want to:</th>
<th>In this property</th>
<th>Type or select</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set controller events monitored for changes</td>
<td>Message Type</td>
<td>CIP Generic</td>
</tr>
<tr>
<td></td>
<td>Service Type</td>
<td>Changes to Detect Set</td>
</tr>
<tr>
<td></td>
<td>Source</td>
<td>tag_name of type DINT[2] or LINT 1 This tag represents a bit mask of the changes monitored for the controller.</td>
</tr>
<tr>
<td></td>
<td>Destination</td>
<td>Leave blank</td>
</tr>
</tbody>
</table>
If you want to: | In this property | Type or select
---|---|---
Get controller events monitored for changes | Message Type | CIP Generic
| Service Type | Changes to Detect Get
| Source | Leave blank
| Destination | tag_name of type DINT[2] or LINT 1
This tag represents a bit mask of the changes monitored for the controller.

See also

*ChangesToDetect Format* on page 47

*Logix 5000 Controllers Messages Programming Manual*, publication no. 1756-PM012

**AuditValue**

Use the Get System Value (GSV) instruction to read the AuditValue attribute programmatically.

![GSV Get System Value](image)

Use the **Message Configuration** dialog box to read the AuditValue attribute through CIP Generic Messages. For complete information on how to access data using an MSG instruction, refer to the *Logix 5000 Controllers Messages Programming Manual*, publication no. 1756-PM0012.

Use these settings to read the Audit Value attribute using the Message Configuration.

| To get AuditValue in this property | Type or select |
---|---|
Message Type | CIP Generic
Service Type | Audit Value Get
Source | Leave blank
Destination | tag_name of type DINT[2] or LINT 1
This tag contains the AuditValue for the controller.

See also

*Logix 5000 Controllers Messages Programming Manual, publication no. 1756-PM012*, publication no. 1756-PM012

**ChangesToDetect format**

| Bit number | Event description |
---|---|
0 | Project stored to removable media |
1 | Online edits modified controller program |
<table>
<thead>
<tr>
<th>Bit number</th>
<th>Event description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Transaction committed. Indicates that a batch of one or more changes has been applied to the controller. For example, this might happen when a routine is imported online or when a module configuration is changed online.</td>
</tr>
<tr>
<td>3</td>
<td>SFC forces enabled</td>
</tr>
<tr>
<td>4</td>
<td>SFC forces disabled</td>
</tr>
<tr>
<td>5</td>
<td>SFC forces removed</td>
</tr>
<tr>
<td>6</td>
<td>SFC element force value changed</td>
</tr>
<tr>
<td>7</td>
<td>I/O forces enabled</td>
</tr>
<tr>
<td>8</td>
<td>I/O forces disabled</td>
</tr>
<tr>
<td>9</td>
<td>I/O forces removed</td>
</tr>
<tr>
<td>10</td>
<td>I/O forces modified</td>
</tr>
<tr>
<td>11</td>
<td>Firmware update attempted</td>
</tr>
<tr>
<td>12</td>
<td>Firmware update from removable media attempted</td>
</tr>
<tr>
<td>13</td>
<td>Remote mode change</td>
</tr>
<tr>
<td>14</td>
<td>Keyswitch mode change</td>
</tr>
<tr>
<td>15</td>
<td>A major fault occurred</td>
</tr>
<tr>
<td>16</td>
<td>All major faults cleared</td>
</tr>
<tr>
<td>17</td>
<td>All major faults cleared through keyswitch</td>
</tr>
<tr>
<td>18</td>
<td>Task properties modified</td>
</tr>
<tr>
<td>19</td>
<td>Program properties modified</td>
</tr>
<tr>
<td>20</td>
<td>Controller time slice modified</td>
</tr>
<tr>
<td>21</td>
<td>Removable media removed</td>
</tr>
<tr>
<td>22</td>
<td>Removable media inserted</td>
</tr>
<tr>
<td>23</td>
<td>Safety signature created</td>
</tr>
<tr>
<td>24</td>
<td>Safety signature deleted</td>
</tr>
<tr>
<td>25</td>
<td>Safety lock</td>
</tr>
<tr>
<td>26</td>
<td>Safety unlock</td>
</tr>
<tr>
<td>27</td>
<td>Constant Tag value changed</td>
</tr>
<tr>
<td>28</td>
<td>Multiple constant Tag values changed</td>
</tr>
<tr>
<td>29</td>
<td>Constant Tag attribute cleared</td>
</tr>
<tr>
<td>30</td>
<td>Constant Tag attribute set</td>
</tr>
<tr>
<td>31</td>
<td>Custom Log Entry Added</td>
</tr>
<tr>
<td>32</td>
<td>Correlation Affected. Indicates a change occurred in the controller that affects the synchronization between the controller and the project file that was downloaded to it. <strong>Tip:</strong> Correlation Affected can include detection of a component that has been created, deleted, or modified, or logic that has been modified. Including the Correlation Affected event updates the audit value for these types of changes, even if other similar events (such as Online edits modified controller program or Task properties modified) are masked.</td>
</tr>
<tr>
<td>33</td>
<td>Safety signature delete inhibited in Run mode</td>
</tr>
<tr>
<td>34</td>
<td>Safety signature delete allowed in Run mode</td>
</tr>
<tr>
<td>35</td>
<td>Data log or Alarm log cleared</td>
</tr>
<tr>
<td>36</td>
<td>Program connection modified</td>
</tr>
</tbody>
</table>
Use the **Security** tab in the **Controller Properties** dialog box (available in versions 20 and later of the application) to configure Change Detection.

To edit the **Changes To Detect** field, type a new value. To select the events to monitor for changes from a list, select **Configure** to open the **Configure Changes to Detect** dialog box.

For additional information on configuring the settings on the **Security** tab, refer to the `<LOGIX>` Security Programming Manual, publication 1756-PM016, or the online help for the Logix Designer application.

**Tip:** To configure Change Detection programmatically use SSV, GSV, or MSG instructions. See Controller change detection.

### Component tracking

Use component tracking to determine whether tracked components have changed. Like change detection, component tracking indicates changes to a project, but at the component level. The Logix Designer application creates a unique signature for each tracked component and compiles those signatures into one overall tracked value to indicate their current state. Component tracking applies to:
• Routines
• Add-On Instructions
• I/O Modules
• Constant tags

Tip: Component tracking is supported only on CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, and GuardLogix 5570 controllers in versions 30 and 31 of the Logix Designer application.

Tracked components appear in the **Tracked Components** dialog box, which is accessible on the **Controller Properties** dialog box - **Security** tab. The tracked value changes when a tracked component updates in a way that affects the execution of a routine or an Add-On Instruction. Updating a component attribute that does not affect execution—for example, descriptions and non-constant tag values—does not change the tracked value.

Tip: When an attribute is changed and then changed back to its previous value, the tracked value does not change. This behavior differs from Change Detection, which registers a change when the change is reversed.

Tip: The **Tracked State** value and the list of tracked components are static as long as the dialog box is open and will not reflect changes from other workstations. If a user on another workstation makes an update, if changes are not showing, close and re-open the **Tracked Components** dialog box to see the changes.

### Recommended limits for tracked components

This table lists the recommended limits on the number of tracked components. If the limits are exceeded, a noticeable impact on performance may occur in the Logix Designer application.

Tip: To optimize performance, configure component tracking so that the tracked state value is calculated on demand rather than at regular intervals.

<table>
<thead>
<tr>
<th>Tracked component</th>
<th>Recommended limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routines</td>
<td>100</td>
</tr>
<tr>
<td>Add-On Instructions</td>
<td>100</td>
</tr>
<tr>
<td>I/O Modules</td>
<td>50</td>
</tr>
<tr>
<td>Tags</td>
<td>300</td>
</tr>
</tbody>
</table>

### FactoryTalk Security permissions that control component tracking

These FactoryTalk Security permissions control a user’s ability to change tracking status for a component:

• Routine: Modify Properties
• Add-On Instruction: Modify
• Module: Modify Properties
• Tag: Modify Properties
Designate project components for tracking to monitor the current state of a component. Use the following instructions to track the state of a routine, an Add-On Instruction, an I/O module, or a constant tag.

**To track the state of a component:**

1. In the **Controller Organizer**, highlight the component to track.
2. Right-click and select **Include in tracking group**.
3. To stop tracking a component, right-click and select **Include in tracking group** again.

   **Tip:** When using the tracked state value to verify that code in a tracked routine is being executed, make sure to track all the routines that call the tracked routine, up to and including the main routine. Tracking all components that call the tracked routine helps to prevent a user from modifying the project so that the tracked routine is no longer called.

**To track the state of constant tags**

1. In the **Tag editor** or the **Data Monitor**, highlight the constant tag to track.
2. Right-click and select **Include in tracking group**.
3. To stop tracking a constant tag, right-click and select **Include in tracking group** again.

   **Tip:** When a base tag or an alias tag is tracked, the base tag and all alias tags are tracked.

**View tracked components**

After selecting project components for tracking, check project component status in the **Tracked Components** dialog box, which is accessible on the **Controller Properties dialog box - Security** tab.

**To view tracked components:**

1. On the **Online** toolbar, click the **Controller Properties** icon.
2. On the **Controller Properties** dialog box, click the **Security** tab.
3. In the **Change Detection** section, click **View components** to open the **Tracked components** dialog box.

**Retrieve the tracked state value using a Message instruction**

Use a Message (MSG) instruction to retrieve the tracked state value. When the controller receives the MSG instruction, it calculates the tracked state value and stores it in a destination tag.

   **Tip:** When using the tracked state value to verify that code in a tracked routine is being executed, make sure to track all the routines that call the tracked routine, up to and including the main routine. Tracking all components that call the tracked routine helps to prevent a user from modifying the project so that the tracked routine is no longer called.
To retrieve the tracked state value using an MSG instruction:

1. Create a MESSAGE tag.
2. Insert an MSG instruction in the ladder logic or Structured Text routine.
3. In the instruction, click Browse to open the Message Configuration dialog box.
4. On the Configuration tab, configure the following settings:
   - For Message Type, select CIP Generic.
   - For Service Type, select Tracked State Value Get.
   - For Source Element, select the source tag element. In this release of Logix Designer, the value of the SINT datatype must be 1.
   - For Destination Element, select the tag where the controller sends the tracked state value. For best results, create a destination tag that is an array of 32 SINTs.
5. On the Communication tab, configure the communication path.
6. Click Apply.
Access status information

Introduction

The controller supports status keywords inserted in logic for monitoring specific events.

- The status keywords are not case sensitive.
- Because the status flags can change quickly, the Logix Designer application does not display the status of the flags. For example, even when a status flag is set, an instruction that references that flag is not highlighted.
- A tag alias cannot be defined to a keyword.

The following table lists the available keywords.

<table>
<thead>
<tr>
<th>To determine if</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>The stored value cannot fit into the destination because it is either:</td>
<td>S:V</td>
</tr>
<tr>
<td>- Greater than the maximum value for the destination.</td>
<td></td>
</tr>
<tr>
<td>- Less than the minimum value for the destination.</td>
<td></td>
</tr>
<tr>
<td><em>Important:</em> Each time S:V goes from cleared to set, it generates a minor fault (type 4, code 4).</td>
<td></td>
</tr>
<tr>
<td>The instruction’s destination value is 0.</td>
<td>S:Z</td>
</tr>
<tr>
<td>The instruction’s destination value is negative.</td>
<td>S:N</td>
</tr>
<tr>
<td>An arithmetic operation causes a carry or borrow that tries to use bits that are outside the data type.</td>
<td>S:C</td>
</tr>
<tr>
<td>For example:</td>
<td></td>
</tr>
<tr>
<td>- Adding 3 + 9 causes a carry of 1.</td>
<td></td>
</tr>
<tr>
<td>- Subtracting 25 - 18 causes a borrow of 10.</td>
<td></td>
</tr>
<tr>
<td>This is the first, normal scan of the routines in the current program.</td>
<td>S:FS</td>
</tr>
<tr>
<td>At least one minor fault has been generated.</td>
<td>S:MINOR</td>
</tr>
<tr>
<td>- The controller sets this bit when a minor fault occurs due to program execution.</td>
<td></td>
</tr>
<tr>
<td>- The controller does not set this bit for minor faults that are not related to program execution, such as battery low.</td>
<td></td>
</tr>
</tbody>
</table>

Status of S:FS when the project has an SFC

The state of S:FS depends on the status of the SFC.

- When using S:FS in an action of a sequential function chart (SFC), S:FS is set (on) for one scan each time the step goes active. S:FS = step_name.FS.
- If the SFC calls a routine, S:FS is set (on) for one scan each time the step that calls the routine goes active. S:FS = step_name.FS.
  
  If the SFC does not call a routine, S:FS is set (on) for the first scan of the task.
EXAMPLE SFC Calls a Ladder Diagram
Suppose several steps in an SFC call the same ladder diagram routine. And
suppose the ladder diagram uses S:FS. Each time one of those steps goes active,
S:FS turns on for one scan of the ladder diagram.

EXAMPLE Several Tasks but No SFC
Suppose two tasks use ladder diagrams. When the first task runs for the first time,
S:FS turns on for one scan. After that, S:FS stays off for that task. When the other
task runs for the first time, S:FS turns on for one scan in that task. S:FS stays off in
the first task that ran.

Get and set system data

The controller stores system data in objects. There is no status file, as in the
PLC-5 controller. Use the GSV/SSV instructions to get and set controller
system data that is stored in objects.

- The GSV instruction retrieves the specified information and places it
  in the destination.
- The SSV instruction sets the specified attribute with data from the
  source.

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

To create a GSV/SSV instruction, use Logix Designer online help to find the
required information, and create the instruction in the Logix Designer
application. The following summary describes the steps.

- Open the online help in Logix Designer application to find the object
  and data type to use in the GSV/SSV instruction.
- In the Logix Designer application, create a tag for the value of the
  attribute.
- In the Ladder Logic editor, create the GSV/SSV instruction.

To get or set system data

1. Open the project in the Logix Designer application.
2. From the Help menu, choose Contents.
3. On the Help window, click the Index tab.
4. In the Type in the keyword to find field, type GSV/SSV objects and
   click Display.
5. In the GSV/SSV Objects help window, click the object.

<table>
<thead>
<tr>
<th>To get or set</th>
<th>Select this object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customize instructions for commonly-used logic</td>
<td>Add01InstructionDefinition</td>
</tr>
<tr>
<td>Different Axis objects depending on motion application</td>
<td>Axis (Several types)</td>
</tr>
<tr>
<td>System-overhead time slice</td>
<td>Controller</td>
</tr>
<tr>
<td>Physical hardware of a controller</td>
<td>ControllerDevice</td>
</tr>
</tbody>
</table>
6. In the list of attributes for the object, identify the attribute to access. For attribute information, see the Logix 5000 Controllers General Instruction Reference Manual, publication no. 1756-PM003.

7. In Logix Designer application, create a tag for the value of the attribute.

<table>
<thead>
<tr>
<th>If the data type of the attribute is</th>
<th>Then</th>
</tr>
</thead>
<tbody>
<tr>
<td>One element (for example, DINT)</td>
<td>Create a tag for the attribute.</td>
</tr>
</tbody>
</table>
| More than one element (for example, DINT[7]) | • Create a user-defined data type that matches the organization of data that is used by the attribute.  
• Create a tag for the attribute and use the data type identified for the object. |

8. In the Ladder Logic routine, enter the appropriate instruction.

<table>
<thead>
<tr>
<th>To</th>
<th>Enter this instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get the value of an attribute</td>
<td>GSV</td>
</tr>
<tr>
<td>Set the value of an attribute</td>
<td>SSV</td>
</tr>
</tbody>
</table>

9. Assign the required operands to the instruction.

<table>
<thead>
<tr>
<th>For this operand</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class name</td>
<td>Name of the object.</td>
</tr>
</tbody>
</table>
| Instance name   | Name of the specific object (for example, name of the required I/O module, task, message).  
• Not all objects require this entry.  
• To specify the current task, program, or routine, select THIS. |
| Attribute name  | Name of the attribute. |
| Dest (GSV)      | Tag that stores the retrieved value.  
If the tag is a user-defined data type or an array, select the first member or element. |
<table>
<thead>
<tr>
<th>For this operand</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source (SSV)</td>
<td>Tag that stores the value to be set. If the tag is a user-defined data type or an array, select the first member or element.</td>
</tr>
</tbody>
</table>

The following example gets the current date and time.

**EXAMPLE**

Get a System Value

At the first scan, the following rung of logic gets the DateTime attribute of the WallClockTime object and stores it in the wall_clock tag, which is based on a user-defined data type.

For more GSV/SSV information, see the *Logix 5000 Controllers General Instructions Reference Manual*, publication no. 1756-RM003.

**See also**

Logix 5000 Controllers General Instructions Reference Manual, publication no. 1756-RM003
Index

A
access
  status information 53
AuditValue 47

B
bridge
  module connection 9

C
Change Detection 45
ChangesToDetect 45
communicate
  other controllers 9
CompactFlash
  log 26
Component tracking 49
configure
  I/O module 9
connection
  failure 11
  fault checkbox 12
  I/O fault 11
  inhibit 9
  major fault 12
  monitor 13
controller
  log entries 23
  memory information 15

D
DINT 22

F
fault
  communication loss 11
  I/O connection 11
file
  log 34

G
GSV/SSV
  objects 54

I
I/O module
  communication loss 11
  connection fault 11
  inhibit 9
  path configuration 9
inhibit
  attention symbol 9
  connection 9
  I/O module 9
  ladder logic 9
INT 22

L
ladder
  sample log file 36
log
  CompactFlash card 26
  counters 29
  custom entries 35
  entries 25
  file format 34
  file location 33
  header 24
  memory buffer 26
  modification count 30
  sample ladder file 38

M
memory
  determine amount 15
  estimate free amount 17
  run message instruction 20
  types 15
  view controller usage 18
monitor
  I/O connection 13

O
objects
  data storage 54
Index

P

path
connection route 9

S

status
access data 53
memory 15
monitor 54

system data
access 54

T

Track components 49
Rockwell Automation support

Use these resources to access support information.

<table>
<thead>
<tr>
<th>Technical Support Center</th>
<th>Find help with how-to videos, FAQs, chat, user forums, and product notification updates.</th>
<th>rok.auto/support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledgebase</td>
<td>Access Knowledgebase articles.</td>
<td>rok.auto/knowledgebase</td>
</tr>
<tr>
<td>Local Technical Support Phone Numbers</td>
<td>Locate the telephone number for your country.</td>
<td>rok.auto/phonesupport</td>
</tr>
<tr>
<td>Literature Library</td>
<td>Find installation instructions, manuals, brochures, and technical data publications.</td>
<td>rok.auto/literature</td>
</tr>
<tr>
<td>Product Compatibility and Download Center (PCDC)</td>
<td>Get help determining how products interact, check features and capabilities, and find associated firmware.</td>
<td>rok.auto/pcdc</td>
</tr>
</tbody>
</table>

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.