Logix 5000 Controllers Ladder Diagram

1756 ControlLogix, 1756 GuardLogix, 1769 CompactLogix, 1769 Compact GuardLogix, 789 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).

---

Allen-Bradley, Rockwell Software, Rockwell Automation, and TechConnect are trademarks of Rockwell Automation, Inc.

Trademarks not belonging to Rockwell Automation are property of their respective companies.
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

This table identifies changes that apply to all information about a subject in the manual and the reason for the change. For example, the addition of new supported hardware, a software design change, or additional reference material would result in changes to all of the topics that deal with that subject.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updated Logix Designer application screen shots.</td>
<td>The Studio 5000 Logix Designer® interface has been modified in versions 31 and later.</td>
</tr>
<tr>
<td>Updated supported controller models.</td>
<td>Added 5069 CompactGuardLogix to the list of supported controllers.</td>
</tr>
</tbody>
</table>

New or enhanced features

None in this release.
Table of contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td></td>
</tr>
<tr>
<td>Studio 5000 environment</td>
<td>7</td>
</tr>
<tr>
<td>Additional resources</td>
<td>7</td>
</tr>
<tr>
<td>Legal notices</td>
<td>8</td>
</tr>
<tr>
<td>Chapter 1</td>
<td></td>
</tr>
<tr>
<td>Introduction</td>
<td>11</td>
</tr>
<tr>
<td>Instruction</td>
<td>11</td>
</tr>
<tr>
<td>Branch</td>
<td>11</td>
</tr>
<tr>
<td>Rung condition</td>
<td>13</td>
</tr>
<tr>
<td>Write ladder logic</td>
<td>13</td>
</tr>
<tr>
<td>Choose the required instructions</td>
<td>13</td>
</tr>
<tr>
<td>Arrange the input instructions</td>
<td>15</td>
</tr>
<tr>
<td>Arrange the output instructions</td>
<td>16</td>
</tr>
<tr>
<td>Choose a tag name for an operand</td>
<td>16</td>
</tr>
<tr>
<td>Enter ladder logic</td>
<td>17</td>
</tr>
<tr>
<td>Append an element to the cursor location</td>
<td>18</td>
</tr>
<tr>
<td>Drag and drop an element</td>
<td>18</td>
</tr>
<tr>
<td>Assign instruction operands</td>
<td>18</td>
</tr>
<tr>
<td>Create and assign a new tag</td>
<td>18</td>
</tr>
<tr>
<td>Choose a name or an existing tag</td>
<td>20</td>
</tr>
<tr>
<td>Drag and drop a tag from the Tags window</td>
<td>20</td>
</tr>
<tr>
<td>Assign an immediate (constant) value</td>
<td>20</td>
</tr>
<tr>
<td>Enter a rung comment</td>
<td>20</td>
</tr>
<tr>
<td>Language switching</td>
<td>21</td>
</tr>
<tr>
<td>Verify the routine</td>
<td>22</td>
</tr>
</tbody>
</table>

Index
Preface

This manual shows how to program Logix 5000 controllers with the relay ladder programming language. 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 that is based on the Logix 5000 operating system.

Studio 5000 environment

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.

Additional resources

These documents contain additional information concerning related Rockwell Automation products.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logix 5000 Controllers Program Parameters Programming Manual, publication 1756-PM021</td>
<td>Describes how to use program parameters when programming Logix 5000 controllers.</td>
</tr>
</tbody>
</table>
Resource Description

Logix 5000 Controllers General Instructions Reference Manual, publication 1756-RM003 Describes the available instructions for a Logix 5000 controller.

Logix 5000 Controllers Process and Drives Instructions Reference Manual, publication 1756-RM006 Describes how to program a Logix 5000 controller for process or drives applications.

Logix 5000 Controllers Motion Instruction Set Reference Manual, publication MOTION-RM002 Describes how to program a Logix 5000 controller for motion applications.


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

Copyright notice

Copyright © 2018 Rockwell Automation Technologies, Inc. All Rights Reserved. Printed in USA.

This document and any accompanying Rockwell Software products are copyrighted by Rockwell Automation Technologies, Inc. Any reproduction and/or distribution without prior written consent from Rockwell Automation Technologies, Inc. is strictly prohibited. Please refer to the license agreement for details.

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.

Other Licenses

The software included in this product contains copyrighted software that is licensed under one or more open source licenses. Copies of those licenses are included with the software. Corresponding Source code for open source packages included in this product can be located at their respective web site(s).

You may alternately obtain complete Corresponding Source code by contacting Rockwell Automation via our 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.
A full list of all open source software used in this product and their corresponding licenses can be found in the OPENSOURCE folder included with the Release Notes. The default installed location of these licenses is C:\Program Files (x86)\Common Files\Rockwell\Help\<Product>\ReleaseNotes\OPENSOURCE\index.htm.

Trademark Notices


Any Rockwell Automation logo, software or hardware product not mentioned herein is also a trademark, registered or otherwise, of Rockwell Automation, Inc.

Other Trademarks

CmFAS Assistant, CmDongle, CodeMeter, CodeMeter Control Center, and WIBU are trademarks of WIBU-SYSTEMS AG in the United States and/or other countries. Microsoft is a registered trademark of Microsoft Corporation in the United States and/or other countries. ControlNet is a trademark of ControlNet International. DeviceNet is a trademark of the Open DeviceNet Vendors Association (ODVA). Ethernet/IP is a trademark of ControlNet International under license by ODVA.

All other trademarks are the property of their respective holders and are hereby acknowledged.

Warranty

This product is warranted in accordance with the product license. The product’s performance may be affected by system configuration, the application being performed, operator control, maintenance, and other related factors. Rockwell Automation is not responsible for these intervening factors. The instructions in
this document do not cover all the details or variations in the equipment, procedure, or process described, nor do they provide directions for meeting every possible contingency during installation, operation, or maintenance. This product’s implementation may vary among users.

This document is current as of the time of release of the product; however, the accompanying software may have changed since the release. Rockwell Automation, Inc. reserves the right to change any information contained in this document or the software at any time without prior notice. It is your responsibility to obtain the most current information available from Rockwell when installing or using this product.

Environmental compliance


Contact Rockwell Automation

Customer Support Telephone — 1.440.646.3434

Online Support — http://www.rockwellautomation.com/support/
Program ladder diagram

Introduction
You organize ladder diagram as rungs on a ladder and put instructions on each rung. There are two basic types of instructions:

- **Input instruction**: An instruction that checks, compares, or examines specific conditions in your machine or process.
- **Output instruction**: An instruction that takes some action, such as turn on a device, turn off a device, copy data, or calculate a value.

Instruction

![Diagram of ladder diagram with input and output instructions]

Branch

A branch is two or more instructions in parallel.

![Diagram of ladder diagram with a branch]

There is no limit to the number of parallel branch levels that you can enter. This example shows a parallel branch with five levels. The main rung is the first branch level, followed by four additional branches.

You can nest branches to as many as 6 levels. This example shows a nested branch. The bottom output instruction is on a nested branch that is three levels deep.

Large rungs with complex, nested branches result in having to scroll through the ladder editor and may end up spanning multiple pages when you print the logic. To make it easier to maintain, divide the logic into multiple smaller rungs.
The controller evaluates ladder instructions based on the rung condition preceding the instruction (rung-condition-in).

Only create instructions that affect the rung-condition-in of subsequent instructions on the rung.

- If the rung-condition-in to an input instruction is true, the controller evaluates the instruction and sets the rung-condition-out to match the results of the evaluation.
  - If the instruction evaluates to true, the rung-condition-out is true.
  - If the instruction evaluates to false, the rung-condition-out is false.
- An output instruction does not change the rung-condition-out.
  - If the rung-condition-in to an output instruction is true, the rung-condition-out is set to true.
  - If the rung-condition-in to an output instruction is false, the rung-condition-out is set to false.

Writing ladder logic requires that you choose the input and output instructions, and choose the tag names for operands.

1. Identify the conditions to check and separate them from the action to take for the rung.

2. Choose the appropriate input instruction for each condition and the appropriate output instruction for each action.

For more information on specific instructions, see:

- Logix 5000 Controllers General Instructions Reference Manual, publication **1756-RM003**
- Logix 5000 Controllers Process and Drives Instructions Reference Manual, publication **1756-RM006**
- Logix 5000 Controllers Motion Instruction Set Reference Manual, publication **MOTION-RM002**
Tip: I/O module data updates asynchronously to the execution of logic. If you reference an input multiple times in your logic, the input could change state between separate references. If you need the input to have the same state for each reference, buffer the input value and reference that buffer tag.

You can also use Input and Output program parameters which automatically buffer the data during logix execution. See the Logix 5000 Controllers Program Parameters Programming Manual, publication 1756-PM021.

The examples in this chapter use two simple instructions to help you learn how to write ladder diagram logic. The rules that you learn for these instructions apply to all other instructions.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Name</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>⋮ ⋮</td>
<td>Examine If Closed</td>
<td>XIC</td>
<td>An input instruction that looks at one bit of data.</td>
</tr>
<tr>
<td></td>
<td>If the bit is</td>
<td></td>
<td>Then the instruction (rung-condition-out) is</td>
</tr>
<tr>
<td></td>
<td>On (1)</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Off (0)</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>⋮ ⋮</td>
<td>Output Energize</td>
<td>OTE</td>
<td>An output instruction that controls one bit of data.</td>
</tr>
<tr>
<td></td>
<td>If the instructions to the left (rung-condition-in) are</td>
<td></td>
<td>Then the instruction turns the bit</td>
</tr>
<tr>
<td></td>
<td>True</td>
<td>On (1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>False</td>
<td>Off (0)</td>
<td></td>
</tr>
</tbody>
</table>
## Arrange the input instructions

Determine how to arrange the input instructions on the rung, as shown below.

<table>
<thead>
<tr>
<th>To check multiple input conditions when:</th>
<th>Arrange the input instructions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• All conditions must be met in order to take action.</td>
<td>In series: condition_1 condition_2 condition_3…</td>
</tr>
<tr>
<td>For example, if condition_1 AND condition_2 AND condition_3…</td>
<td></td>
</tr>
<tr>
<td>• Any one of several conditions must be met in order to take action.</td>
<td>In parallel: condition_1 condition_2 condition_3…</td>
</tr>
<tr>
<td>For example, if condition_1 OR condition_2 OR condition_3…</td>
<td></td>
</tr>
<tr>
<td>• There is a combination of the above.</td>
<td>In combination: condition_1 condition_2 condition_3…</td>
</tr>
<tr>
<td>For example: If condition_1 AND condition_2… OR If condition_3 AND condition_2…</td>
<td></td>
</tr>
</tbody>
</table>

### Tip:

The controller executes all instructions on a rung regardless of their rung-condition-in. For optimal performance of a series of instructions, sequence the instructions from most likely to be false on the left to least likely to be false on the right.

```
Instruction most likely to be FALSE
------------------------
[ ] [ ] [ ] [ ]
```

```
Instruction least likely to be FALSE
------------------------
[ ] [ ] [ ] [ ]
```

When the controller finds a false instruction, it executes the remaining instructions in the series with their rung-condition-in set to false. Typically, an instruction executes faster when its rung-condition-in (rung) is false rather than true.
**Arrange the output instructions**  
Place at least one output instruction to the right of the input instructions. You can enter multiple output instructions on a rung of logic:

<table>
<thead>
<tr>
<th>Option</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place the output instructions in sequence on the rung (serial).</td>
<td>![Diagram of serial sequence]</td>
</tr>
<tr>
<td>Place the output instructions in branches (parallel).</td>
<td>![Diagram of parallel branches]</td>
</tr>
<tr>
<td>Place the output instructions between input instructions. The last instruction on the rung must be an output instruction.</td>
<td>![Diagram of mixed sequence and branches]</td>
</tr>
</tbody>
</table>

**Choose a tag name for an operand**

Tag names follow these formats:

<table>
<thead>
<tr>
<th>For a:</th>
<th>Specify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag</td>
<td>tag_name</td>
</tr>
<tr>
<td>Bit number of a larger data type</td>
<td>tag_name.bit_number</td>
</tr>
<tr>
<td>Member of a structure</td>
<td>tag_name.member_name</td>
</tr>
<tr>
<td>Element of a one dimension array</td>
<td>tag_name[x]</td>
</tr>
<tr>
<td>Element of a two dimension array</td>
<td>tag_name[xy]</td>
</tr>
<tr>
<td>Element of a three dimension array</td>
<td>tag_name[x,y,z]</td>
</tr>
<tr>
<td>Element of an array within a structure</td>
<td>tag_name.member_name[x]</td>
</tr>
<tr>
<td>Member of an element of an array</td>
<td>tag_name[x,y,z].member_name</td>
</tr>
</tbody>
</table>

where:
- $x$ is the location of the element in the first dimension.
- $y$ is the location of the element in the second dimension.
- $z$ is the location of the element in the third dimension.

For a structure within a structure, add **.member_name**.
### Choose a Tag Name for an Operand

<table>
<thead>
<tr>
<th>To Access</th>
<th>The tag name looks like this:</th>
</tr>
</thead>
<tbody>
<tr>
<td>machine_on tag</td>
<td>machine_on</td>
</tr>
<tr>
<td>bit number 1 of the one_shots tag</td>
<td>one_shots.1</td>
</tr>
<tr>
<td>DN member (bit) of the running_seconds timer</td>
<td>running_seconds.DN</td>
</tr>
<tr>
<td>mix member of the north_tank tag</td>
<td>north_tank.mix</td>
</tr>
<tr>
<td>element 2 in the recipe array and element 1,1 in the tanks array</td>
<td></td>
</tr>
<tr>
<td>element 2 in the preset array within the north_tank tag</td>
<td></td>
</tr>
<tr>
<td>part_advance member of element 1 in the drill array</td>
<td></td>
</tr>
</tbody>
</table>

### Example:

A new routine contains a rung that is ready for instructions.

Use the **Language Element** toolbar to add a ladder diagram element.
Append an element to the cursor location

**Important:** Use caution when copying and pasting components between different versions of the Logix Designer application. The application only supports pasting to the same version or newer version. Pasting to a prior version of the application is not supported. When pasting to a prior version, the paste action may succeed but the results may not be as intended.

1. Click to select the instruction, branch, or rung that is above or to the left of where you want to add an element.

2. On the Language Element toolbar, click the button for the element that you want to add.

Drag and drop an element

Drag the button for the element directly to the desired location. A green dot shows a valid placement location (drop point).

Assign instruction operands

After you add an instruction to a ladder rung, you assign tags to the instruction operands. You can create a new tag, use an existing tag, or assign a constant value.

Create and assign a new tag

1. Click the operand area of the instruction.

2. Type a name for the tag and press the Enter key.

3. Right-click the tag name and then click New "tag_name".
4. In the **New Parameter or Tag** dialog box, in the **Usage** box, choose the usage.

5. In the **New Parameter or Tag** dialog box, in the **Data Type** box, click the button.

6. In the **Select Data Type** dialog box, choose the data type for the tag.

If you want to define the tag as an array, in the **Array Dimensions** boxes, enter the number of elements in each dimension.

7. Click **OK**.

8. In the **New Parameter or Tag** dialog box, choose the scope for the tag.

9. Click **OK**.
**Choose a name or an existing tag**

1. Double-click the operand area, and then click \( \text{?} \). The Tag Browser window appears.

2. Select the name or tag:

<table>
<thead>
<tr>
<th>To select as:</th>
<th>Do this:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label, routine name,</td>
<td>Click the name.</td>
</tr>
<tr>
<td>or similar type of</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td></td>
</tr>
<tr>
<td>Tag</td>
<td>Double-click the tag name.</td>
</tr>
<tr>
<td>Bit number</td>
<td>A. Click the tag name.</td>
</tr>
<tr>
<td></td>
<td>B. To the right of the tag name, click ( \text{?} ).</td>
</tr>
<tr>
<td></td>
<td>C. Click the required bit.</td>
</tr>
</tbody>
</table>

3. Press the **Enter** key or click a different spot on the ladder diagram to close the Tag Browser.

**Drag and drop a tag from the Tags window**

1. Find the tag in the **Controller Tags** or the **Program Parameters and Local Tags** window.

2. Double-click the tag to select it.

3. Click and drag the tag to its location on the instruction. A green dot appears to show you where you can drop the tag.

**Assign an immediate (constant) value**

1. Click the operand area of the instruction.

2. Type the value and press the **Enter** key.

**Enter a rung comment**

When entering a rung of ladder logic, you can add comments that explain the purpose of your rung.

To enter a rung comment, perform this procedure.
1. Right-click the rung number of your ladder logic and then click **Edit Rung Comment**.

   ![Edit Rung Comment dialog box](image)

   The **Rung Comment** dialog box appears.

2. Type your rung comment, and then click the green check to save your changes or click the red X to discard your changes.

**Language switching**

With version 17 and later of the application, you have the option to display project documentation, such as tag descriptions and rung comments for any supported localized language. You can store project documentation for multiple languages in a single project file rather than in language-specific project files. You define all the localized languages that the project supports and set the current, default, and optional custom localized language. The application uses the default language if the current language’s content is blank for a particular component of the project. However, you can use a custom language to tailor documentation to a specific type of project file user.

Enter the localized descriptions in your project, either when programming in that language or by using the import/export utility to translate the documentation off-
line and then import it back into the project. Once you enable language switching, you can dynamically switch between languages.

Project documentation that supports multiple languages within a project includes:

- Component descriptions in tags, routines, programs, user-defined data types, and Add-On Instructions.
- Equipment phases.
- Trends.
- Controllers.
- Alarm Messages (in ALARM_ANALOG and ALARM_DIGITAL configuration).
- Tasks.
- Property descriptions for modules in the Controller Organizer.
- Rung comments, SFC text boxes, and FBD text boxes.

For more information on enabling a project to support multiple translations of project documentation, see the online help.

**Verify the routine**

As you program your routine (s), it is a good idea to periodically verify your work.

1. In the Standard toolbar click the Verify icon.

2. Errors are listed in the Output window on the Errors tab at the bottom of the application.
   a. To go to the first error or warning, press the F4 key.
   b. Correct the error according to the description in the Errors tab.
   c. Repeat until you have corrected all errors.

3. To close the Output window, press the Alt+1 keys.
Index

A
address 13

B
branch
  ladder logic 11

C
create
  tag 18

I
immediate value
  ladder logic 20

L
ladder logic
  arrange input instructions 14
  arrange output instructions 15
  assign immediate value 20
  branch 11
  develop 13
  rung condition 12

N
name
  tag name 16

R
routine
  verify 22
  rung condition 12

T
tag
  assign 18
  choose name 16
  create 18
Rockwell Automation support

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

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

Installation assistance

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

<table>
<thead>
<tr>
<th>Location</th>
<th>Phone Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States or Canada</td>
<td>1.440.646.3434</td>
</tr>
<tr>
<td>Outside United States or Canada</td>
<td>Use the Worldwide Locator available at <a href="http://www.rockwellautomation.com/locations">http://www.rockwellautomation.com/locations</a>, or contact your local Rockwell Automation representative.</td>
</tr>
</tbody>
</table>

New product satisfaction return

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

<table>
<thead>
<tr>
<th>Location</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td>Contact your distributor. You must provide a Customer Support case number (call the phone number above to obtain one) to your distributor to complete the return process.</td>
</tr>
<tr>
<td>Outside United States</td>
<td>Please contact your local Rockwell Automation representative for the return procedure.</td>
</tr>
</tbody>
</table>

Documentation feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete the feedback form, publication RA-DU002.

www.rockwellautomation.com