

# Logix5000 Controllers Ladder Diagram



**Allen-Bradley**

Catalog Numbers 1756 ControlLogix, 1769 CompactLogix, 1789 SoftLogix, 1794 FlexLogix, PowerFlex 700S with DriveLogix

Programming Manual



## Important User Information

Solid state equipment has operational characteristics differing from those of electromechanical equipment. Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls (publication [SGI-1.1](#) available from your local Rockwell Automation sales office or online at <http://www.rockwellautomation.com/literature/>) describes some important differences between solid state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

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.

---

### IMPORTANT

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

---

### 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

---

### 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.

---

## Table of Contents

### Preface

Purpose of this Manual . . . . .	5
How to Use this Manual . . . . .	5

### Program Ladder Diagram

#### Chapter 1

Introduction . . . . .	7
Instruction . . . . .	7
Branch . . . . .	7
Rung Condition . . . . .	9
Write Ladder Logic . . . . .	10
Choose the Required Instructions . . . . .	10
Arrange the Input Instructions . . . . .	11
Arrange the Output Instructions . . . . .	12
Choose a Tag Name for an Operand . . . . .	12
Enter Ladder Logic . . . . .	13
Append an Element to the Cursor Location . . . . .	14
Drag and Drop an Element . . . . .	14
Assign Instruction Operands . . . . .	15
Create and Assign a New Tag . . . . .	15
Choose a Name or an Existing Tag . . . . .	16
Drag a Tag From the Tags Window . . . . .	16
Assign an Immediate (Constant) Value . . . . .	16
Enter a Rung Comment . . . . .	17
Language Switching . . . . .	17
Verify the Routine . . . . .	18



## Purpose of this Manual

This manual shows how to program Logix5000 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 Logix5000 controllers. For a complete list of common procedures manuals, see the Logix 5000 Controllers Common Procedures Programming Manual, publication 1756-PM001.

The term Logix5000 controller refers to any controller that is based on the Logix5000 operating system, such as:

- CompactLogix controllers
- ControlLogix controllers
- DriveLogix controllers
- FlexLogix controllers
- SoftLogix5800 controllers

## How to Use this Manual

Some text is formatted differently from the rest of the text.

<b>Text that is</b>	<b>Identifies</b>	<b>For example</b>	<b>Means</b>
<i>Italic</i>	the actual name of an item that you see on your screen or in an example	Right-click <i>User-Defined ...</i>	Right-click the item that is named User-Defined.
<i>courier</i>	information that you must supply based on your application (a variable)	Right-click <i>name_of_program ...</i>	You must identify the specific program in your application. Typically, it is a name or variable that you have defined.
enclosed in brackets	a keyboard key	Press [Enter].	Press the Enter key.

**Notes:**

# Program Ladder Diagram

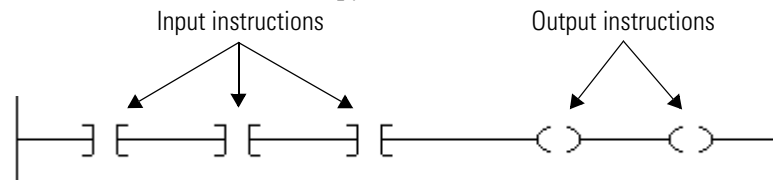
## Introduction

## Instruction

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.

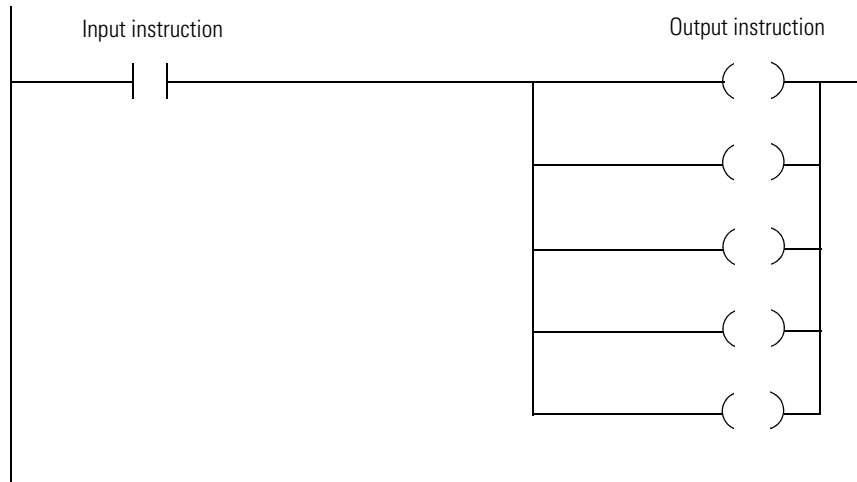


## Branch

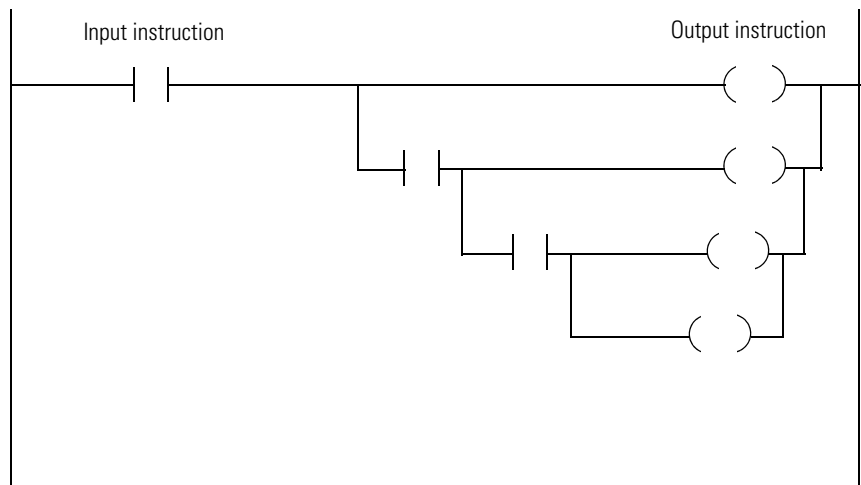
A branch is two or more instructions in parallel.



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.

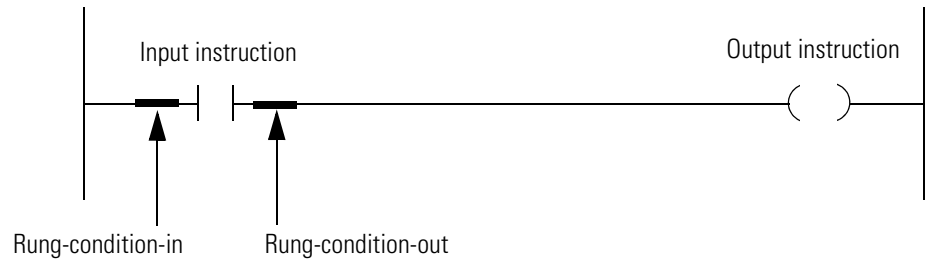


Large rungs with complex, nested branches result in having to scroll through the ladder editor and may end up spanning multiple pages when you print the logic. For easier maintainability, divide the logic into multiple smaller rungs.



## Rung Condition

The controller evaluates ladder instructions based on the rung condition preceding the instruction (rung-condition-in).



Only input instructions 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.

## Write Ladder Logic

## Choose the Required Instructions

1. Separate the conditions to check from the action to take.
2. Choose the appropriate input instruction for each condition and the appropriate output instruction for each action.

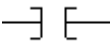
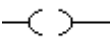
To choose specific instructions, see:

- *Logix5000 Controllers General Instructions Reference Manual*, publication 1756-RM003
- *Logix5000 Controllers Process and Drives Instructions Reference Manual*, publication 1756-RM006
- *Logix5000 Controllers Motion Instruction Set Reference Manual*, publication 1756-RM007

**TIP**

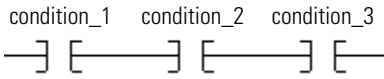
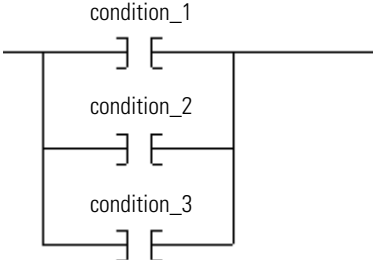
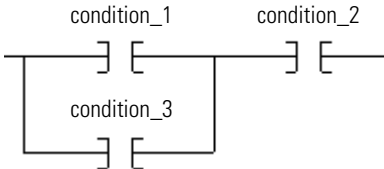
I/O module data updates asynchronously to the execution of logic. If you reference an input multiple times in your logic, the input could change state between separate references. If you need the input to have the same state for each reference, buffer the input value and reference that buffer tag.

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.

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

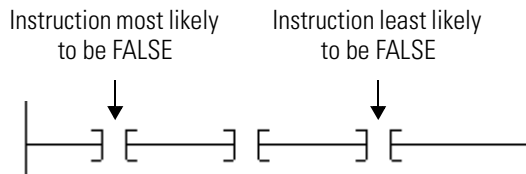
## Arrange the Input Instructions

Determine how to arranged the input instructions on the rung:

To check multiple input conditions when	Arrange the input instructions
All conditions must be met in order to take action For example, If condition_1 AND condition_2 AND condition_3...	In series: 
Any one of several conditions must be met in order to take action For example, If condition_1 OR condition_2 OR condition_3...	In parallel: 
There is a combination of the above For example, If condition_1 AND condition_2... OR If condition_3 AND condition_2...	In combination: 

**TIP**

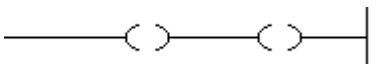
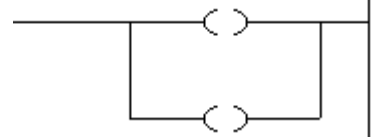

The controller executes all instructions on a rung regardless of their rung-condition-in. For optimal performance of a series of instructions, sequence the instructions from most likely to be false on the left to least likely to be false on the right.



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 per rung of logic:

Option	Example
Place the output instructions in sequence on the rung (serial).	
Place the output instructions in branches (parallel).	
Place the output instructions between input instructions, as long as the last instruction on the rung is an output instruction.	

## Choose a Tag Name for an Operand

Tag names follow these formats:

For a	Specify
Tag	<i>tag_name</i>
Bit number of a larger data type	<i>tag_name.bit_number</i>
Member of a structure	<i>tag_name.member_name</i>
Element of a one dimension array	<i>tag_name[x]</i>
Element of a two dimension array	<i>tag_name[x,y]</i>
Element of a three dimension array	<i>tag_name[x,y,z]</i>
Element of an array within a structure	<i>tag_name.member_name[x]</i>
Member of an element of an array	<i>tag_name[x,y,z].member_name</i>

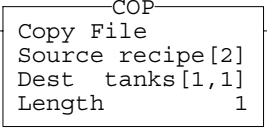
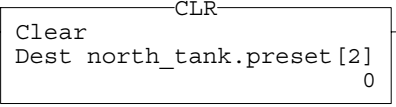
where:

- *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 an additional *member\_name*.

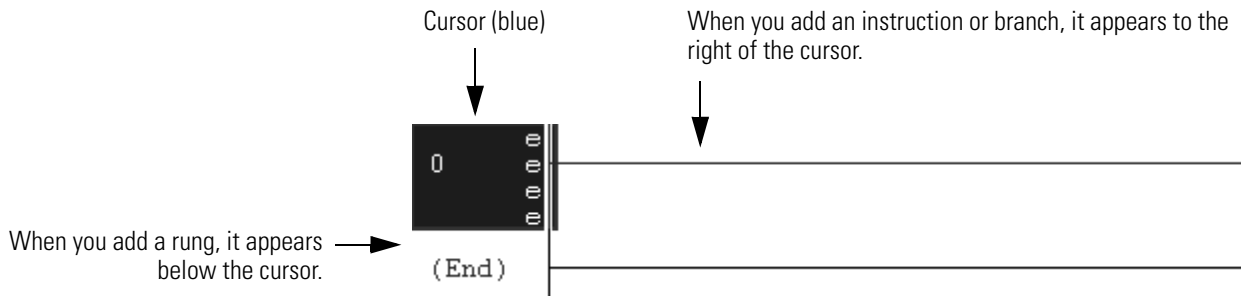
**EXAMPLE**

Choose a Tag Name for an Operand

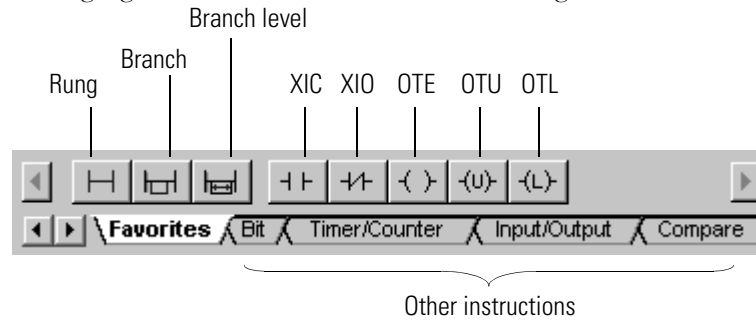
To access	The tag name looks like this
machine_on tag	machine_on
bit number 1 of the one_shots tag	one_shots.1
DN member (bit) of the running_seconds timer	running_seconds.DN
mix member of the north_tank tag	north_tank.mix
element 2 in the recipe array and element 1,1 in the tanks array	 <pre> COP Copy File Source recipe[2] Dest tanks[1,1] Length 1                     </pre>
element 2 in the preset array within the north_tank tag	 <pre> CLR Clear Dest north_tank.preset[2] 0                     </pre>
part_advance member of element 1 in the drill array	drill[1].part_advance

**Enter Ladder Logic**

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

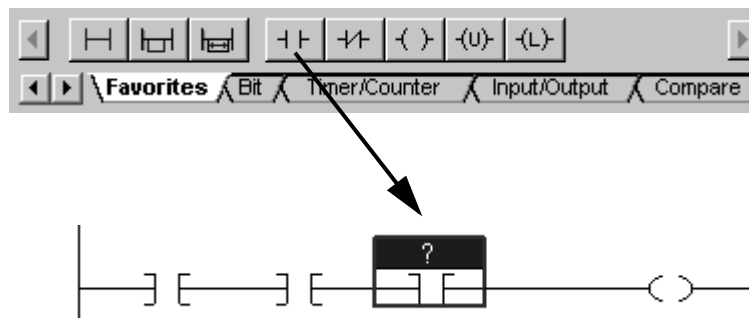
### IMPORTANT

Use caution when copying and pasting components between different versions of RSLogix 5000 programming software. RSLogix 5000 software only supports pasting to the same version or newer version of RSLogix 5000. Pasting to a prior version of RSLogix 5000 software is not supported. When pasting to a prior version, the paste action may succeed but the results may not be as intended.

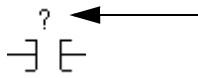
1. Click (select) the instruction, branch, or rung that is above or to the left of where you want to add an element.
2. On the Language Element toolbar, click the button for the element that you want to add.

## 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). For example



## Assign Instruction Operands

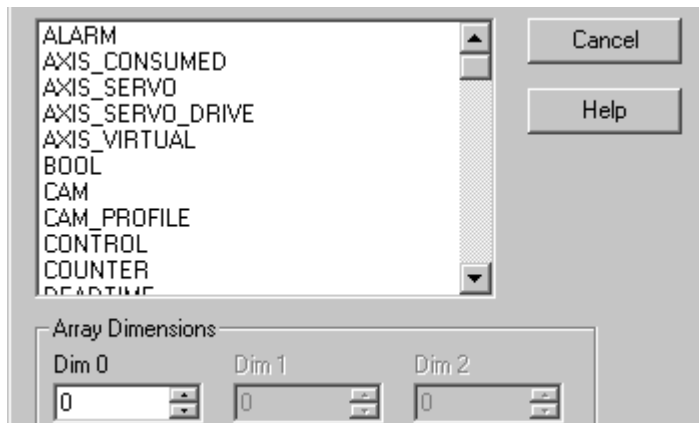


## Create and Assign a New Tag

1. Click the operand area of the instruction.
2. Type a name for the tag and press [Enter].
3. Right-click the tag name and select New “tag\_name”.
4. Click the  button.

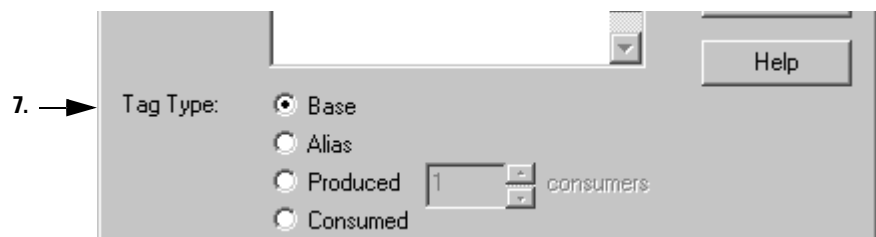


5. Select the data type for the tag.



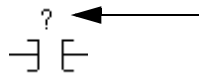
If you want to define the tag as an array, type the number of elements in each dimension.

6. Click OK.
7. Choose the scope for the tag.



8. Click OK.

### Choose a Name or an Existing Tag



1. Double-click the operand area.

A text entry box opens.

2. Click the ▼?
3. Select the name:

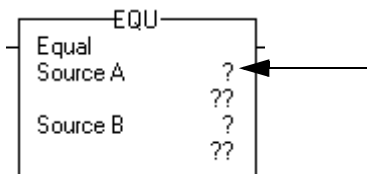
To select a	Do this
Label, routine name, or similar type of name	Click the name.
Tag	Double-click the tag name.
Bit number	A. Click the tag name. B. To the right of the tag name, click ▼. C. Click the required bit.

4. Press [Enter] or click a different spot on the diagram.

### Drag a Tag From the Tags Window

1. Find the tag in the Tags window.
2. Click the tag two or three times until it highlights.
3. Drag the tag to its location on the instruction.

### Assign an Immediate (Constant) Value



1. Click the operand area of the instruction.
2. Type the value and press [Enter].

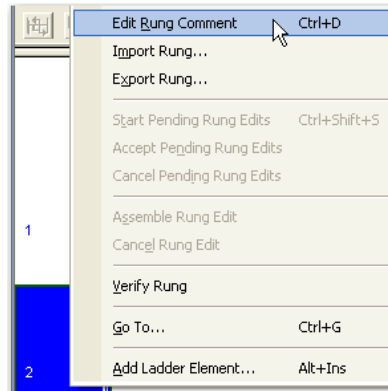


## Enter a Rung Comment

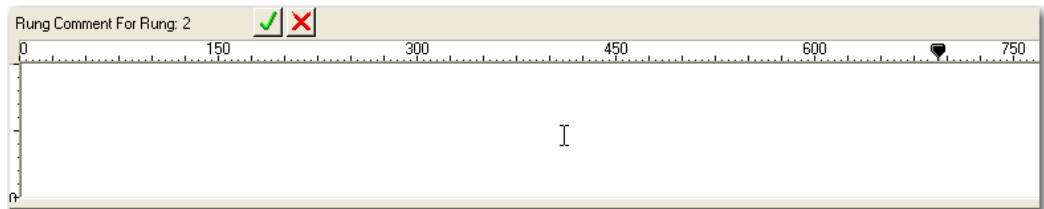
When entering a rung of ladder logic, you can add comments that explain the purpose of your rung.

To enter a rung comment, perform this procedure.

1. Right-click on the rung number of your ladder logic.



The Rung Text Edit dialog appears.



2. Enter your rung comment in the Text Edit dialog, and click the green check button to confirm your comments or the red X to cancel your comments.

## Language Switching

With RSLogix 5000 software, version 17, 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 will support and set the current, default, and optional custom localized language. The software 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 RSLogix 5000 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 in RSLogix 5000 software, you can dynamically switch between languages as you use the software.


Project documentation that supports multiple translations 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), periodically verify your work:

1. In the top-most toolbar of the RSLogix 5000 window, click .
2. If any errors are listed at the bottom of the window:
  - a. To go to the first error or warning, press [F4].
  - b. Correct the error according to the description in the Results window.
  - c. Go to step 1.
3. To close the Results window, press [Alt] + [1].



# 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 manuals, a knowledge base of FAQs, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools.

For an additional level of technical phone support for installation, configuration, and troubleshooting, we offer TechConnect support programs. For more information, contact your local distributor or Rockwell Automation representative, or visit <http://www.rockwellautomation.com/support/>.

## Installation Assistance

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

United States or Canada	1.440.646.3434
Outside United States or Canada	Use the <a href="#">Worldwide Locator</a> at <a href="http://www.rockwellautomation.com/support/americas/phone_en.html">http://www.rockwellautomation.com/support/americas/phone_en.html</a> , or contact your local Rockwell Automation representative.

## 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.

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

## Documentation Feedback

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

**[www.rockwellautomation.com](http://www.rockwellautomation.com)**

---

### Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444

Europe/Middle East/Africa: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication 1756-PM008C-EN-P - October 2009

Supersedes Publication 1756-PM008B-EN-P - July 2008

Copyright © 2009 Rockwell Automation, Inc. All rights reserved. Printed in the U.S.A.