



Allen-Bradley

Logix5000™ Controllers Quick Start

**1756 ControlLogix®,
1769 CompactLogix™,
1789 SoftLogix™,
1794 FlexLogix™, PowerFlex®
700S with DriveLogix™**

Quick Start

**Rockwell
Automation**

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.ab.com/manuals/gi>) 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 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
- recognize the consequence

IMPORTANT

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

SHOCK HAZARD



Labels may be located on or inside the drive to alert people that dangerous voltage may be present.

Introduction

This release of this document contains new and updated information. To find new and updated information, look for change bars, as shown next to this paragraph.

New or Updated Information

The document contains the following changes:

This change:	Starts on page:
Program and Test a Simple Project—new chapter that highlights the minimum steps for programming a Logix5000 controller	1-1
Export/Import Ladder Logic	3-6
Enter a Sequential Function Chart	3-16
Assign Operands to an Instruction in Structured Text	3-15
Describe a User-Defined Data Type	4-2
Add Rung Comments	4-4
Enter and Edit Rung Comments Using Microsoft® Excel	4-5
Add Comments to a Function Block Diagram or SFC	4-7
Add Comments to Structured Text	4-9
Edit Logic While Online—addition of SFC and structured text	6-1
Finalize All Edits in a Program	6-5
Browse Logic for a Tag, Instruction, Comment, Etc	7-7

Notes:

When to Use This Manual

The manual is one of various Logix5000 manuals.

You are
here



To:	See:
get started with a Logix5000 controller	<i>Logix5000™ Controllers Quick Start</i> , publication 1756-QS001
Look up abbreviated information and procedures regarding programming languages, instructions, communications, and status	<i>Logix5000 Controllers System Reference</i> , publication 1756-QR007
program a Logix5000 controller—detailed and comprehensive information	<i>Logix5000 Controllers Common Procedures</i> , publication 1756-PM001
program a specific Logix5000 programming instruction	<ul style="list-style-type: none"> • <i>Logix5000 Controllers General Instructions Reference Manual</i>, publication 1756-RM003 • <i>Logix5000 Controllers Process and Drives Instructions Reference Manual</i>, publication 1756-RM006 • <i>Logix5000 Controllers Motion Instruction Set Reference Manual</i>, publication 1756-RM007
import or export a Logix5000 project or tags from or to a text file	<i>Logix5000 Controllers Import/Export Reference Manual</i> , publication 1756-RM084
convert a PLC-5 or SLC 500 application to a Logix5000 project	<i>Logix5550 Controller Converting PLC-5 or SLC 500 Logic to Logix5550 Logic Reference Manual</i> , publication 1756-6.8.5
integrate a specific Logix5000 controller within a system of controllers, I/O modules, and other devices	<ul style="list-style-type: none"> • <i>CompactLogix System User Manual</i>, publication 1769-UM007 • <i>ControlLogix System User Manual</i>, publication 1756-UM001 • <i>DriveLogix Controller User Manual</i>, publication 20D-UM002 • <i>FlexLogix System User Manual</i>, publication 1794-UM001 • <i>SoftLogix5800 System User Manual</i>, publication 1789-UM002
control devices over an EtherNet/IP network	<i>EtherNet/IP Modules in Logix5000 Control Systems User Manual</i> , publication ENET-UM001
control devices over an ControlNet™ network	<i>ControlNet Modules in Logix5000 Control Systems User Manual</i> , publication CNET-UM001
control devices over an DeviceNet™ network	<i>DeviceNet Modules in Logix5000 Control Systems User Manual</i> , publication DNET-UM004

Purpose of This Manual

This manual provides a starter set of procedures to:

- establish communication with a Logix5000 controller
- program a Logix5000 controller
- perform online maintenance tasks such as search and edit logic, run a histogram, clear faults, and force I/O values.

A Logix5000 controller is any of the following:

- 1756 ControlLogix® controllers
- 1769 CompactLogix™ controllers
- 1789 SoftLogix5800™ controllers
- 1794 FlexLogix™ controllers
- PoweFlex®700S with DriveLogix™ controllers

Who Should Use this Manual

This manual is for those who program or maintain industrial automation systems.

To use this manual, you must already have experience with:

- programmable controllers
- industrial automation systems
- personal computers and Windows® 95, Windows 98, Windows NT®, or Windows 2000 operating system

How to Use this Manual

As you use this manual, you will see some terms that are formatted differently from the rest of the text:

Text that is:	Identifies:	For example:	Means:
<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 on 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.

	Chapter 1	
Program and Test a Simple Project	Using This Chapter	1-1
	Create a Project for the Controller	1-2
	Create a Project	1-3
	Conventions for Names	1-3
	Add Your I/O Modules	1-4
	Look at Your I/O Data	1-5
	Enter Ladder Logic	1-7
	Open a Routine	1-7
	Enter Ladder Logic	1-8
	Enter a Function Block Diagram	1-9
	Create a Routine	1-9
	Call the Routine	1-10
	Enter a Function Block Diagram	1-11
	Configure a Function Block Instruction	1-12
	Assign Alias Tags for Your Devices	1-13
	Show or Hide Alias Information	1-14
	Establish a Serial Connection to the Controller	1-15
	Download a Project to the Controller	1-17
	Select the Operating Mode of the Controller	1-19
	Chapter 2	
Organize a Project	Using This Chapter	2-1
	Configure the Task Execution	2-2
	Create Additional Programs	2-3
	Create User-Defined Data Types	2-5
	Define Your Routines	2-7
	Define a Routine for Each Section of Your Machine or Process	
	2-7	
	Identify the Programming Languages That Are Installed	2-8
	Assign a Programming Language to Each Routine	2-8
	Divide Each Routine Into More Meaningful Increments	2-9
	Assign Main Routines	2-10
	Configure the Controller	2-11
	Configure I/O Modules	2-12
	Chapter 3	
Program a Project Offline	Using This Chapter	3-1
	Enter Ladder Logic	3-2
	Drag and Drop an Element	3-3
	Use the Keyboard to Add an Element	3-4
	Enter Logic Using ASCII Text	3-4
	Enable Quick Keys	3-5
	Export/Import Ladder Logic	3-6
	When You Import Rungs	3-6

Export Rungs	3-7
Import Rungs	3-8
Check Alias Tags	3-8
Enter a Function Block Diagram	3-9
Use the Keyboard to Add an Element	3-10
Connect Elements	3-10
Resolve a Loop	3-11
Add Sheet	3-11
Use a Faceplate for a Function Block	3-12
Set Up a Topic	3-13
Add a Faceplate to Microsoft Excel Software	3-13
Enter Structured Text	3-14
Browse For an Instruction	3-15
Assign Operands to an Instruction	3-15
Enter a Sequential Function Chart	3-16
Enter an SFC	3-17
Assign Operands	3-18
Create a Tag	3-19
Select an Existing Tag	3-19
Verify a Project	3-20
Guidelines for Tags	3-22

Chapter 4

Document a Project

Using This Chapter	4-1
Describe a User-Defined Data Type	4-2
Turn Pass-Through and Append Descriptions On or Off	4-3
Paste a Pass-Through Description	4-3
Add Rung Comments	4-4
Enter and Edit Rung Comments Using Microsoft® Excel	4-5
Export the Existing Comments	4-5
Edit the Export File	4-6
Import the New Comments	4-6
Add Comments to a Function Block Diagram or SFC	4-7
Set the Word Wrap Option	4-7
Add a Text Box	4-8
Add Comments to Structured Text	4-9

Chapter 5

Go Online to the Controller

Using This Chapter	5-1
Establish EtherNet/IP Communication with the Controller	5-2
Equipment and Information That You Need	5-3
Connect Your EtherNet/IP Device and Computer	5-3
Assign an IP Address to the Controller or Communication Module	5-4
Configure an Ethernet Driver	5-5

	Go Online to a Controller	5-6
	If Your Computer Has the Project For the Controller... .	5-6
	If Your Computer <i>Does Not</i> Have the Project For the Controller.... .	5-7
	Chapter 6	
Program a Project Online	Using This Chapter	6-1
	Edit Logic While Online	6-1
	Start a Pending Edit	6-3
	Make and Accept Your Edits	6-3
	Test the Edits.	6-4
	Assemble and Save the Edits	6-4
	Finalize All Edits in a Program	6-5
	Chapter 7	
Troubleshoot the Controller	Using This Chapter	7-1
	Troubleshoot I/O Communication	7-2
	Clear a Major Fault	7-4
	Search a Project	7-5
	Browse Logic	7-7
	Force an I/O Value	7-8
	Create and Run a Trend (Histogram)	7-11
	View Scan Time	7-13

Program and Test a Simple Project

Using This Chapter

This chapter introduces the basic programming sequence for a Logix5000™ controller.

- It covers the steps required to develop and test a ladder or function block diagram.
- The examples in the chapter show how to control a digital or analog output based on the state of a digital or analog input.

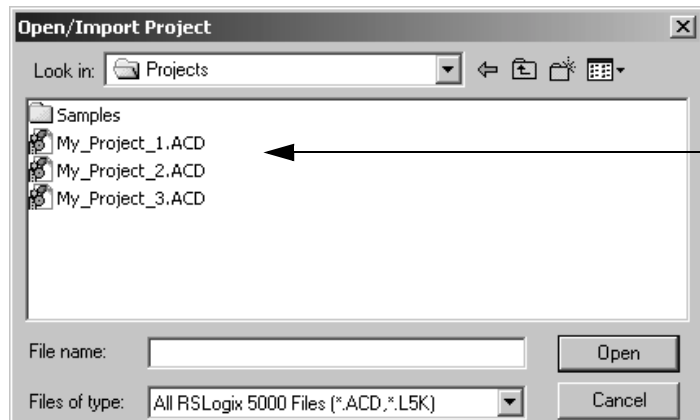
To program and test a simple project:

Step:	Page:
Create a Project for the Controller	1-2
Add Your I/O Modules	1-4
Look at Your I/O Data	1-5
Enter Ladder Logic	1-7
Enter a Function Block Diagram	1-9
Assign Alias Tags for Your Devices	1-13
Establish a Serial Connection to the Controller	1-15
Download a Project to the Controller	1-17
Select the Operating Mode of the Controller	1-19

The rest of the chapters in this publication provide more detailed information on how to program, edit, and troubleshoot a project.

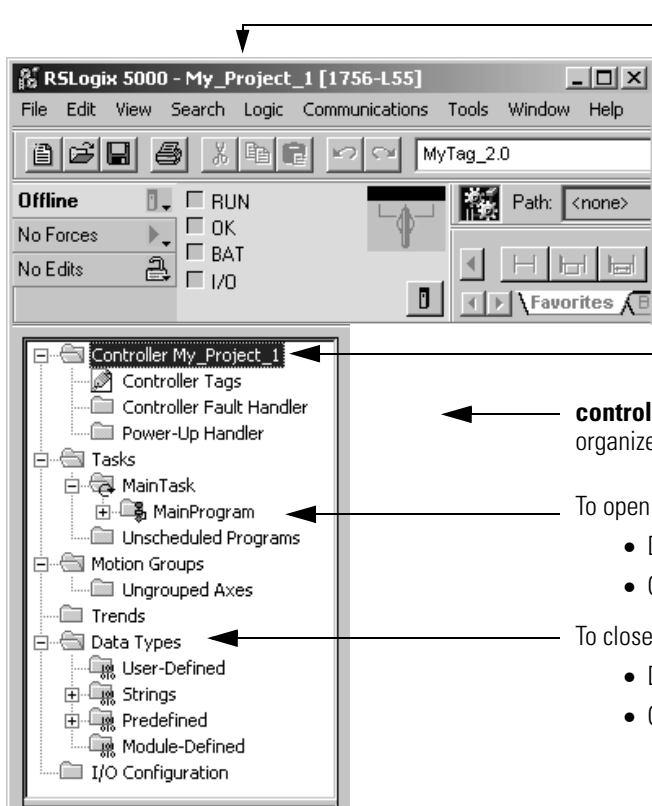
Create a Project for the Controller

To configure and program a Logix5000 controller, you use RSLogix™ 5000 software to create and manage a project for the controller.



project – The file on your workstation (or server) that stores the logic, configuration, data, and documentation for a controller.

- The file for the project has an .ACD extension.
- When you create a project, the project name is the same as the name of the controller.
- The controller name is independent of the project name. You can rename either the project name or the controller name.



name of the project

If you rename the project or controller, both names are shown.

name of the controller

controller organizer – graphical overview of the project. Use the controller organizer to navigate to the various components of a project.

To open a folder and show its contents, either:

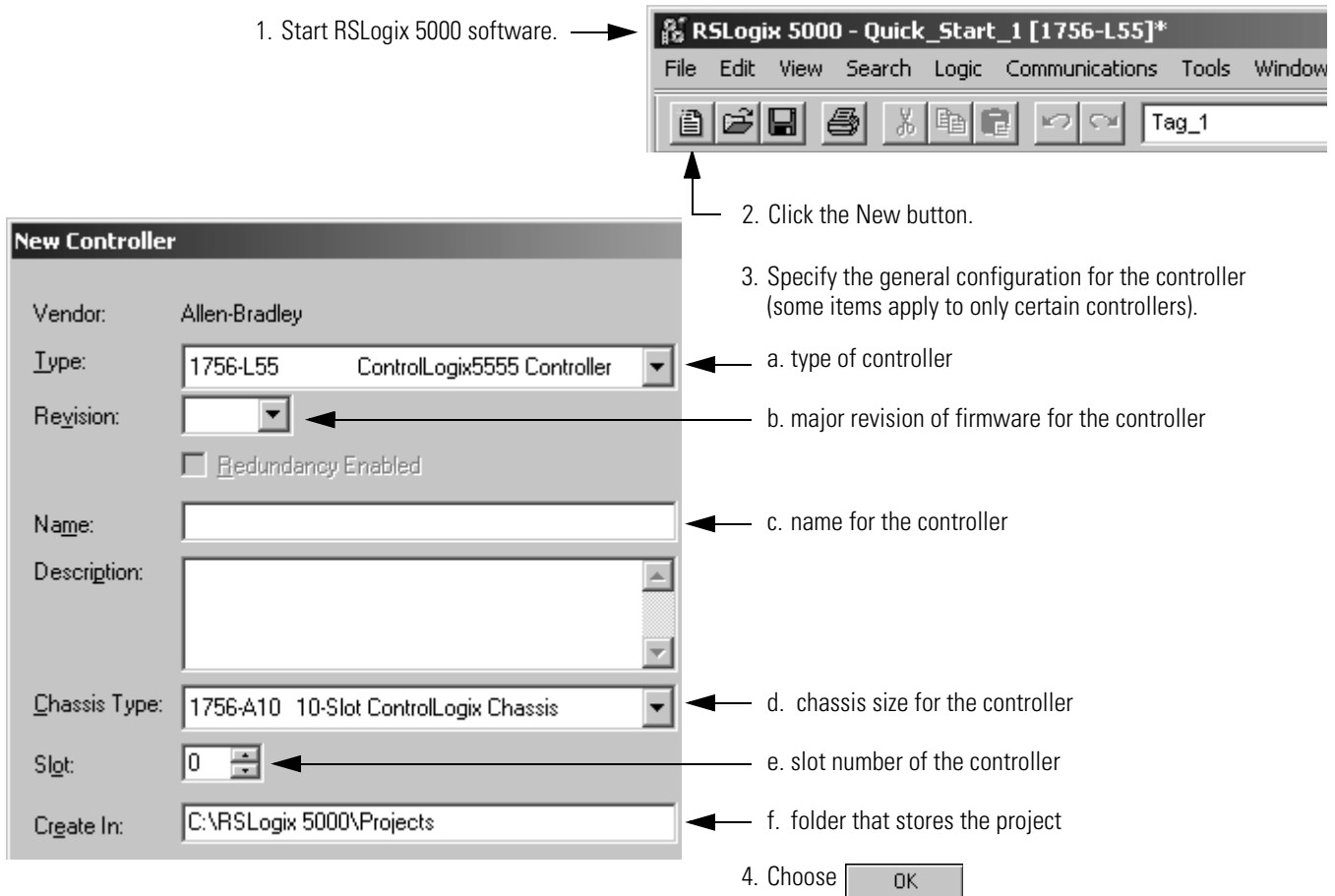
- Double-click the folder.
- Click the + sign.

To close a folder and hide its contents, either:

- Double-click the folder.
- Click the – sign.

Create a Project

1. Start RSLogix 5000 software. →



2. Click the New button.

3. Specify the general configuration for the controller (some items apply to only certain controllers).

a. type of controller

b. major revision of firmware for the controller

c. name for the controller

d. chassis size for the controller

e. slot number of the controller

f. folder that stores the project

4. Choose **OK**

Conventions for Names

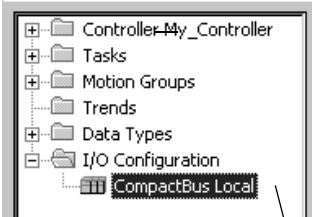
Throughout a Logix5000 project, you define names for the different elements of the project such as the controller, data addresses (tags), routines, I/O modules, etc. As you enter names, follow these rules:

- only letters, numbers, and underscores (_)
- must start with a letter or an underscore
- ≤ 40 characters
- no consecutive or trailing underscores
- *not* case sensitive

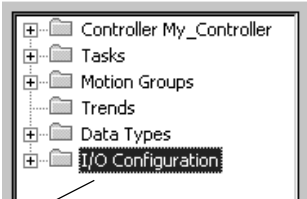
Add Your I/O Modules

To communicate with an I/O modules in your system, you add the modules to the I/O Configuration folder of the controller. The properties you select for each module defines the behavior of the module.

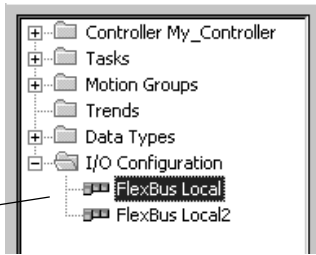
CompactLogix Controller

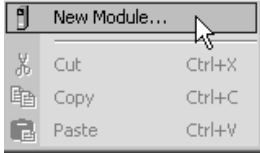


ControlLogix Controller

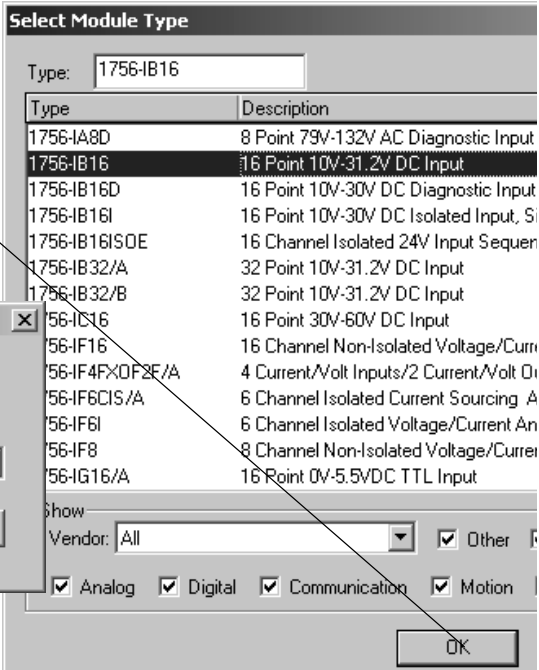


FlexLogix Controller



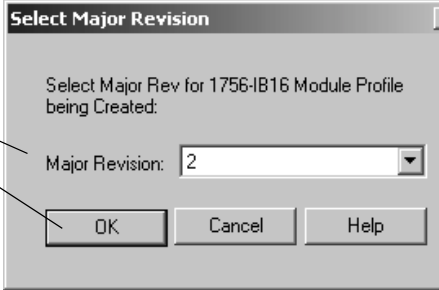


1. Right-click and choose *New Module*.
2. Select the type of module.

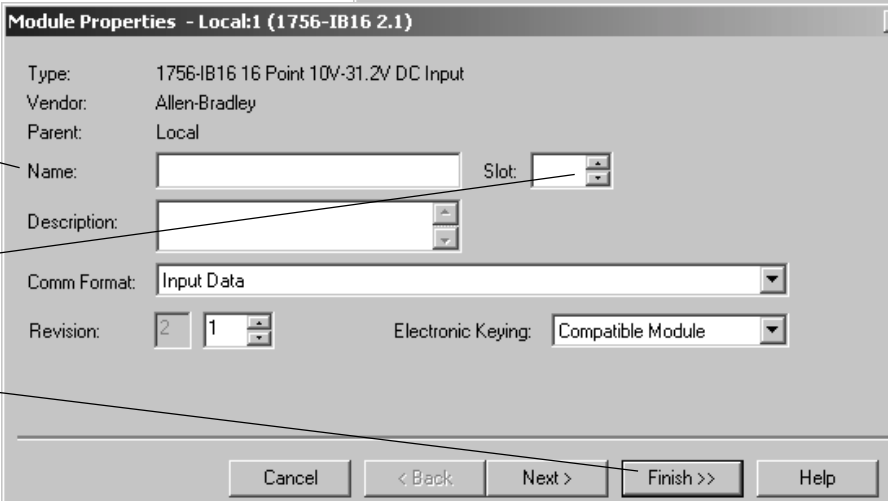


Type	Description
1756-IA8D	8 Point 79V-132V AC Diagnostic Input
1756-IB16	16 Point 10V-31.2V DC Input
1756-IB16D	16 Point 10V-30V DC Diagnostic Input
1756-IB16I	16 Point 10V-30V DC Isolated Input, Si
1756-IB16ISOE	16 Channel Isolated 24V Input Sequen
1756-IB32/A	32 Point 10V-31.2V DC Input
1756-IB32/B	32 Point 10V-31.2V DC Input
1756-IC16	16 Point 30V-60V DC Input
1756-IF16	16 Channel Non-Isolated Voltage/Curr
1756-IF4FXDF2F/A	4 Current/Volt Inputs/2 Current/Volt O
1756-IF6CIS/A	6 Channel Isolated Current Sourcing A
1756-IF6I	6 Channel Isolated Voltage/Current An
1756-IF8	8 Channel Non-Isolated Voltage/Curre
1756-IG16/A	16 Point 0V-5.5VDC TTL Input

3. Select the revision of module.



4. Type a name for the module (up to 40 characters with *no* spaces).
5. Select the location of the module in the chassis or rail.
6. Accept the default configuration for the module.



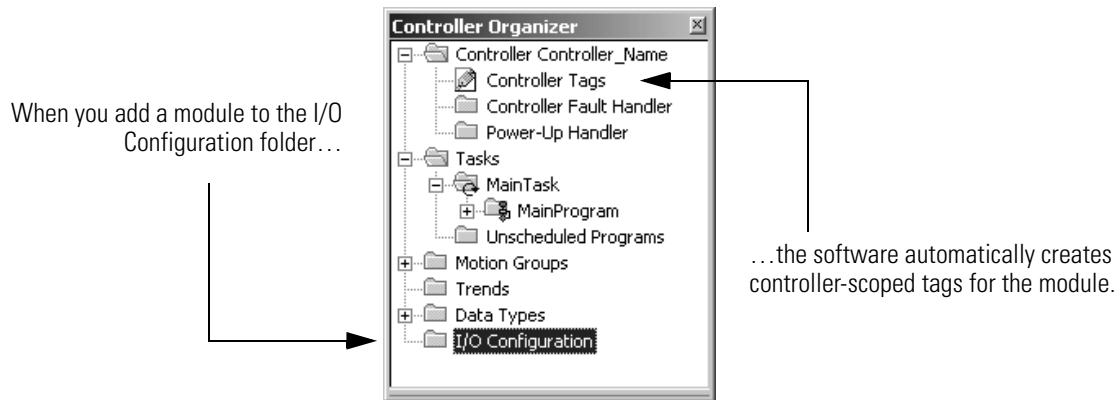
Module Properties - Local:1 (1756-IB16 2.1)

Type: 1756-IB16 16 Point 10V-31.2V DC Input
Vendor: Allen-Bradley
Parent: Local
Name: Slot:
Description:
Comm Format: Input Data
Revision: 2 1 Electronic Keying: Compatible Module

Buttons: Cancel < Back Next > Finish >> Help

Look at Your I/O Data

I/O information is presented as a set of tags.

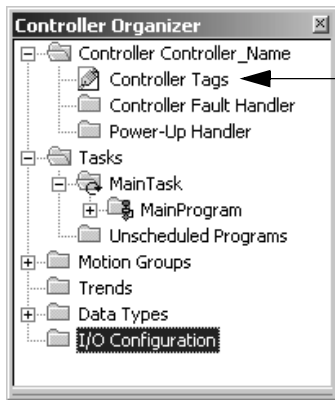


An I/O address follows this format:

Location *:Slot* *:Type* *.Member* *.SubMember* *.Bit*

= Optional

Where:	Is:
<i>Location</i>	Network location LOCAL = same chassis or DIN rail as the controller ADAPTER_NAME = identifies remote communication adapter or bridge module
<i>Slot</i>	Slot number of I/O module in its chassis or DIN rail
<i>Type</i>	Type of data I = input O = output C = configuration S = status
<i>Member</i>	Specific data from the I/O module; depends on what type of data the module can store. <ul style="list-style-type: none"> For a digital module, a Data member usually stores the input or output bit values. For an analog module, a Channel member (CH#) usually stores the data for a channel.
<i>SubMember</i>	Specific data related to a Member.
<i>Bit</i>	Specific point on a digital I/O module; depends on the size of the I/O module (0-31 for a 32-point module)



1. Right-click and choose *Monitor Tags*.

Values are shown in the following styles:

Style	Base	Notation
Binary	2	2#
Decimal	10	NA
Hexadecimal	16	16#
Octal	8	8#
Exponential	NA	0.0000000e+000
Float	NA	0.0

A blue arrow indicates that when you change the value, it immediately takes effect.

Tag Name	Value	Force Mask	Style
+ Local:0:C	{...}	{...}	
+ Local:0:I	{...}	{...}	
- Local:0:O	{...}	{...}	
- Local:0:O.Data	2#000...		Binary
- Local:0:O.Data.0	0		Decimal
- Local:0:O.Data.1	0		Decimal
- Local:0:O.Data.2	0		Decimal

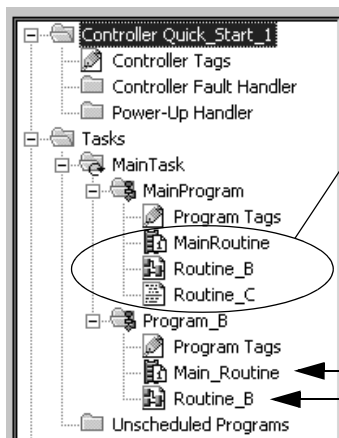
2. To see a value in a different style, select the desired style.

3. To change a value, click the Value cell, type the new value, and press the [Enter] key.

4. To expand a tag and show its members, click the + sign.

Enter Ladder Logic

For a Logix5000 controller, you enter your logic in routines.



routine – provide the executable code (logic) for a program (similar to a program file in a PLC or SLC controller).

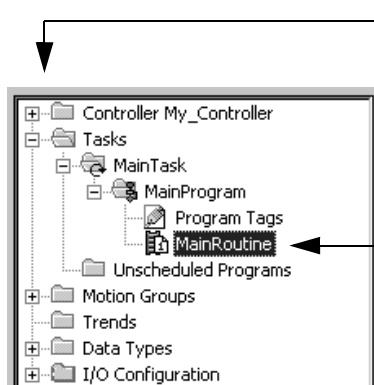
main routine – For each program, you assign a main routine.

- When the program executes, its main routine automatically executes.
- Use the main routine to control the execution of the other routines in the program.
- To call (execute) another routine (subroutine) within the program, use a Jump to Subroutine (JSR) instruction.

subroutine – Any routine other than the main routine or fault routine. To execute a subroutine, use a Jump to Subroutine (JSR) instruction in another routine, such as the main routine.

Open a Routine

When you create a project, the software automatically creates a main routine that uses the ladder diagram programming language.



To open a folder and show its contents, either:

- Double-click the folder.
- Click the + sign.

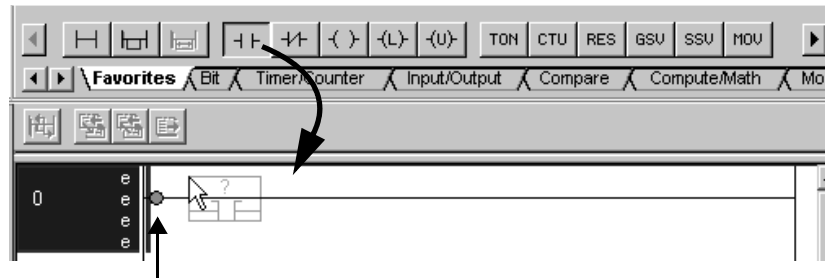
To open a routine, double-click the routine.

Enter Ladder Logic

One way to enter logic is to drag buttons from a toolbar to the desired location.

To add ladder logic, drag the button for the rung or instruction directly to the desired location.

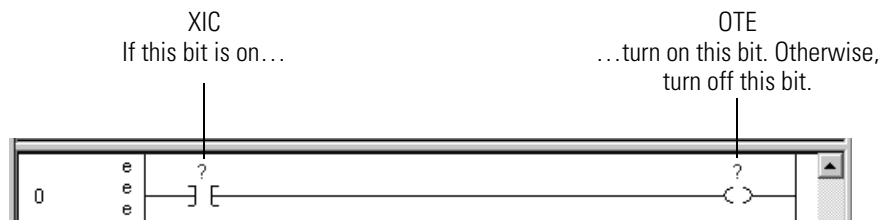
You can enter your logic and leave the operands undefined. After you enter a section of logic, go back and assign the operands.



A green dot shows a valid placement location (drop point).

EXAMPLE

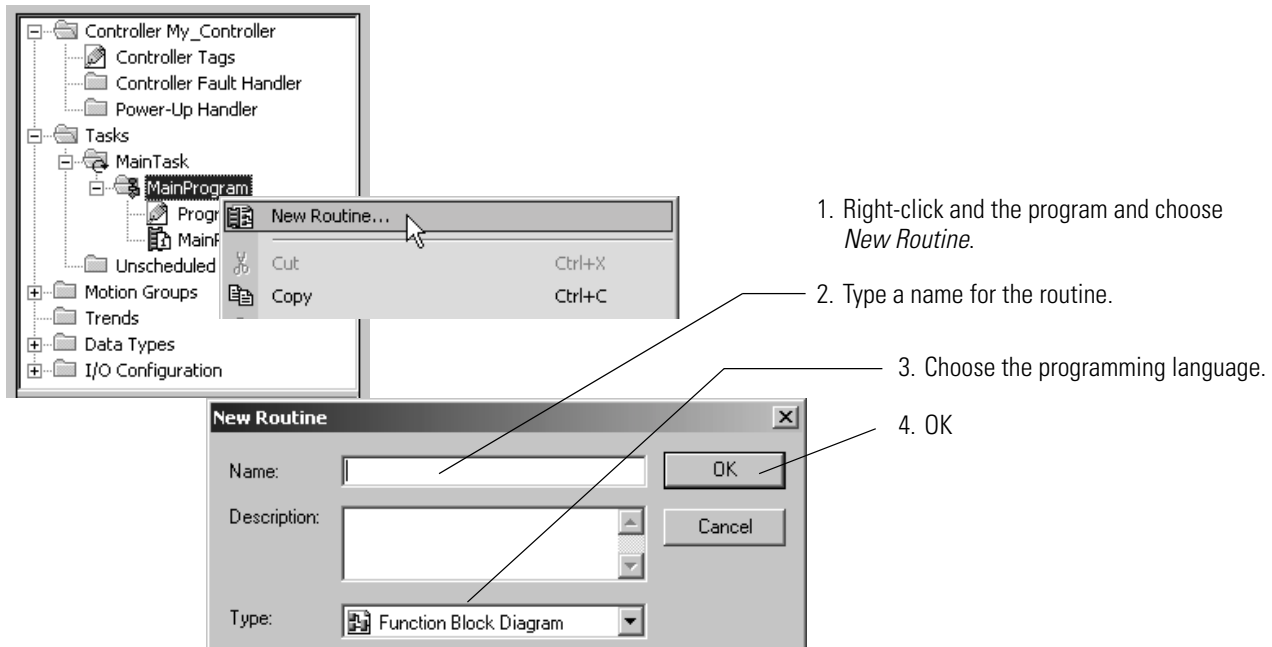
In the following example, an Examine If Closed (XIC) instruction checks the on/off state of a pushbutton. If the pushbutton is on, the Output Energize (OTE) instruction turns on a light.



Enter a Function Block Diagram

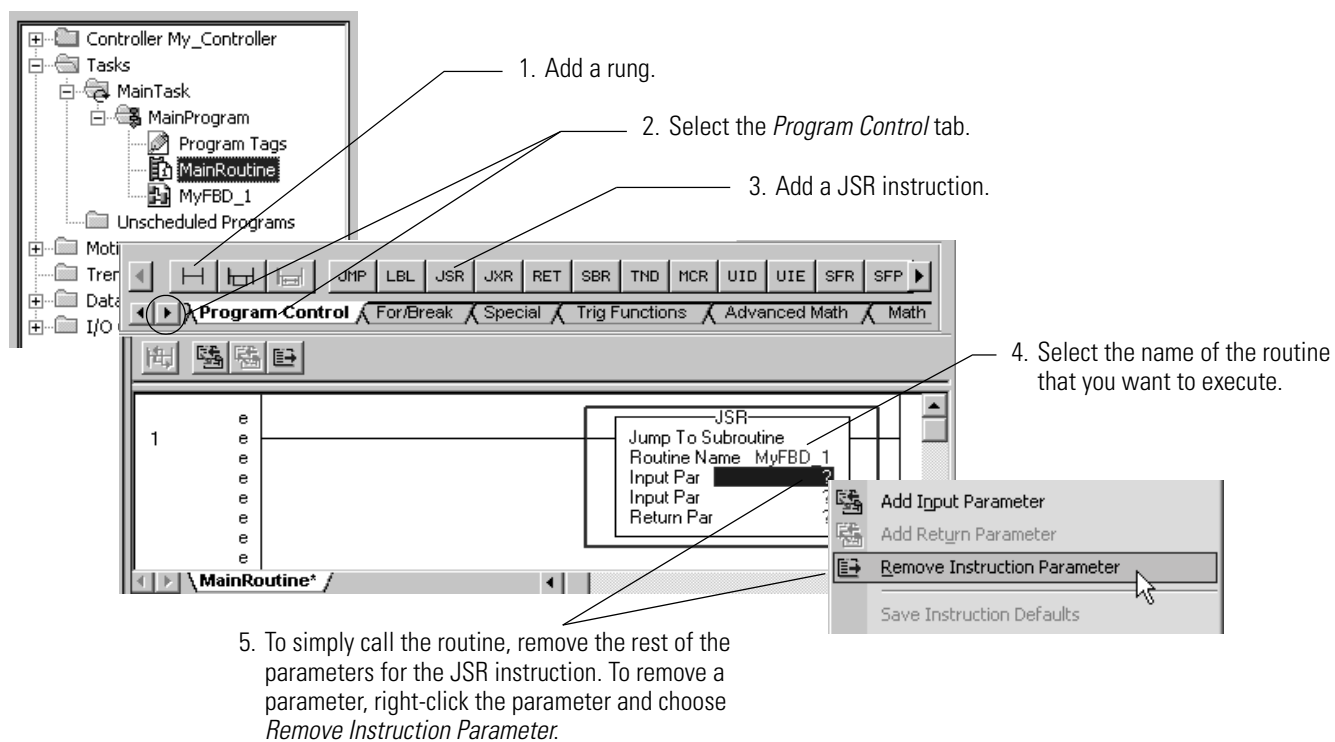
Create a Routine

Each routine in your project uses a specific programming language. To program in a different language, such as function block diagram, create a new routine.

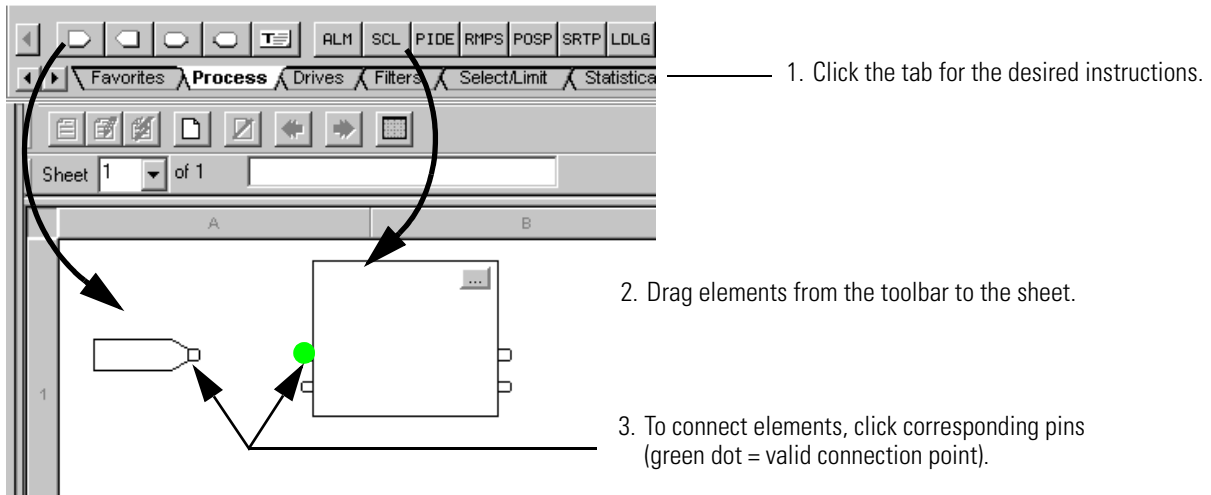


Call the Routine

To execute a routine other than the main routine, use a Jump to Subroutine (JSR) instruction to call the routine.

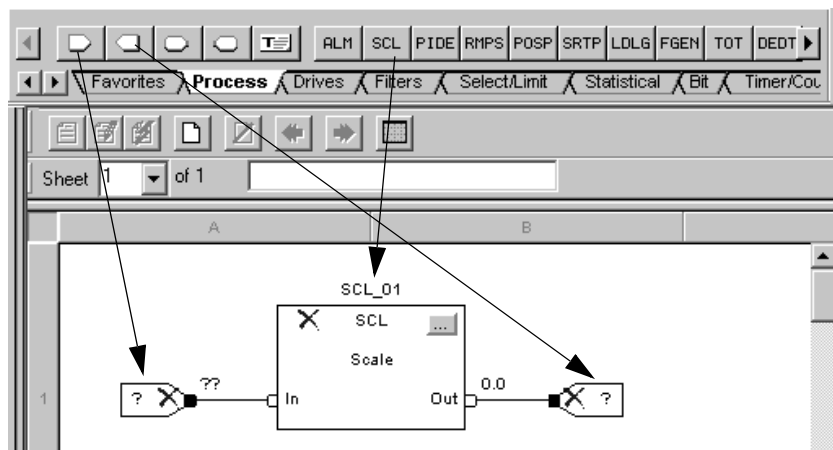


Enter a Function Block Diagram



EXAMPLE

In the following example, an Input Reference (IREF) reads the value of an analog input and sends the value to a Scale (SCL) instruction. The SCL instruction converts the value to engineering uses and sends it to an Output Reference (OREF). The OREF writes the value to an analog output.



Configure a Function Block Instruction

To assign specific values (parameters) to a function block:

1. Click the configuration button.

2. To change the value of a parameter, click the value cell, type the new value, and press [Enter].

For example, in the SCL instruction, specify the following parameters:

- InRawMax – maximum input value
- InRawMin – minimum input value
- InEUMax – maximum engineering value
- InEUMin – minimum engineering value

3. OK

Vis	Name	Value	Type	Description
<input type="checkbox"/>	EnableIn	1	BOOL	Enable Input. If False, the...
<input checked="" type="checkbox"/>	In	0.0	REAL	The analog signal input to ...
<input type="checkbox"/>	InRawMax	0.0	REAL	The maximum value attain...
<input type="checkbox"/>	InRawMin	0.0	REAL	The minimum value attain...
<input type="checkbox"/>	InEUMax	0.0	REAL	The maximum scaled valu...
<input type="checkbox"/>	InEUMin	0.0	REAL	The minimum scaled value...
<input type="checkbox"/>	Limiting	0	BOOL	Limiting selector. If TRUE,...
<input type="checkbox"/>	EnableOut	0	BOOL	Enable Output.
<input checked="" type="checkbox"/>	Out	0.0	REAL	This is the output of the S...
<input type="checkbox"/>	MaxAlarm	0	BOOL	The above maximum input...
<input type="checkbox"/>	MinAlarm	0	BOOL	The below minimum input ...
<input type="checkbox"/>	Status	16#0000_0000	DINT	Bit mapped status of the f...
<input type="checkbox"/>	InstructFault	0	BOOL	Instruction generated a fault
<input type="checkbox"/>	InRawRangeInv	0	BOOL	InRawMin <= InRawMax

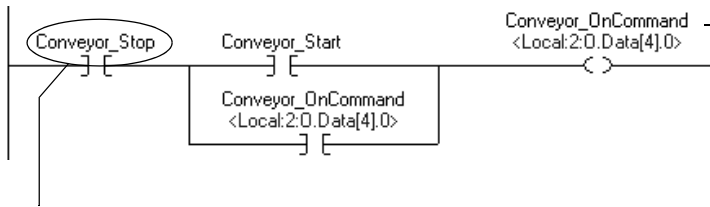
Status: OK

Execution Order Number: <routine not verified>

OK Cancel Apply Help

Assign Alias Tags for Your Devices

While you can use the input and output tags of a module directly in your logic, it is a lot easier to use alias tags.



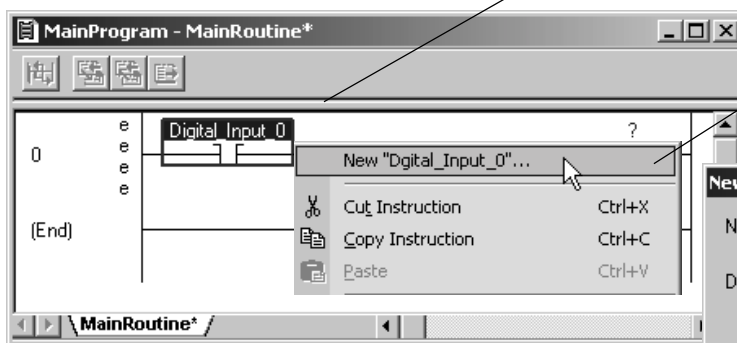
As an option, create tags that describe each device without pointing them to the actual addresses of the devices. Later, convert the tags to aliases for the data of the devices.

alias tag – a tag that represents another tag

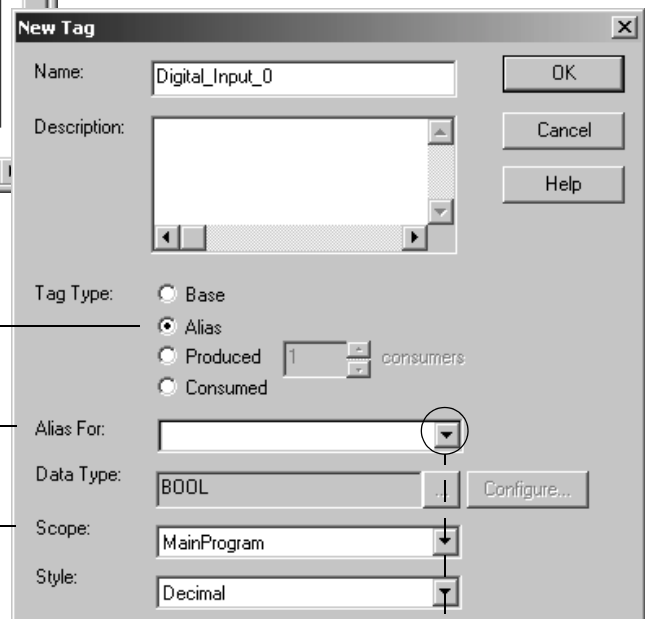
- Both tags share the same data.
- When the data changes, both tags change.
- An alias tag provides a descriptive name for data, such as DeviceNet input or output data.
- If the location of the data changes, simply point the alias tag lets to the new location without editing your logic.

1. Enter your logic.

2. Type a descriptive tag name for the device.



3. Right-click the tag name and choose *New...*

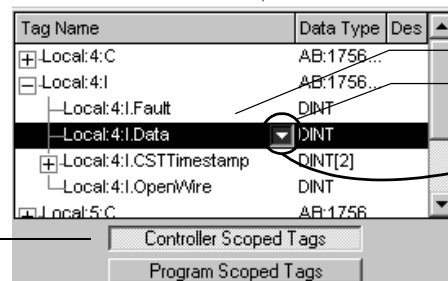


4. Select the *Alias* button.

5. Select the tag that this alias tag represents.

6. Select the scope for the alias tag.

7. Choose *OK*.



Select the address of the data.

To select a bit, click the ▼.

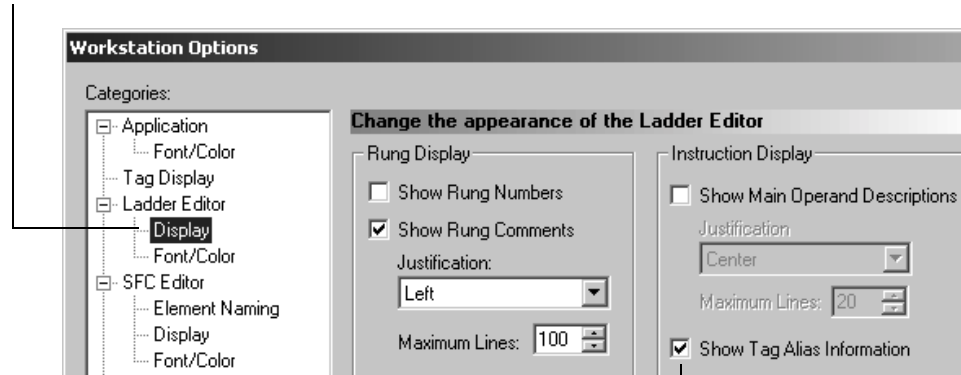
0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31

Look in the controller-scoped tags.

Show or Hide Alias Information

To show or hide that alias information for a tag:

1. Choose *Tools* ⇒ *Options*.
2. Select the *Ladder Editor Display* category.

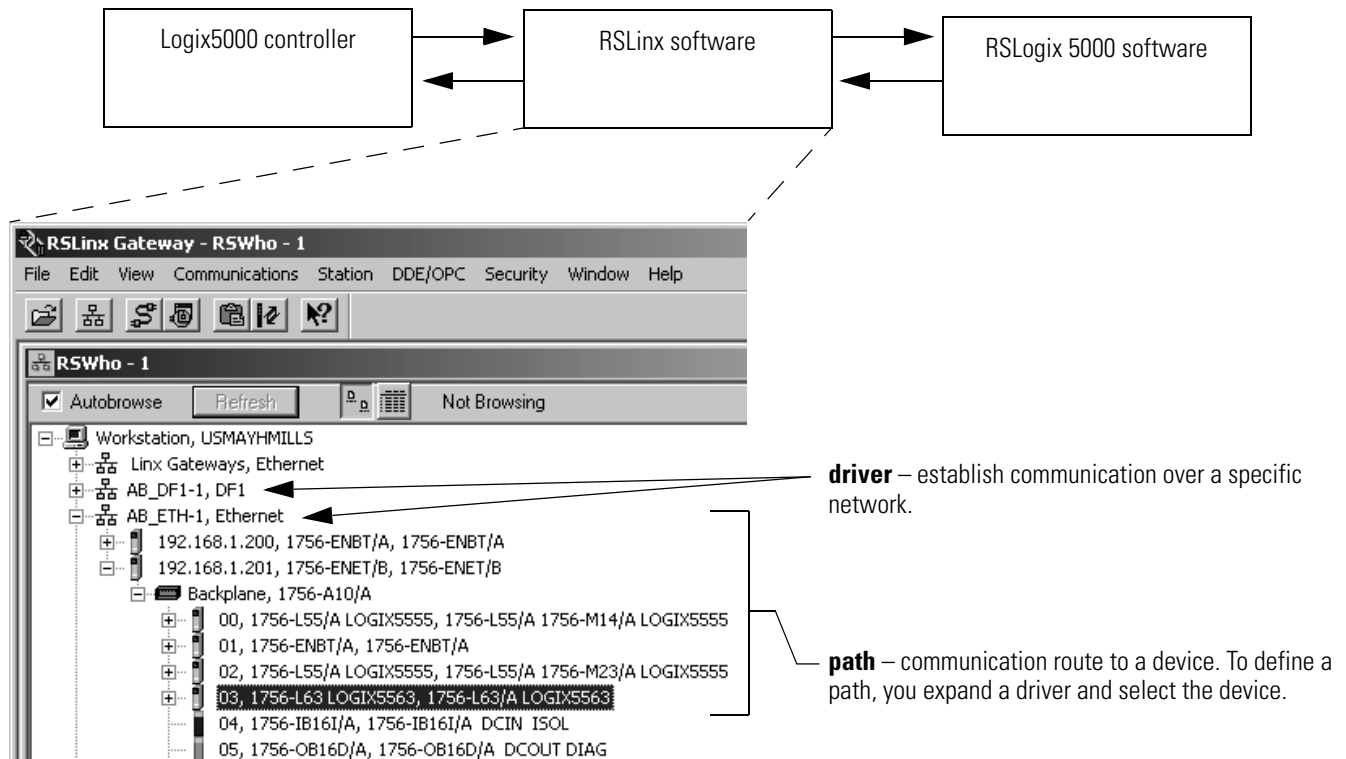


3. Check or uncheck this box.

4. Choose .

Establish a Serial Connection to the Controller

RSLink® software handles communication between Logix5000 controllers and your software programs, such as RSLogix 5000 software. To communicate with a controller (e.g., download, monitor data), configure RSLink software for the required communication.



Use a serial cable to establish a point-to-point connection between the serial ports on your computer and controller.

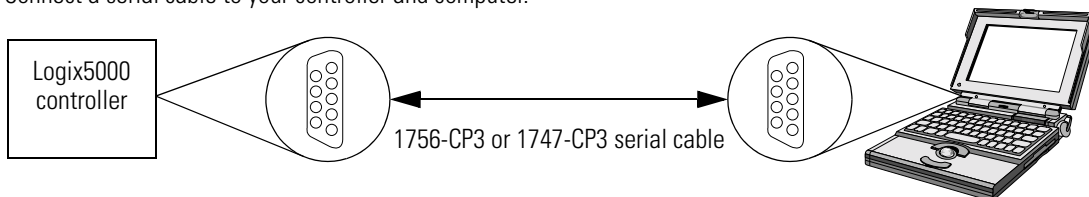
WARNING



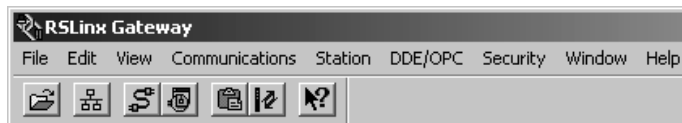
If you connect or disconnect the serial cable with power applied to this module or the serial device on the other end of the cable, an electrical arc can occur. This could cause an explosion in hazardous location installations.

Be sure that power is removed or the area is nonhazardous before proceeding.

1. Connect a serial cable to your controller and computer.

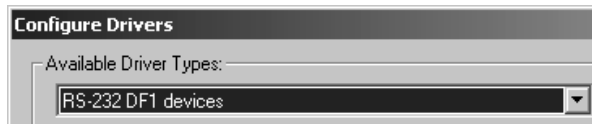


2. Configure an RS-232 driver:



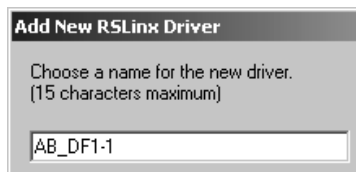
a. Start RSLinx software.

b. Click .

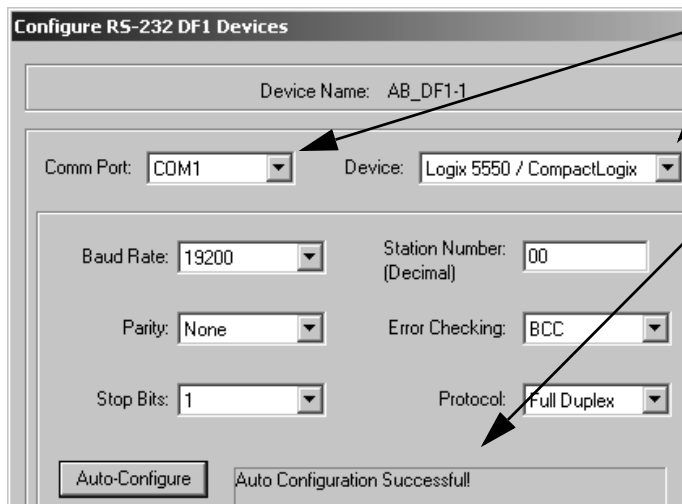


c. Select *RS-232 DF1* devices and choose





d. Accept the default name.

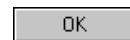


e. Select the COM port of your computer.

f. Select *Logix 5550/CompactLogix*.

g. Choose 

h. When the auto-configuration completes, choose



Configured Drivers:	
Name and Description	Status
AB_DF1-1 DF1 Sta: 0 COM1: RUNNING	Running
AB_ETH-1 A-B Ethernet RUNNING	Running
AB_ETH-2 A-B Ethernet RUNNING	Running

← The driver is successfully configured and running.

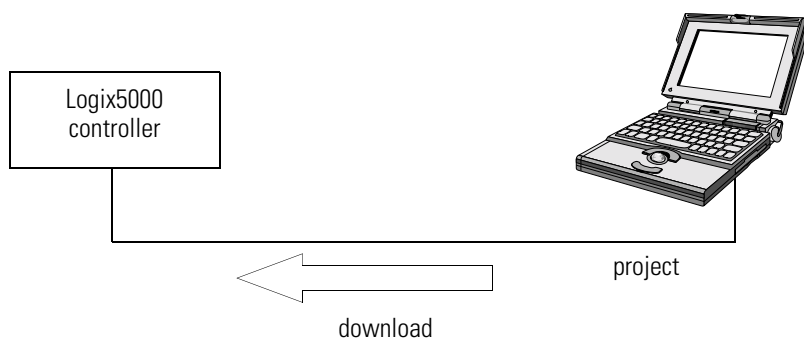
Download a Project to the Controller

To execute a project in a controller, download the project to the controller.

ATTENTION



When you download a project or update firmware, all active servo axes are turned off. Before you download a project or update firmware, make sure that this *will not* cause any unexpected movement of an axis.



download – transfer a project from your computer to the controller so you can run the project.

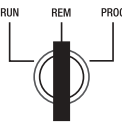
- When you download a project, you lose the project and data that is currently in the controller, if any.
- If the revision of the controller does not match the revision of the project, you are prompted to update the firmware of the controller. RSLogix 5000 software lets you update the firmware of the controller as part of the download sequence.

IMPORTANT

To update the firmware of a controller, first install a firmware upgrade kit.

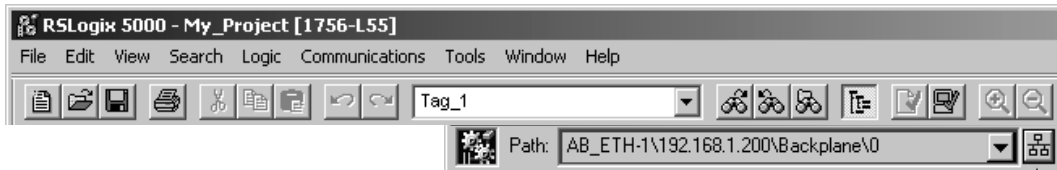
- An upgrade kit ships on a supplemental CD along with RSLogix 5000 software.
- To download an upgrade kit, go to www.ab.com. Choose *Product Support*. Choose *Firmware Updates*.

1. Turn the keyswitch of the controller to:



2. Define the path to the controller:

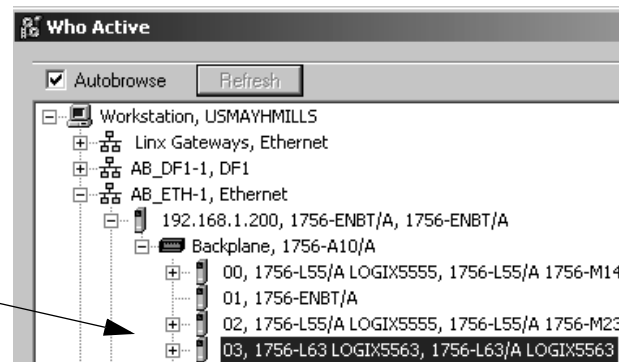
- a. Open the RSLogix 5000 project that you want to download.



- b. Click

- c. Browse to the controller.

- To open a level, click the + sign.
- When you see the controller, select it.



3. Download the project:

- a. Choose

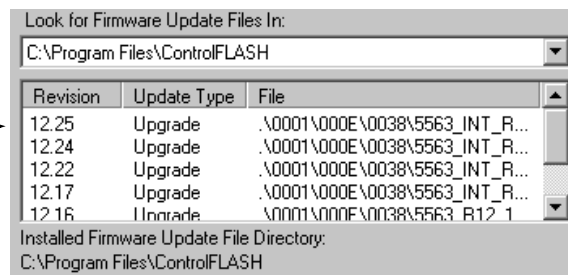
Which response did RSLogix 5000 software give?

Download to the controller.

- b. Choose

Failed to download to the controller. The revision of the offline project and controller's firmware are not compatible.

- b. Choose



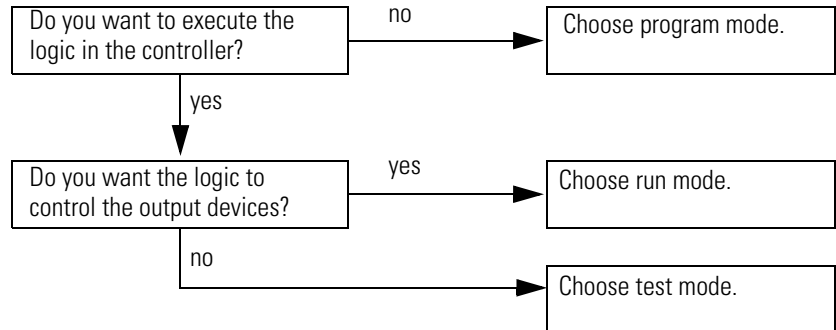
- c. Choose the revision for the controller.

- d. Choose and then

Select the Operating Mode of the Controller

To execute or stop executing the logic in a controller, change the operating mode of the controller.

1. Determine which mode you want for the controller:

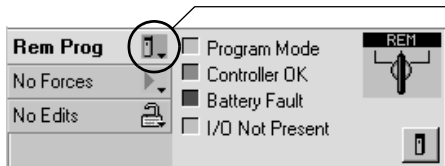


2. Turn the keyswitch to **RUN REM PROG**



3. Go online with the controller.

4. Select the mode.



Notes:

Organize a Project

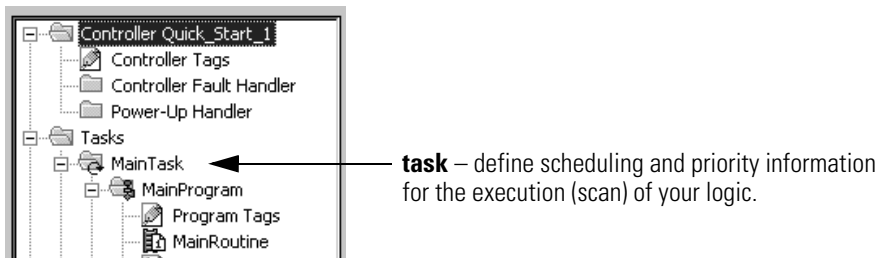
Using This Chapter

This chapter provides more detailed information on how to organize the program lay-out and data structures for the controller:

If you want to:	See page:
Configure the Task Execution	2-2
Create Additional Programs	2-3
Create User-Defined Data Types	2-5
Define Your Routines	2-7
Assign Main Routines	2-10
Configure the Controller	2-11
Configure I/O Modules	2-12

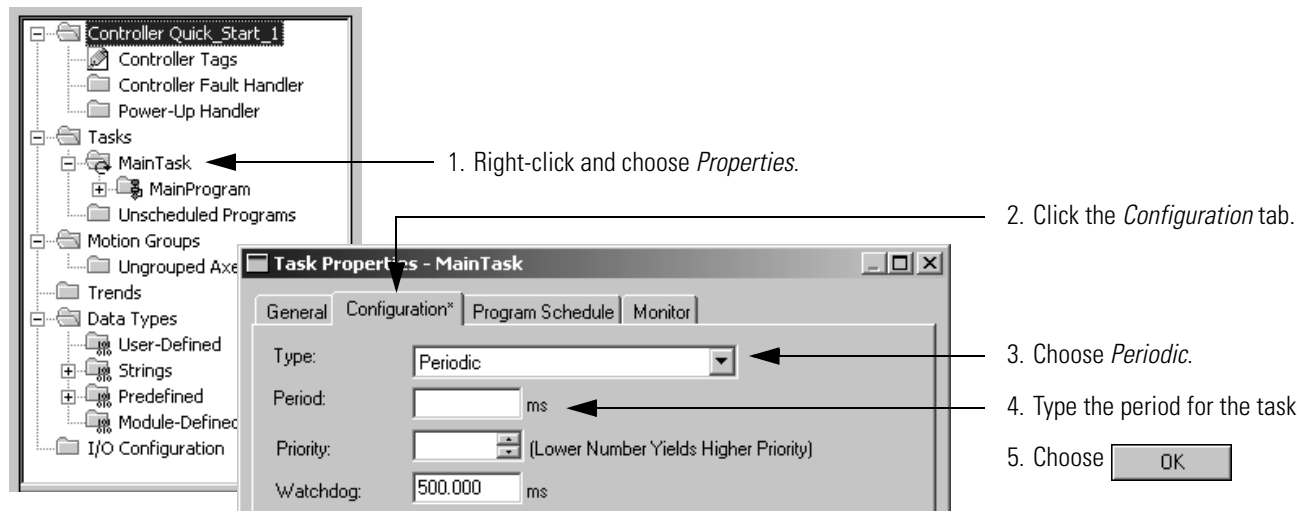
Configure the Task Execution

A new project contains a default task for the execution of your logic.



In this quick start, we limit the project to a single task with one of the following types of execution:

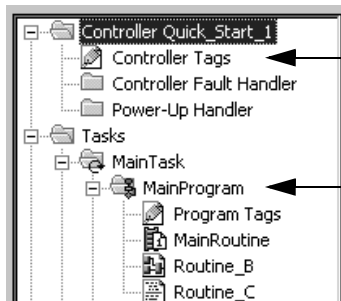
If you want to execute your logic:	Then configure the task for this type of execution:
<p>all of the time</p> <p>execution of logic</p>	<p>continuous</p> <p>This is the default configuration of <i>MainTask</i>.</p>
<p>at a specific period</p> <p>execution of logic</p>	<p>periodic</p> <p>You define the period at which the task executes.</p>



To use multiple tasks or execute a task when a specific event (trigger) occurs, see *Logix5000 Controllers Common Procedures*, publication 1756-PM001.

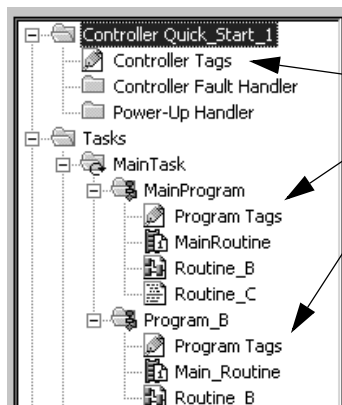
Create Additional Programs

A Logix5000 controller lets you divide your application into multiple programs, each with its own tags (data).



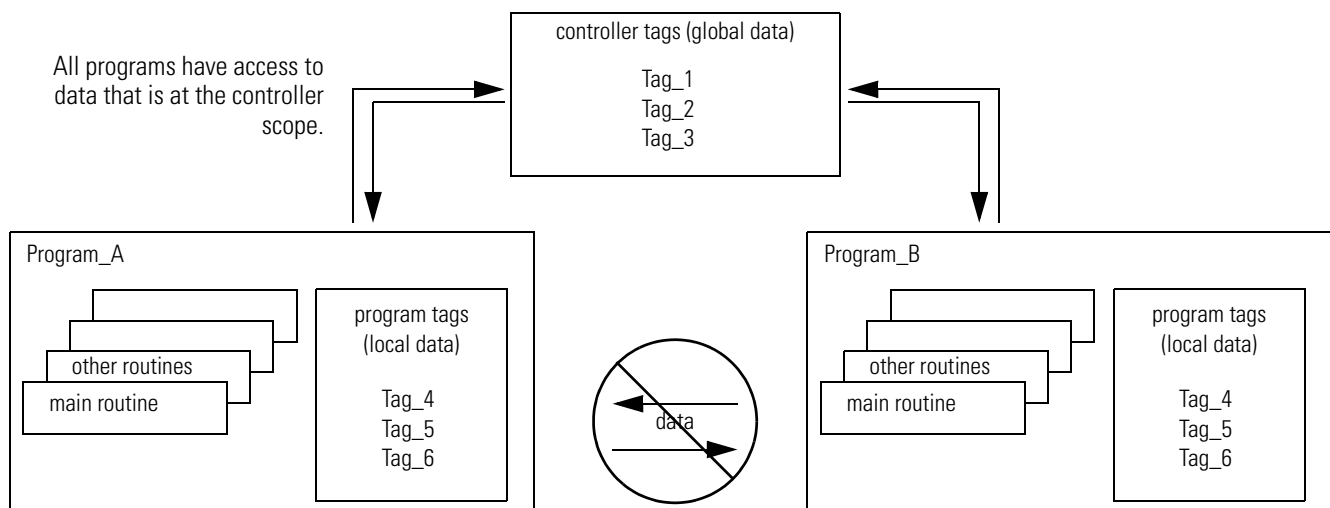
tag – store data. There is no fixed data table or numeric format for data addresses. The tag name is the address (no cross-reference to a physical address). You create the tags that you want to use.

program – isolate logic and data from other logic and data. Each program contains one or more logic routines as associated data.



scope – define whether a tag is accessible to all programs (controller tag) or limited to a specific program (program tag). Data at the program scope is isolated from other programs.

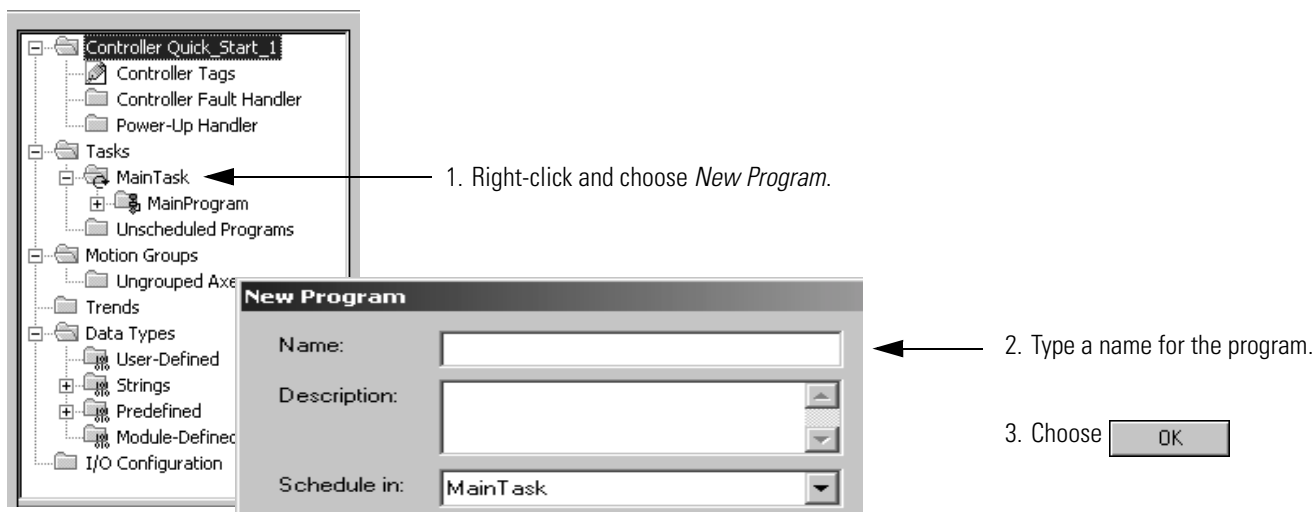
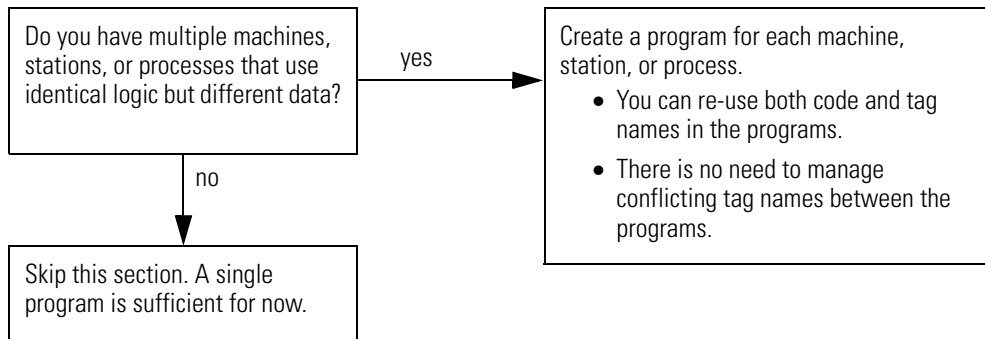
There is no need to manage conflicting tag names between the programs.



Data at the program scope is isolated from other programs:

- Routines cannot access data that is at the program scope of another program.
- You can re-use the tag name of a program-scoped tag in multiple programs.

For example, both Program_A and Program_B can have a program tag named Tag_4.



TIP

Names:

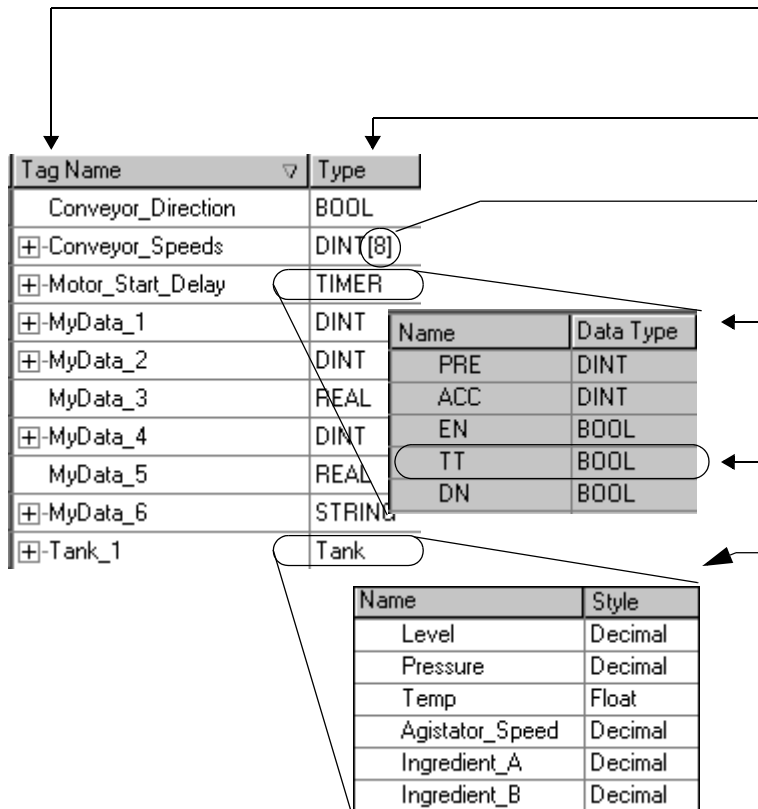
- only letters, numbers, and underscores (_)
- must start with a letter or an underscore
- ≤ 40 characters
- no consecutive or trailing underscores
- *not* case sensitive

Certain tags must be controller scope.

If you want to use a tag:	Then use this scope:
in more than one program in the project	Controller Tags
in a Message (MSG) instruction	
to produce or consume data	
to communicate with a PanelView terminal	Program Tags for the program
in a single program only	

Create User-Defined Data Types

User-defined data types let you organize your data to match your machine or process. This streamlines program development and creates self-documenting code that is easier to maintain.



tag – store data. There is no fixed data table or numeric format for data addresses. The tag name is the address. You create the tags that you want to use.

data type – define the type of data that a tag stores, such as a bit, integer, floating-point value, string, etc.

array – define a block of data (file). The entire block uses the same data type. It can have 1, 2, or 3 dimensions.

structure – combine a group of data types into a re-usable format (template for tags). Use a structure as the basis for multiple tags with the same data lay-out.

member – describe an individual piece of data within a structure

user-defined data type – create your own structure that emulates your devices. A user-defined data type stores all the data related to a specific aspect of your system. This keeps related data together and easy to locate, regardless of its data type.

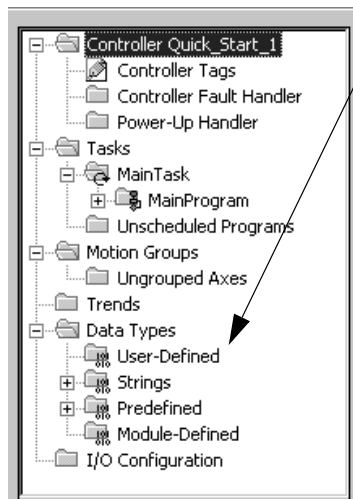
As you create user-defined data types, follow these guidelines:

Guideline:	Details:
1. Consider the pass-through of descriptions.	See <i>Describe a User-Defined Data Type</i> on page 4-2.
2. Data that represents an I/O device requires additional programming.	If you include members that represent I/O devices, you must use logic to copy the data between the members in the user-defined data type and the corresponding I/O tags.
3. If you include an array as a member, limit the array to a single dimension.	Multi-dimension arrays are <i>not</i> permitted in a user-defined data type.
4. When you use the BOOL, SINT, or INT data types, place members that use the same data type in sequence:	Logix5000 controllers allocate memory in 4-byte chunks. If you sequence smaller data types together, the controller packs as many as it can fit into a 4-byte chunk.

more efficient	less efficient
BOOL	BOOL
BOOL	DINT
BOOL	BOOL
DINT	DINT
DINT	BOOL

To create a user-defined data type and tags that use the data type:

1. Create a user-defined data type:



a. Right-click and choose *New Data Type*.

b. Type a name for the data type (*not* the name of a tag that will use the data type).

c. Enter the members.
As an option, type a description for each member.

d. Choose

Data Type: Tank

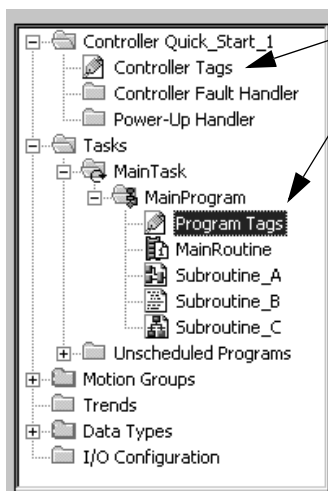
Name:

Description:

Members:

Name	Data Type
Level	DINT
Pressure	DINT
Temp	REAL
Agistator_Speed	DINT
Ingredient_A	BOOL
Ingredient_B	BOOL

2. Create a tag that uses the user-defined data type:



a. Right-click the scope that you want for the tag and choose *Edit Tags*.

b. Type a name for the tag

c. Type the name of the user-defined data type from step 1.

Tag Name	Alias For	Base Tag	Type
MyData_4			DINT
MyData_5			REAL
MyData_6			STRING
Tank_1			Tank ...

3. If you want the tag to be an array (multiple instances of the data type):

c. Select the data type and click

d. Specify the array dimensions.

e. Choose

Data Types:

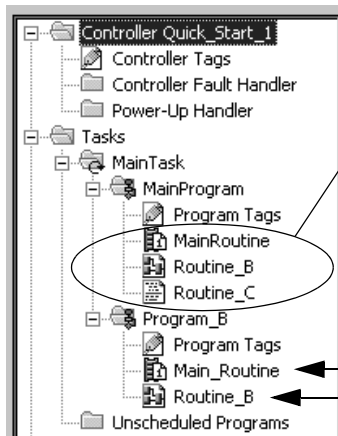
SELECTABLE_NEGATE
SELECTED_SUMMER
SERIAL_PORT_CONTROL
SFC_ACTION
SFC_STEP
SFC_STOP
SINT
SPLIT_RANGE
STRING
Tank
TIMER

Array Dimensions:

Dim 0	Dim 1	Dim 2
<input type="text" value="4"/>	<input type="text" value="3"/>	<input type="text" value="2"/>

Define Your Routines

Once your project has the required programs, you have to define and create the routines for each program.



routine – provide the executable code (logic) for a program (similar to a program file in a PLC or SLC controller).

main routine – For each program, you assign a main routine.

- When the program executes, its main routine automatically executes.
- Use the main routine to control the execution of the other routines in the program.
- To call (execute) another routine (subroutine) within the program, use a Jump to Subroutine (JSR) instruction.

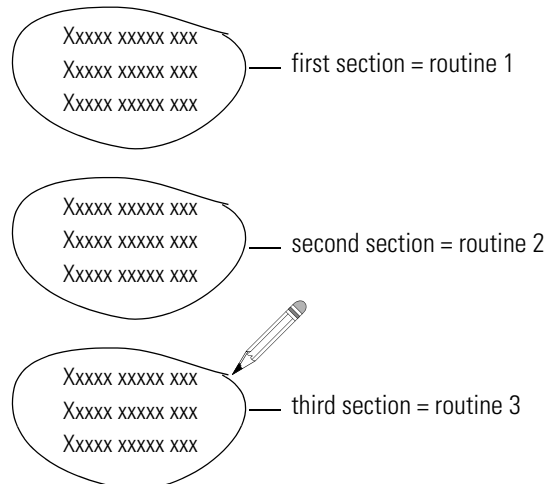
subroutine – Any routine other than the main routine or fault routine. To execute a subroutine, use a Jump to Subroutine (JSR) instruction in another routine, such as the main routine.

Define a Routine for Each Section of Your Machine or Process

To make your project easier to develop, test, and troubleshoot, divide it into routines (subroutines):

1. Identify each physical section of your machine or process.
2. Assign a routine for each of those sections.

Description of Your Machine or Process



Identify the Programming Languages That Are Installed

To determine which programming languages are installed on your version of RSLogix 5000 software:

1. Start RSLogix 5000 software.
2. From the *Help* menu, choose *About RSLogix 5000*.

To add a programming language, see *ControlLogix Selection Guide*, publication 1756-SG001.

Assign a Programming Language to Each Routine

For each routine, choose a programming language:

- Logix5000 controllers let you use the following languages:
 - ladder logic
 - function block diagram
 - sequential function chart
 - structured text
- Use any combination of the languages in the same project.

In general, if a routine represents:	Then use this language:
continuous or parallel execution of multiple operations (not sequenced)	ladder logic
boolean or bit-based operations	
complex logical operations	
message and communication processing	
machine interlocking	
operations that service or maintenance personnel may have to interpret in order to troubleshoot the machine or process.	function block diagram (FBD)
continuous process and drive control	
loop control	
calculations in circuit flow	
high-level management of multiple operations	sequential function chart (SFC)
repetitive sequences of operations	
batch process	
motion control using structured text	
state machine operations	
complex mathematical operations	structured text
specialized array or table loop processing	
ASCII string handling or protocol processing	

Divide Each Routine Into More Meaningful Increments

If a routine uses this language:	Then:	Example:
ladder logic structured text	Break up large routines into several smaller routines	To continuously execute several complex boolean operations... ...create a separate routine for each operation.
function block diagram (FBD)	Within the FBD routine, make a sheet for each functional loop for a device (motor, valve, etc.).	To control 4 valves, where each valve requires feedback that it is in its commanded position... ...make a separate sheet for each valve.
sequential function chart (SFC)	Break the SFC into steps.	To perform the following sequence: 1. Fill a tank. 2. Mix the ingredients in the tank. 3. Empty the tank... ...make each section (fill, mix, empty) a separate step.

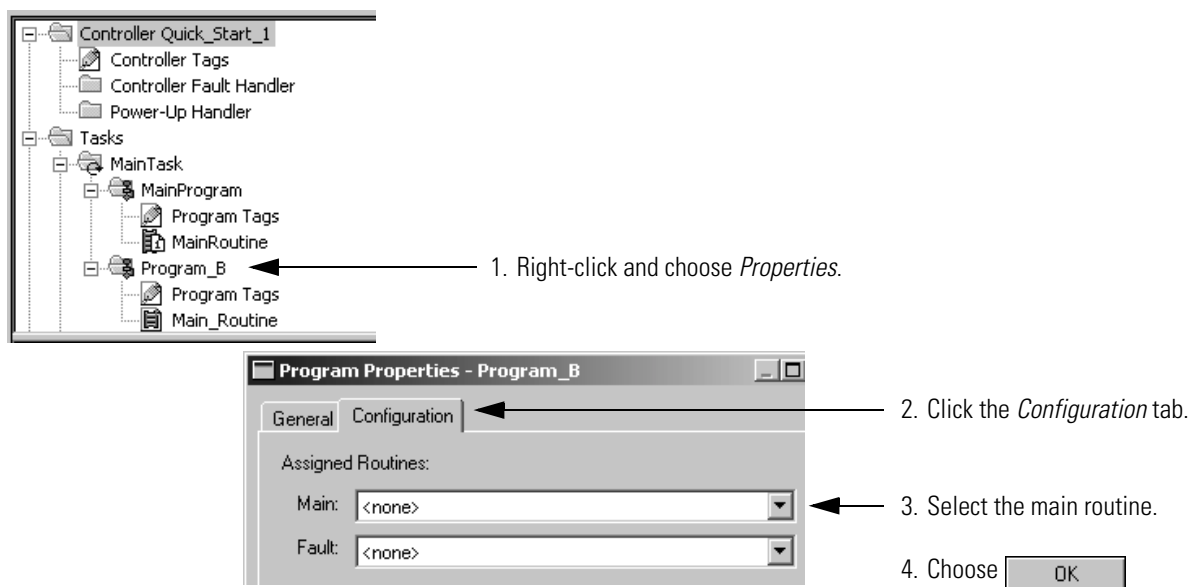
Assign Main Routines

Each program requires a main routine. Once you create your routines, assign a main routine for each program.

IMPORTANT

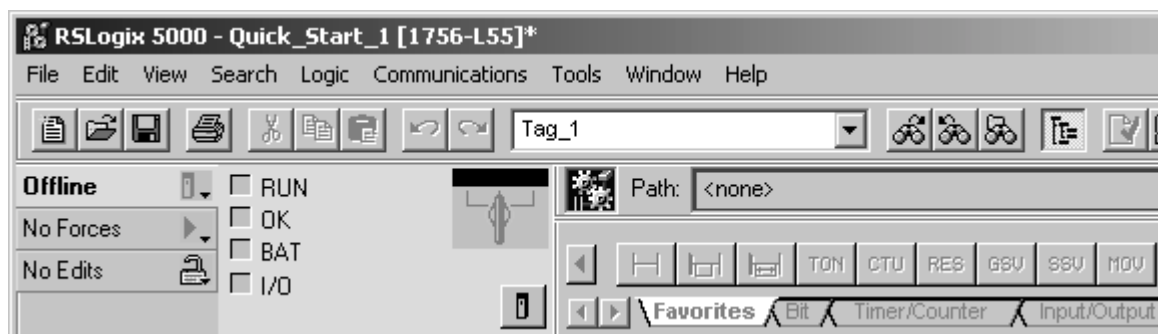
In the default project, *MainProgram* already has a main routine (*MainRoutine*). You have to assign a main routine only for each additional program that you create.

To assign a main routine:



Configure the Controller

If you want to change the configuration of the controller, such as name, chassis size, or slot number, use the Controller Properties dialog box.



1. Click the Controller Properties button.

Minor Faults	Date/Time	Advanced	SFC Execution	File	Nonvolatile Memory
General*					
Serial Port		System Protocol		User Protocol	
Major Faults					
Vendor:	Allen-Bradley				
Type:	1756-L55 ControlLogix5555 Controller				
Revision:	12.1				
Name:	<input type="text"/>				
Description:	<input type="text"/>				
Chassis Type:	1756-A10 10-Slot ControlLogix Chassis				
Slot:	0				

2. Change the required properties (some items apply to only certain controllers).

e. type of controller

f. name of the controller

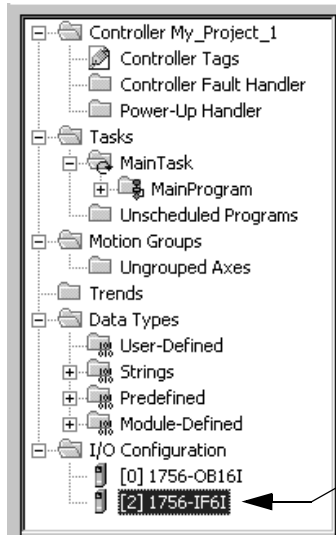
g. chassis size for the controller

h. slot number of the controller

3. Choose

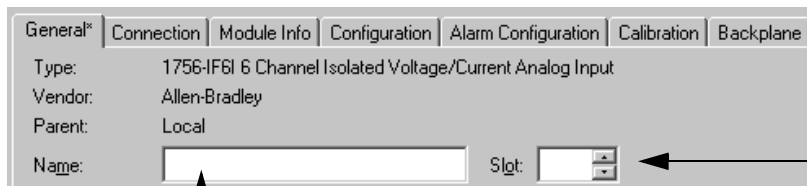
Configure I/O Modules

To change the behavior of a module, use the Module Properties window for the module. The configuration options vary from module to module.



1. Right-click the module and choose *Properties*.

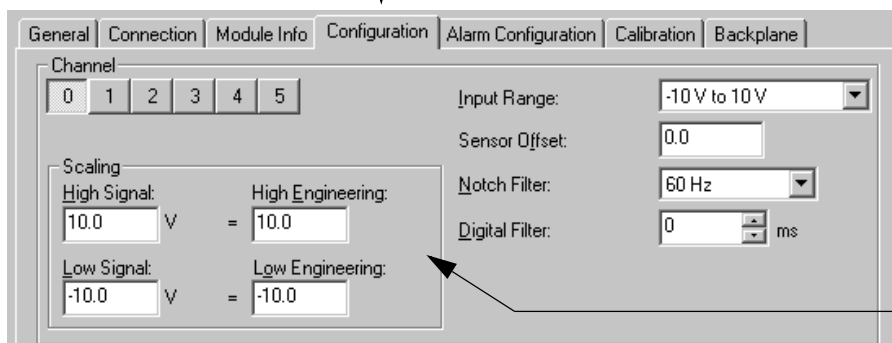
2. To change the name or slot number, use the *General* tab.



location of the module in the chassis or rail

name of the module

3. To change the configuration, click the *Configuration* tab. Some modules have several configuration tabs.



range

scaling

Program a Project Offline

Using This Chapter

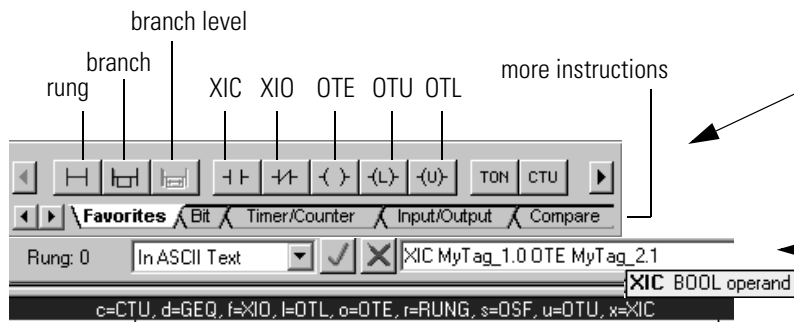
This chapter provides more detailed information on how to program the logic for a routine and create tags for the logic.

If you want to:	See page:
Enter Ladder Logic	3-2
Export/Import Ladder Logic	3-6
Enter a Function Block Diagram	3-9
Use a Faceplate for a Function Block	3-12
Enter Structured Text	3-14
Enter a Sequential Function Chart	3-16
Assign Operands	3-18
Verify a Project	3-20
Review Guidelines for Tags	3-22

In this chapter, you program the project while offline. Online programming requires additional steps. See chapter 6, “Program a Project Online”.

Enter Ladder Logic

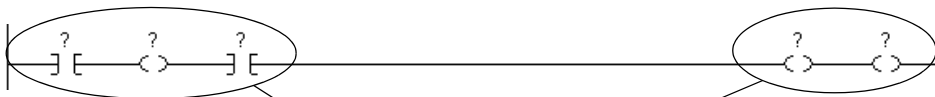
To enter ladder logic, you have the following options:



drag and drop logic elements – Use the Language Element toolbar to drag and drop a rung, branch, or instruction to your routine.

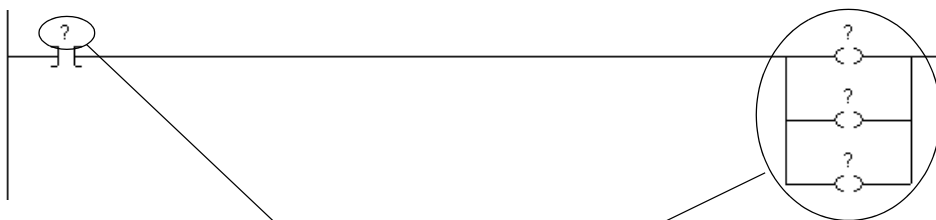
ASCII text – Use ASCII text to enter or edit logic. A tool tip helps you enter the required operands. ASCII text typically uses the following format:
mnemonic operand_1 operand_2

quick keys – Assign a logic element (rung, branch, instruction) to a keyboard key. To add an element to the right or below the cursor, press the designated key for the element.



outputs in series – Place multiple output instructions in sequence (serial) on a rung.

interlace input and output instructions – The last instruction on the rung must be an output instruction.

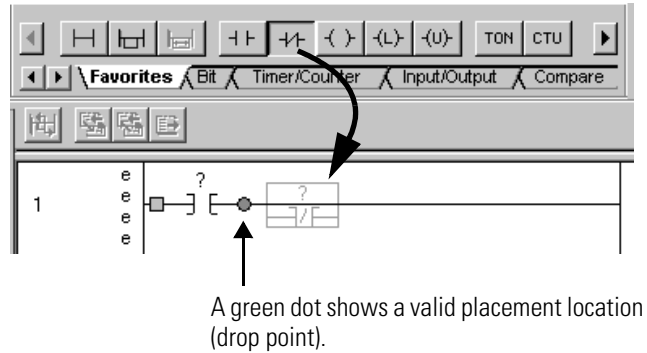


parallel branches – No limit to the number of parallel branches on a rung (nest up to 6 levels).

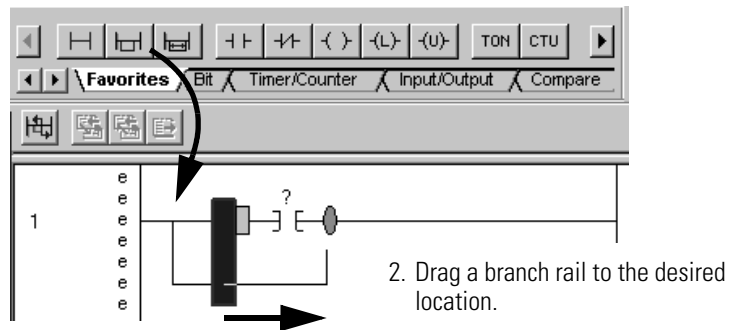
leave operands undefined – enter logic without defining operands. RSLogix 5000 software lets you enter and save logic without assigning operands. This lets you develop your logic in iterations and save libraries of code for re-use.

Drag and Drop an Element

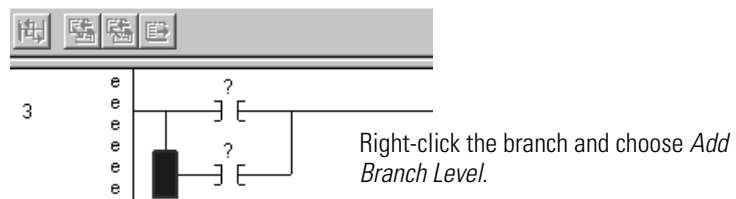
To:	Do this:
add a rung	Drag the button for the rung or instruction directly to the desired location.
add an instruction	



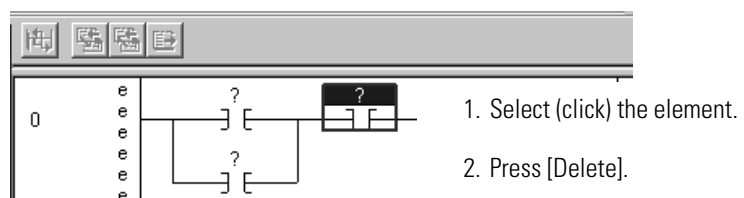
add a branch	1. Drag the branch button to where the branch starts. A green dot shows a valid placement location (drop point).
--------------	--



add a level to a branch

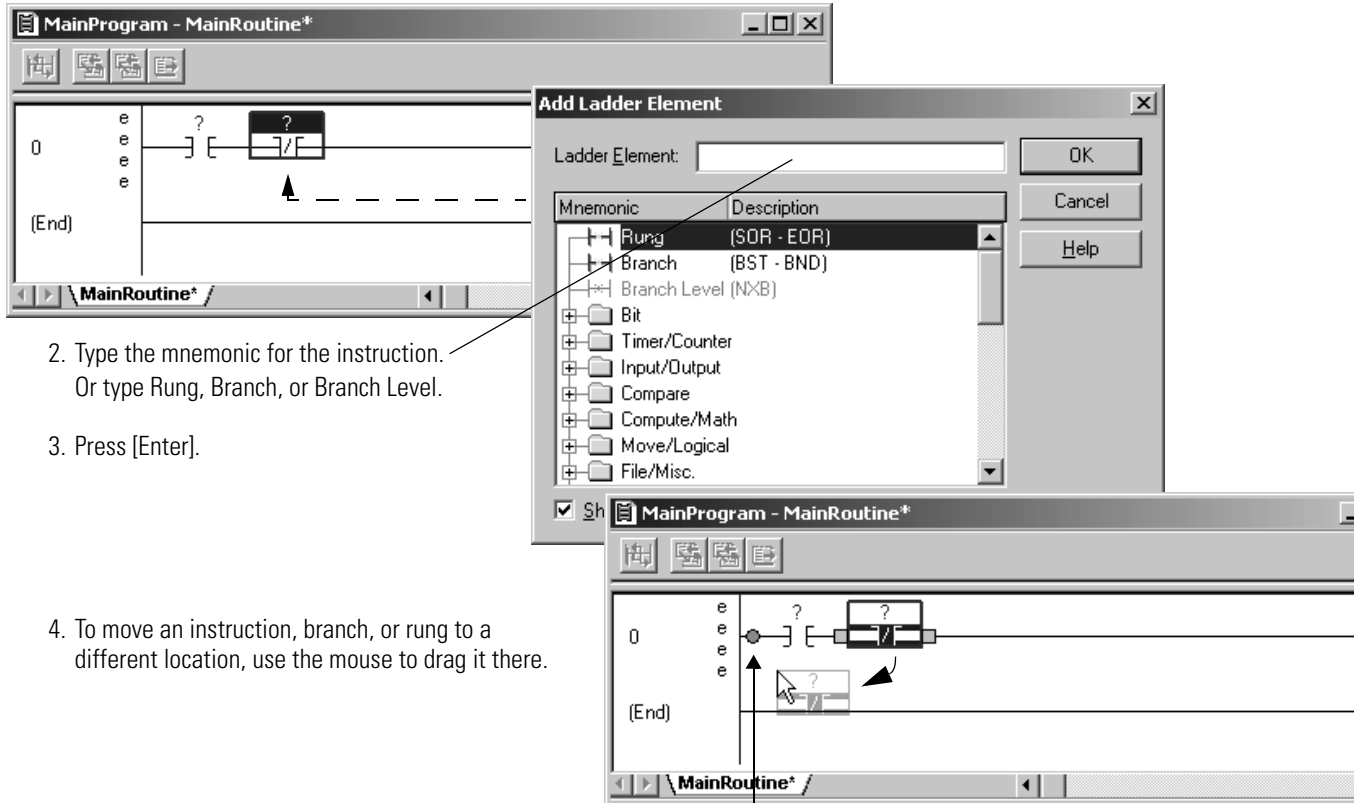


delete an element



Use the Keyboard to Add an Element

1. Press [Insert].



2. Type the mnemonic for the instruction.
Or type Rung, Branch, or Branch Level.
3. Press [Enter].
4. To move an instruction, branch, or rung to a different location, use the mouse to drag it there.

A green dot shows a valid placement location (drop point).

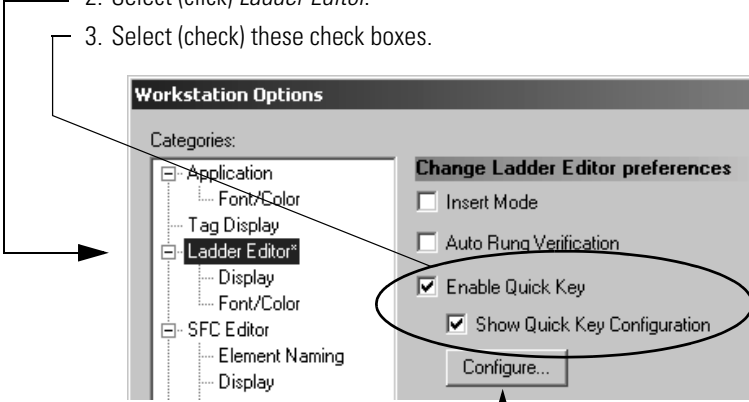


Enter Logic Using ASCII Text

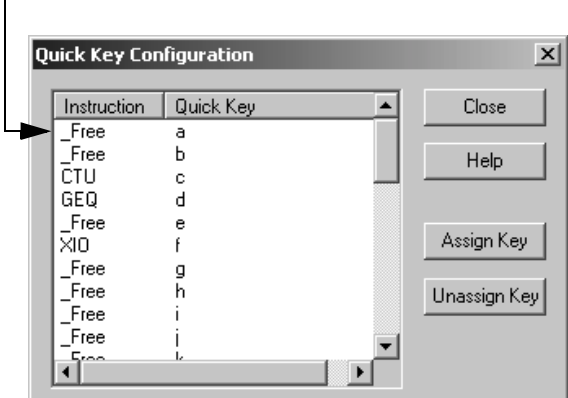
1. Double click the rung.



2. Enter the ASCII text for the rung.

Enable Quick Keys

1. Choose *Tools* ⇒ *Options*.
2. Select (click) *Ladder Editor*.
3. Select (check) these check boxes.
 
4. To assign a key to an element:
 - a. Choose .
 - b. For the desired key, select the element.
 - c. When you have assigned the desired keys, choose .



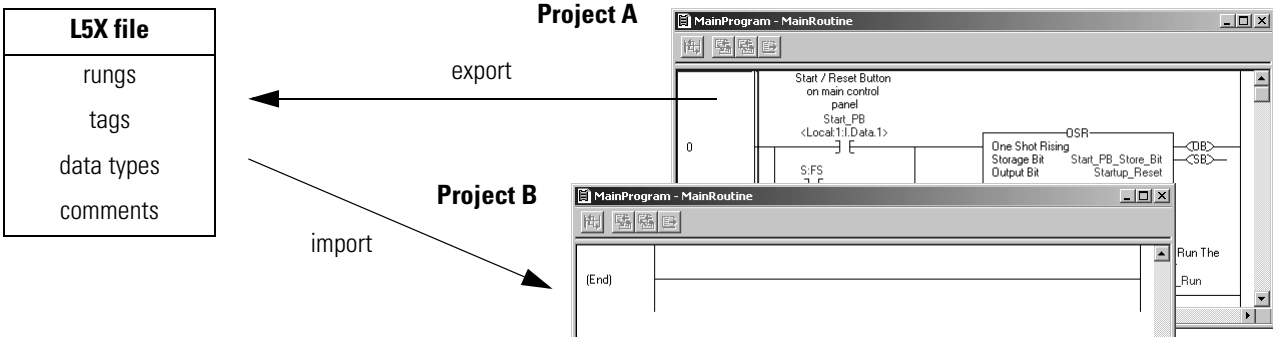
Instruction	Quick Key
_Free	a
_Free	b
CTU	c
GEQ	d
_Free	e
XIO	f
_Free	g
_Free	h
_Free	i
_Free	j
_Free	k

Export/Import Ladder Logic



RSLogix 5000 software
13.0 or later

If you want to re-use ladder logic from another project, simply export the logic to an L5X file and import it into the required project. The L5X file contains all that you need for the logic except I/O modules.



When You Import Rungs...

When you import rungs, RSLogix 5000 software shows a list of the tags and user-defined data types that go along with the rungs. Use the list to manage the tags and data types that are created during the import operation.

The *Operation* column shows what will happen to each tag and data type during the import. The software either creates it, uses an existing one in the project, or discards it (does not import it).

If desired, you can rename a tag to make it fit the project better.

If a tag already exists in the project, you can either:

- Use the existing tag, which discards the tag in the library file and binds the logic to the existing tag.
- Rename the tag, which creates a new one.

No new I/O tags are created.

If you place the variables for the rungs in a user-defined data type, you have less tags to manage.

Import Configuration					
Tags					
Data Types					
	Tag Name	Alias For	Type	Description	Operation
	CN2		Conveyor_Type	Conveyor CN1	Create New
	CN2_M	Local:2:0.Data.0		Conveyor CN1 Motor	Create New
	Estop_Disabled		BOOL	No Estop pressed	Use Existing
	Local:1:I		AB:1756_DI:1:0		Discard
	Local:2:O		AB:1756_DO:0:0		Use Existing

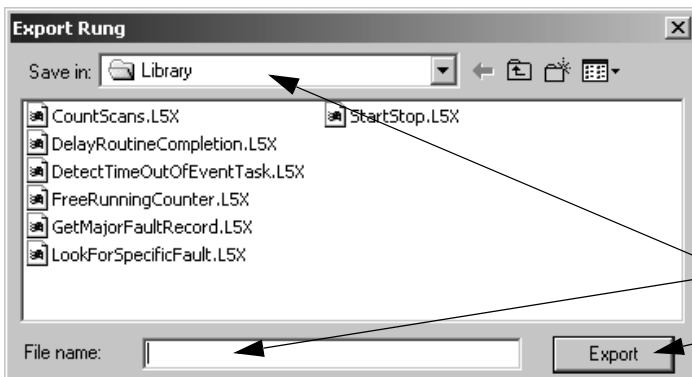
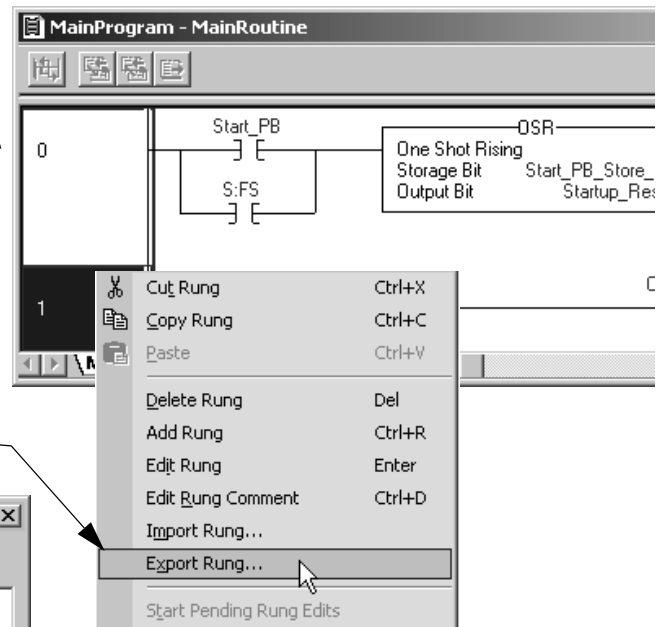
If an I/O tag already exists in the project, the import operation uses this tag for any aliases to that tag name. Once you import a project, make sure you check the alias tags for accuracy.

Export Rungs

1. Select the rungs to export:

If rungs are:	Do this:
in sequence	Click the first rung and then [Shift] + click the last rung.
out of sequence	Click the first rung and then [Ctrl] + click each additional rung.

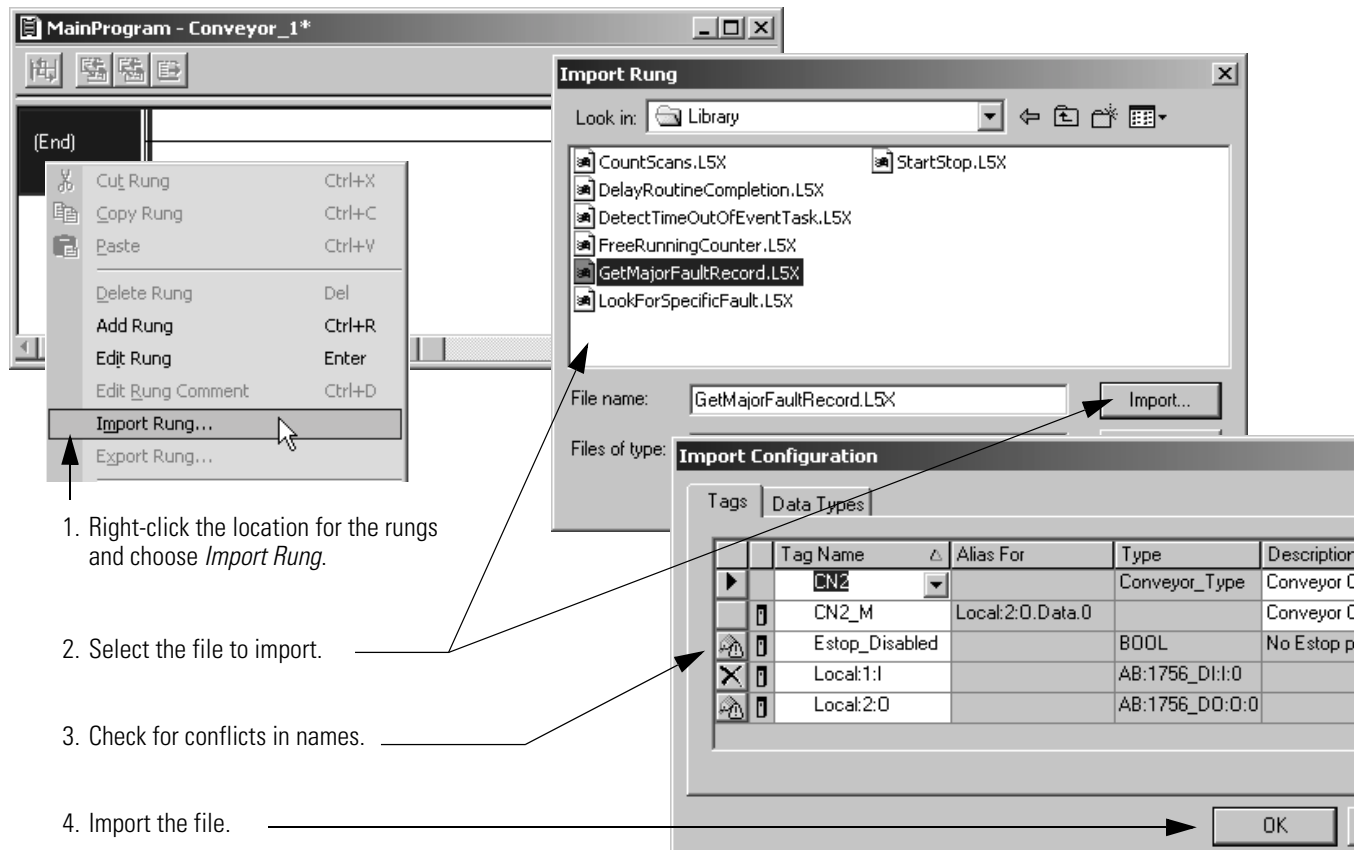
2. Right-click the selection and choose *Export Rung*.



3. Choose a location and name for the file.

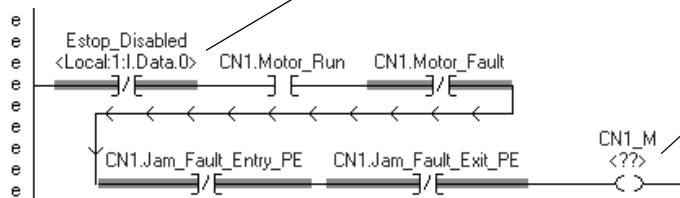
4. Create the file.

Import Rungs



Check Alias Tags

rungs that you imported

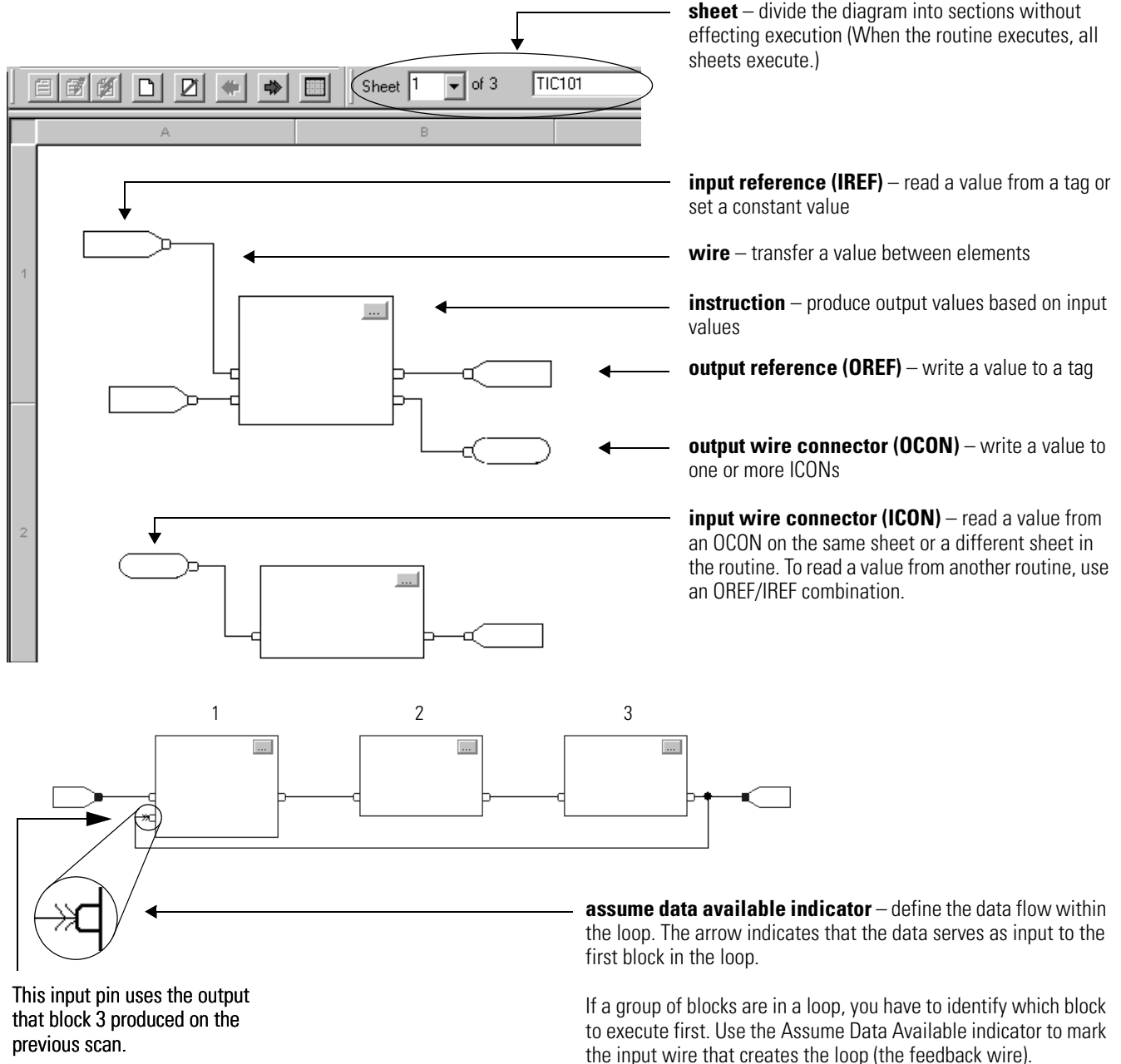


If you import an alias tag, make sure it points to the correct base tag. When a tag is an alias for a tag that already exists in the project, the software sets up the relationship between the alias and base tags.

If the project does not have the base tag, you have to either create the base tag or point the alias to a different base tag.

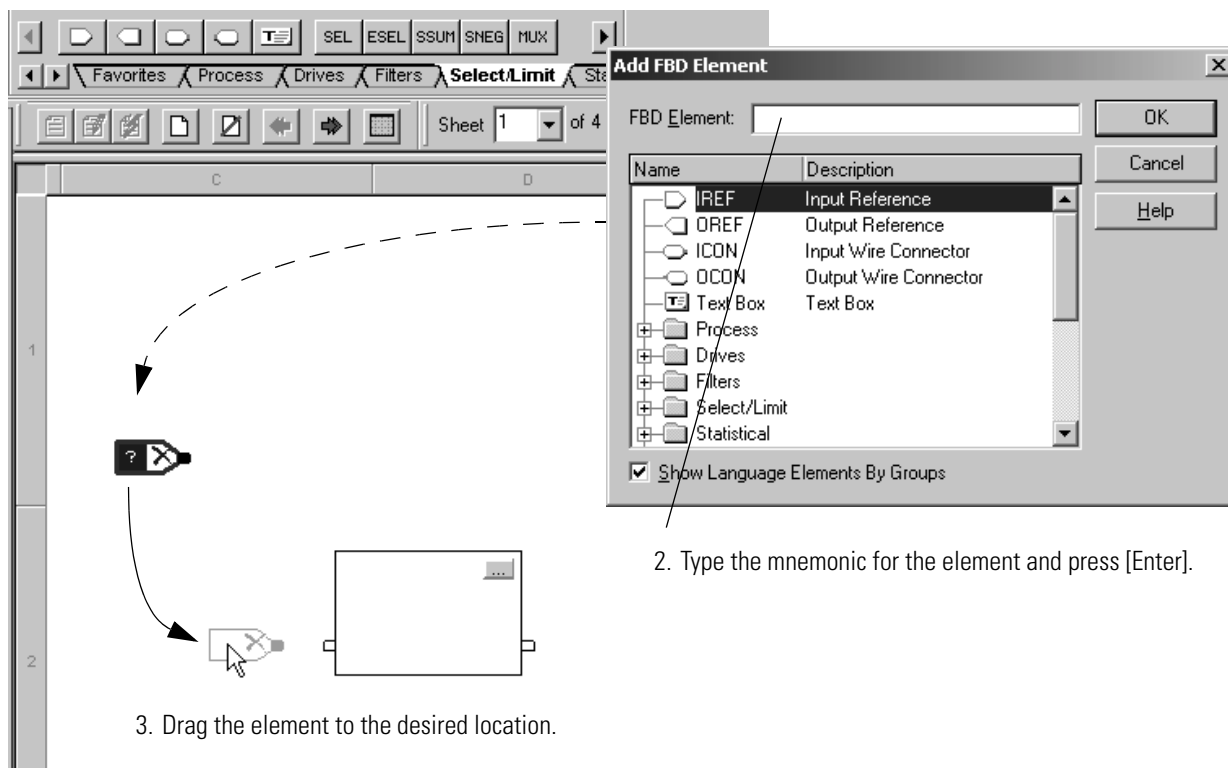
Enter a Function Block Diagram

A function block diagram lets you visually define the flow of data between instructions. The data flow then drives the execution order of the instructions.

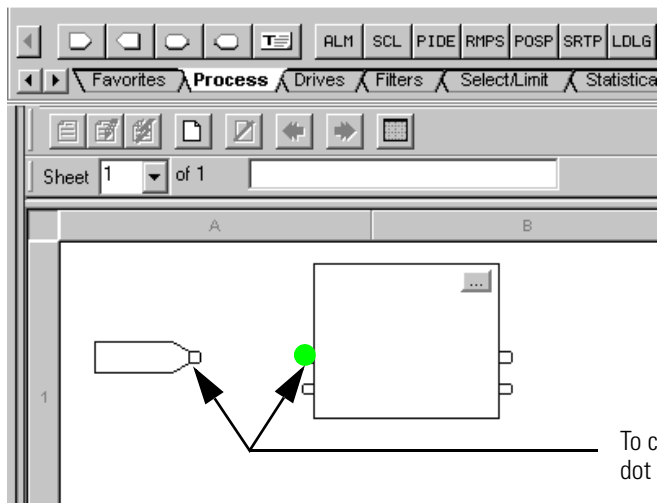


Use the Keyboard to Add an Element

1. Press [Insert].



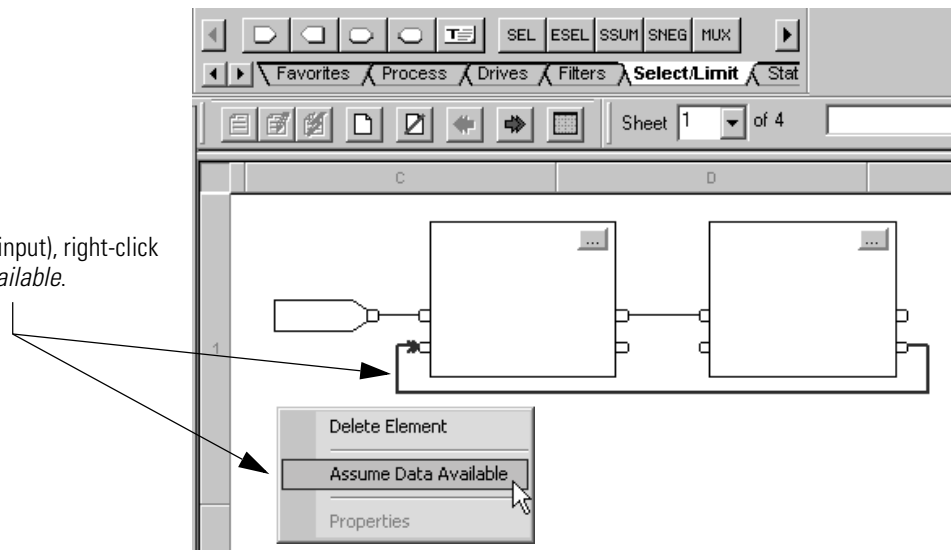
Connect Elements



To connect elements, click corresponding pins (green dot = valid connection point).

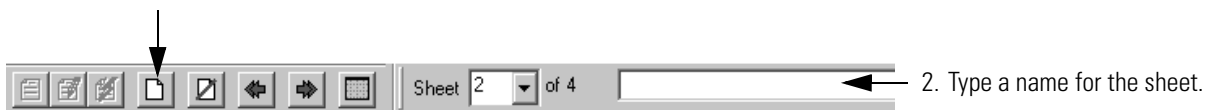
Resolve a Loop

To resolve a loop (define a wire as an input), right-click the wire and choose *Assume Data Available*.



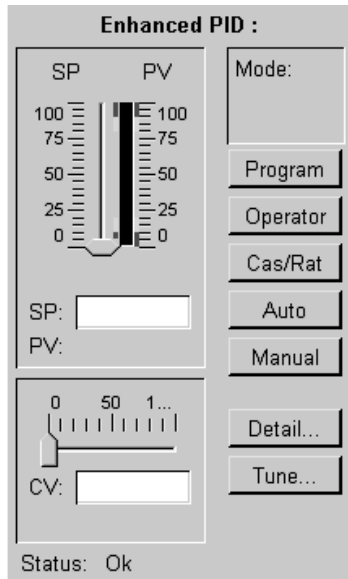
Add Sheet

1. Click the New Sheet button.



Use a Faceplate for a Function Block

RSLogix 5000 software includes faceplates (controls) for some of the function block instructions.



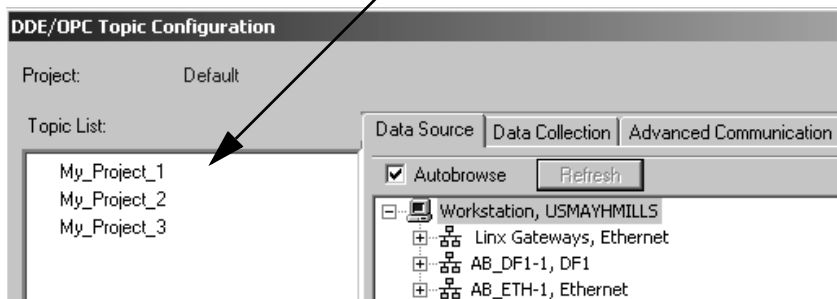
faceplate – Active-X control that lets you interact with a function block instruction.

- Your RSLogix 5000 Enterprise Series software package includes the faceplates but *does not* automatically install them. To use the faceplates, locate them on your software CD and install them separately.
 - Use faceplates in an Active-X container, such as the following software:
 - RSView®32™
 - RSView® SE
 - Microsoft® Excel
 - RSLogix 5000 software is *not* a valid Active-X container.
 - Faceplates communicate with the controller via DDE/OPC topics in RSLinx software. To use RSLinx software for DDE/OPC topics, purchase either:
 - RSLinx software as a separate package
 - RSLogix 5000 professional edition software, which includes RSLinx professional edition software
- RSLinx Lite software, which comes with the other RSLogix 5000 software packages, *does not* provide DDE/OPC communication.

Faceplates are available for the following instructions:

- Alarm (ALM)
- Enhanced Select (ESEL)
- Totalizer (TOT)
- Ramp/Soak (RMPS)
- Discrete 2-State Device (D2SD)
- Discrete 3-State Device (D3SD)
- Enhanced PID (PIDE)

topic – In RSLinx software, a topic represents a specific path to a controller.



RSLogix 5000 software, revision 10.0 or later, automatically creates an RSLinx topic whenever you:

- create a project
- save a project
- change the revision of a project to 10.0 or later

In some cases, you have to update the data source for the topic in RSLinx software.

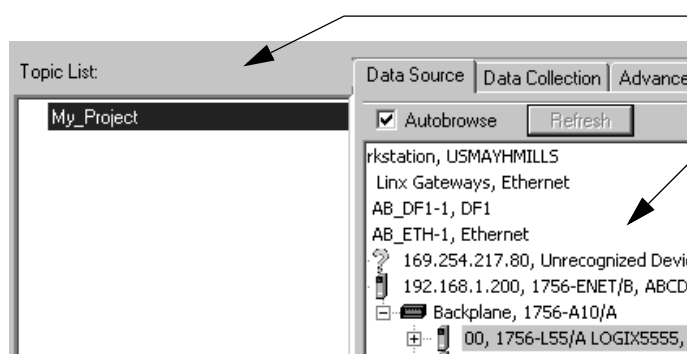
Set Up a Topic

1. Use RSLogix 5000 software to create the topic:



- Set the project path (communication route to the controller).
- Save the project.

2. In RSLinx software, check the topic:

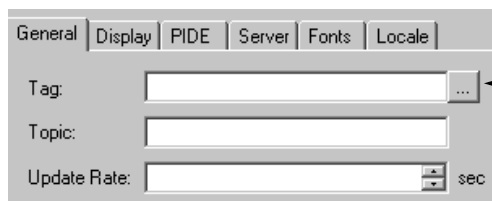


- choose *DDE/OPC* ⇒ *Topic Configuration*.
- Select your project.
- Make sure the data source points to your controller.
- Choose **Done**

Add a Faceplate to Microsoft Excel Software



- Start Microsoft Excel software.
- Choose *View* ⇒ *Toolbars* ⇒ *Control Toolbox*.
- Click and select the *Logix 5000...Faceplate Control* that you want.
- In the location for the faceplate, drag the pointer to the desired size of the faceplate.
- Right-click the faceplate and choose *Logix 5000...Faceplate Control Object* ⇒ *Properties*.



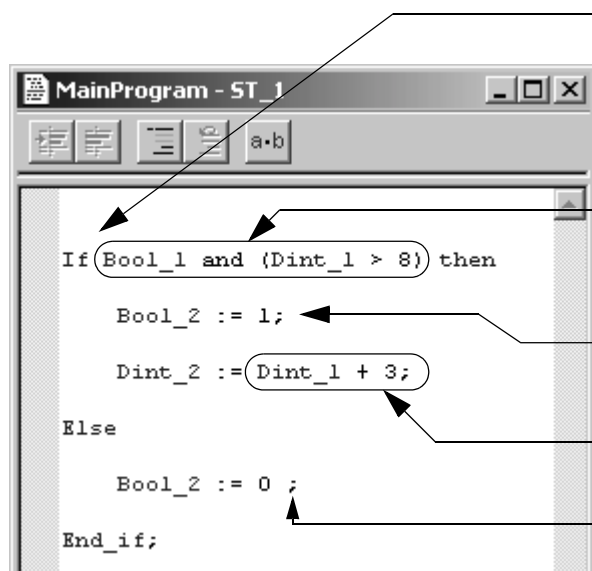
- Click and browse to the tag that the faceplate controls.
- Select the update period for the control.
- Choose **OK**



- To exit design mode and use the control, click here.

Enter Structured Text

Structured text is a textual programming language that uses statements to define what to execute. Structured text can contain these components:



construct – define logical conditions for the execution of other structured text code (i.e., other statements). In this example, the construct is If...Then...Else...End_if.

BOOL expression – check if a tag or equation is true or false. A BOOL expression typically serves as the condition for an action (the if, while, or until of a construct).

assignment – write a value to a tag. The value moves from the right side of the := to the left side.

numeric expression – calculate a value.

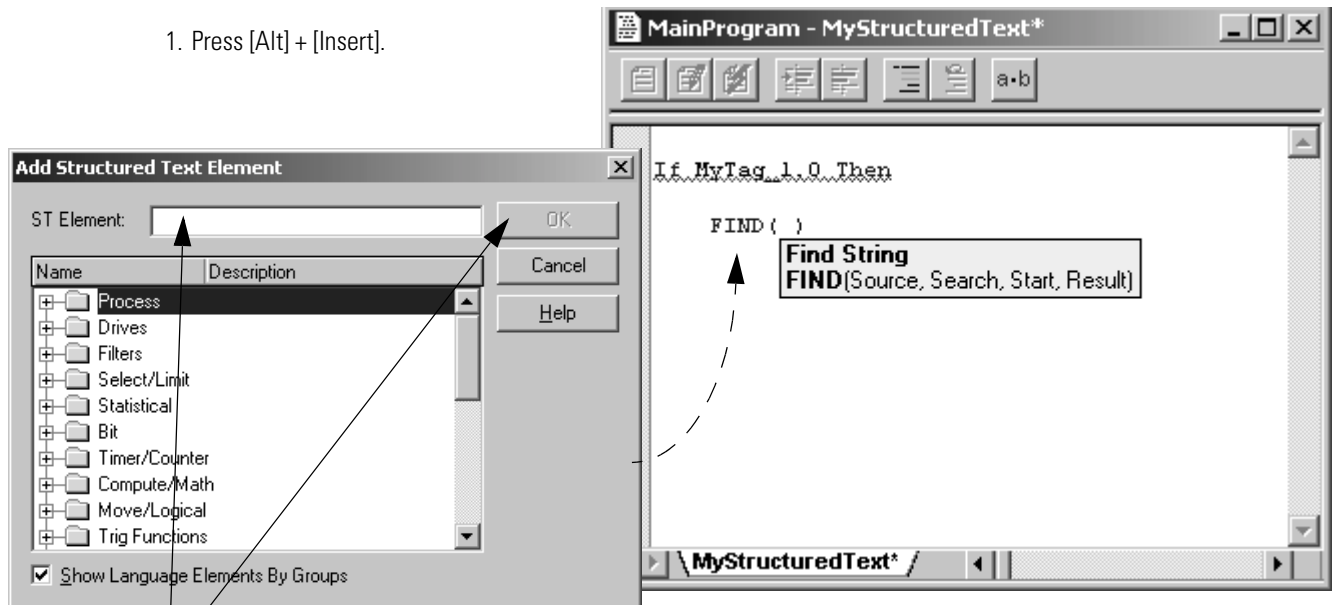
semi colon “;” – terminate an assignment, instruction, or end of a construct.

As you enter structured text, follow these guidelines:

Guideline:	Description:										
1. Structured text is <i>not</i> case sensitive.	Use any combination of upper-case and lower-case letters that makes your text easiest to read. For example, these three variations of “IF” are the same: IF, If, if.										
2. Use tabs, spaces, and carriage returns (separate lines) to make your structured text easier to read.	<p>Tabs, spaces, and carriage returns have no effect on the execution of the structured text.</p> <table> <tr> <th>This:</th><th>Executes the same as this:</th></tr> <tr> <td> <pre>If Bool1 then Bool2 := 1; End_if;</pre> </td><td> <pre>If Bool1 then Bool2 := 1; End_if;</pre> </td></tr> <tr> <td> <pre>Bool2 := 1;</pre> </td><td> <pre>Bool2:=1;</pre> </td></tr> </table>	This:	Executes the same as this:	<pre>If Bool1 then Bool2 := 1; End_if;</pre>	<pre>If Bool1 then Bool2 := 1; End_if;</pre>	<pre>Bool2 := 1;</pre>	<pre>Bool2:=1;</pre>				
This:	Executes the same as this:										
<pre>If Bool1 then Bool2 := 1; End_if;</pre>	<pre>If Bool1 then Bool2 := 1; End_if;</pre>										
<pre>Bool2 := 1;</pre>	<pre>Bool2:=1;</pre>										
3. Write BOOL expressions as either true or false	<p>Use a BOOL expression to determine if specific conditions are true (1) or false (0).</p> <ul style="list-style-type: none"> A BOOL tag is already true (1) or false (0). <i>Do not</i> use an “=” sign to check its state. <table> <tr> <th>This is OK:</th><th>This is NOT OK:</th></tr> <tr> <td>If Bool1 ...</td><td>If Bool1 = 1 ...</td></tr> <tr> <td>If Not (Bool2) ...</td><td>If Bool2 = 0 ...</td></tr> </table> <ul style="list-style-type: none"> To check an integer, REAL, or string, make a comparison (=, <, <=, >, >=, <>). <table> <tr> <th>This is OK:</th><th>This is NOT OK:</th></tr> <tr> <td>If Dint1 > 5 ...</td><td>If Dint1 ...</td></tr> </table>	This is OK:	This is NOT OK:	If Bool1 ...	If Bool1 = 1 ...	If Not (Bool2) ...	If Bool2 = 0 ...	This is OK:	This is NOT OK:	If Dint1 > 5 ...	If Dint1 ...
This is OK:	This is NOT OK:										
If Bool1 ...	If Bool1 = 1 ...										
If Not (Bool2) ...	If Bool2 = 0 ...										
This is OK:	This is NOT OK:										
If Dint1 > 5 ...	If Dint1 ...										
4. For an assignment, start with the destination.	<p>Write an assignment as follows:</p> <pre>Destination := Source;</pre> <p style="text-align: center;">← data</p>										

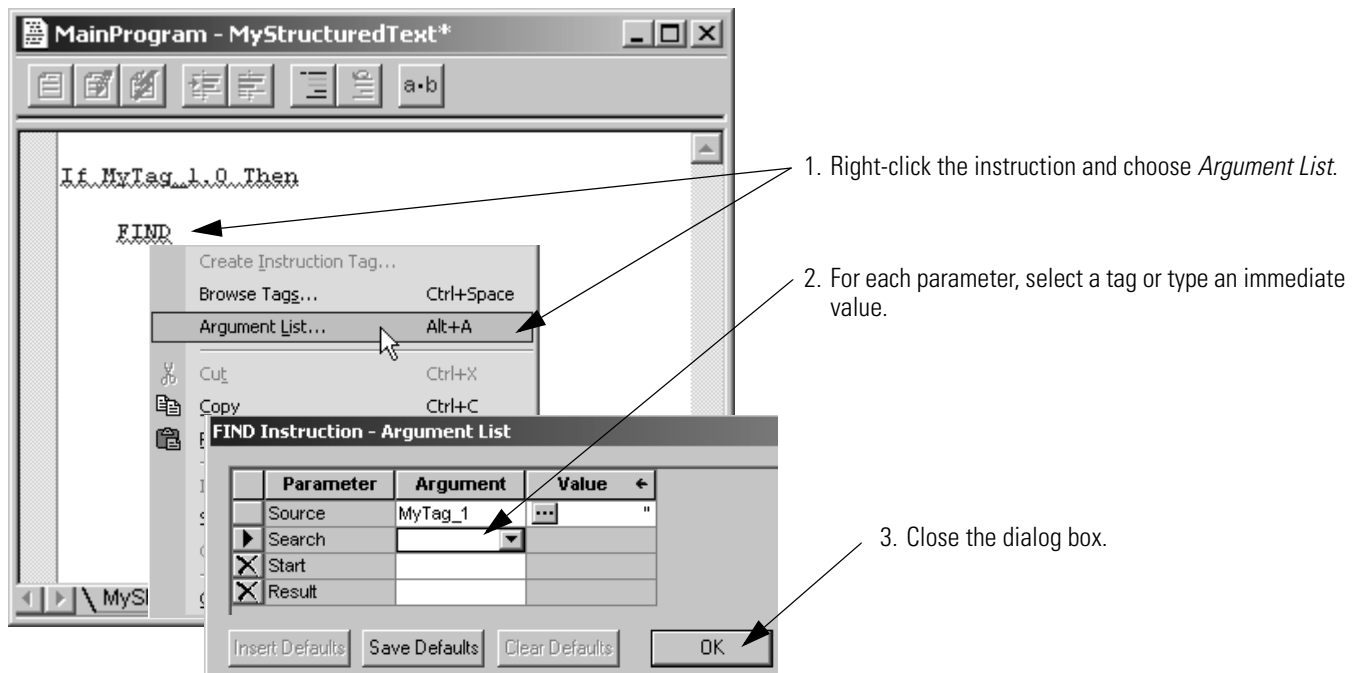
Browse For an Instruction

1. Press [Alt] + [Insert].



2. Type the mnemonic for the instruction and press [Enter].

Assign Operands to an Instruction



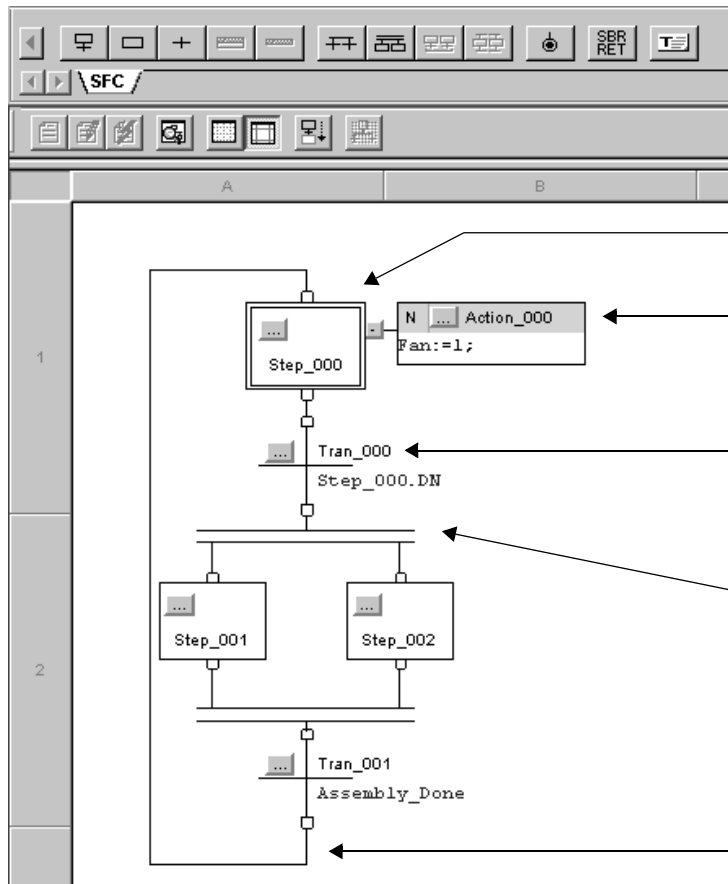
1. Right-click the instruction and choose *Argument List*.

2. For each parameter, select a tag or type an immediate value.

3. Close the dialog box.

Enter a Sequential Function Chart

A sequential function chart (SFC) lets you define a sequence of states (steps) through which your machine or process progresses. The steps can execute structured text, call subroutines, or simply serve as signals for other logic.



step – major function of your process. It contains the actions that occur at a particular time, phase, or station.

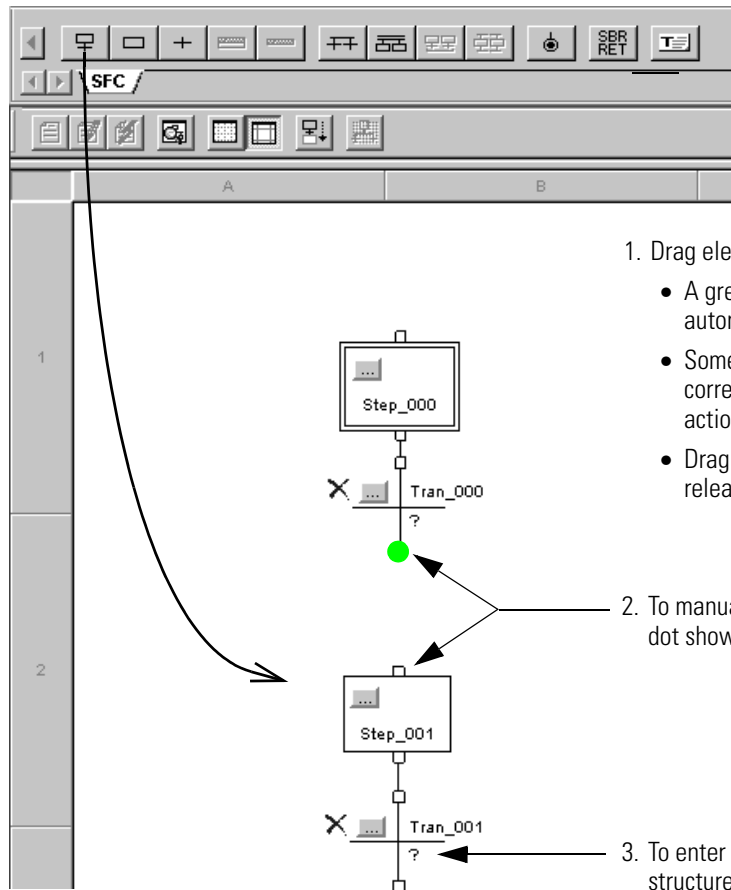
action – one of the functions that a step performs. To program the action, either enter structured text or call a subroutine.

transition – true or false condition that tells the SFC when to go to the next step. To specify the condition, either enter a BOOL expression in structured text or call a subroutine.

branch – execute more than 1 step at the same time (simultaneous) or choose between different steps (selective).

wire – connect one element to another anywhere on the chart.

Enter an SFC



1. Drag elements from the toolbar to the chart.

- A green dot shows a point to which the element will automatically connect if you release the mouse button.
- Some toolbar buttons are active only after you select a corresponding element on the SFC. For example, to add an action, first select a step.
- Drag an action until it is on top of the required step and then release the mouse button.

2. To manually connect elements, click corresponding pins. A green dot shows a valid connection point.

3. To enter structured text, double-click a ? symbol. Then type the structured text and press [Ctrl] + [Enter].

Assign Operands

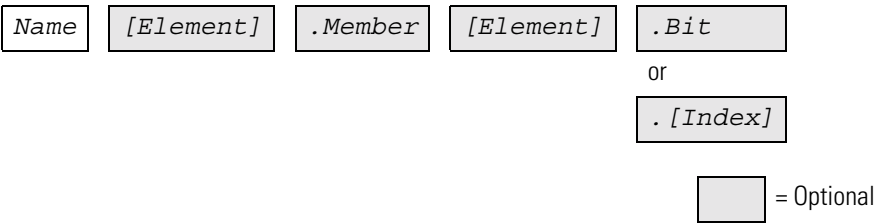
RSLogix 5000 software lets you program according to your workflow. You can enter logic without assigning operands or defining tags. Later, you can go back and assign or define the operands to complete the logic.

The screenshot shows the RSLogix 5000 software interface with three routines visible: **MainProgram - MainRoutine**, **MainProgram - Routine_B***, and **MainProgram - Routine_C**. In **MainRoutine**, there is a logic element with a question mark. In **Routine_B***, there is a logic element with a question mark and a tag **MyData_9**. In **Routine_C**, there is a logic element with a question mark and a tag **MyData_8.0**. Arrows point from the callout boxes to these elements.

missing operand – enter logic without defining operands. RSLogix 5000 software lets you enter and save logic without assigning operands. This lets you develop your logic in iterations and save libraries of code for re-use.

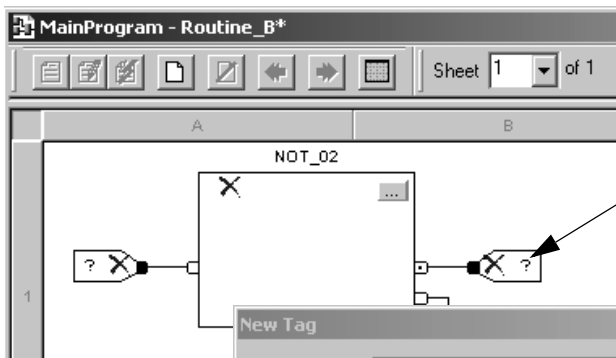
undefined tag – enter a tag name without defining the tag. RSLogix 5000 software lets you enter and save logic without defining all the operands. This lets you develop your logic in iterations.

A tag name follows this format:



Where:	Is:
Name	Name that identifies this specific tag.
Element	Subscript or subscripts that point to a specific element within an array. <ul style="list-style-type: none">Use the element identifier only if the tag or member is an array.Use one subscript for each dimension of the array. For example: [5], [2,8], [3,2,7]. To indirectly (dynamically) reference an element, use a tag or numeric expression that provides the element number. For example, MyArray [Tag_1] , MyArray [Tag_2-1] , MyArray [ABS (Tag_3)] .
Member	Specific member of a structure. <ul style="list-style-type: none">Use the member identifier only if the tag is a structure.If the structure contains another structure as one of its members, use additional levels of the .Member format to identify the required member.
Bit	Specific bit of an integer data type (SINT, INT, or DINT).
Index	To indirectly (dynamically) reference a bit of an integer, use a tag or numeric expression that provides the bit number. For example, MyTag . [Tag_1] , MyTag . [Tag_2-1] , MyTag . [ABS (Tag_4)] .

Create a Tag



New Tag

Name:

Description:

Tag Type: ☒ Base ☐ Alias ☐ Produced ☐ Consumed

Data Type:

Scope:

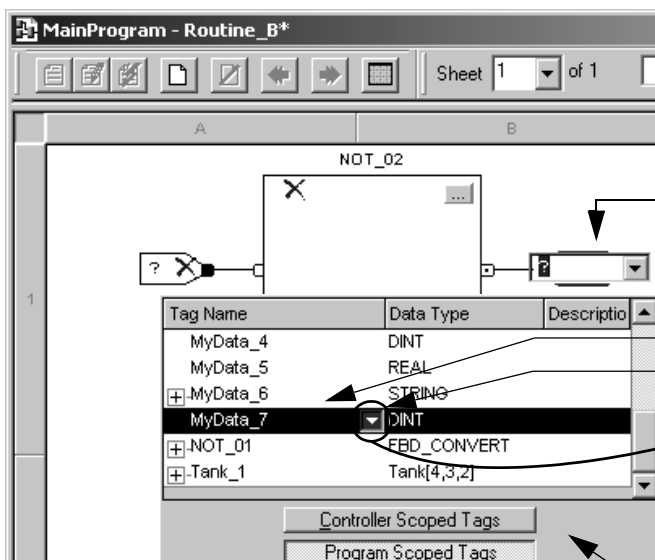
1. Double-click the tag area.
2. Type a name for the tag and press [Enter]
Use underscores "_" in place of spaces.
3. Right-click the tag name and choose New "*Tag_Name*"

4. Type the data type.
To browse for a data type or assign array dimensions, click .

5. Choose the scope for the tag.

6. Choose

Select an Existing Tag



1. Double-click the tag area.

2. Click the ▼.

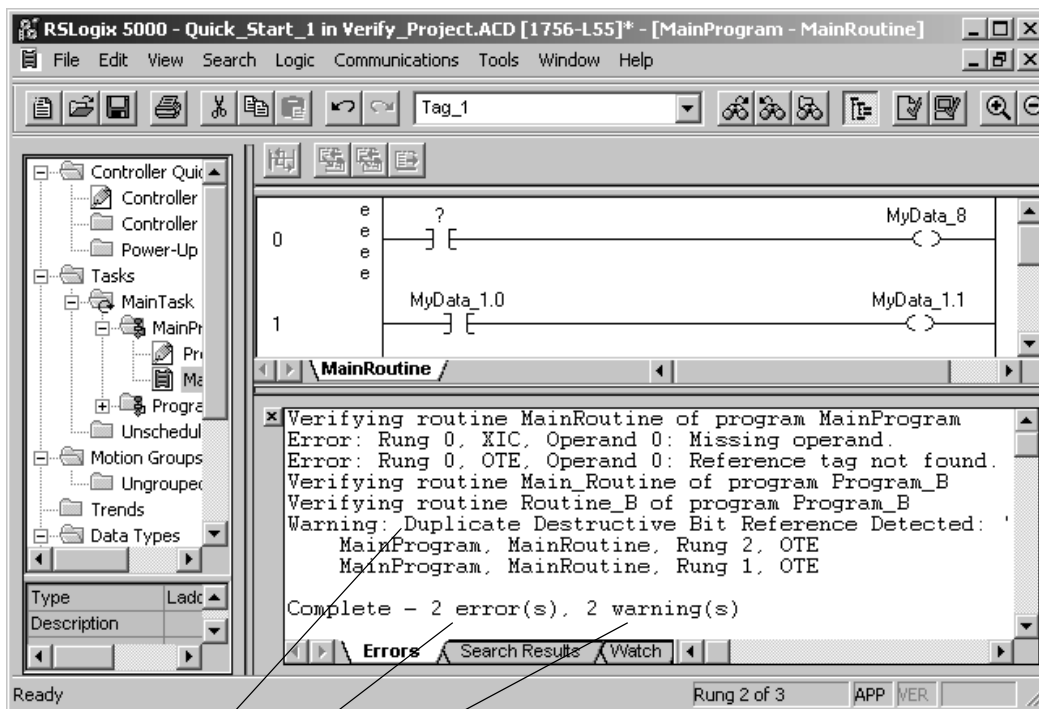
3. Select the desired tag.
To select a bit number, click the ▼.

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31

4. To change the scope of tags in which to look, click the appropriate button.

Verify a Project

As you program your project, periodically verify your work:



verify – check a routine or project for programming errors or incomplete configuration.

warning – situation that may prevent the project from executing as expected. RSLogix 5000 software lets you download a project that contains warnings. Warnings include situations such as duplicate destructive bits and unassigned main routines.

error – situation that you must correct before you download the project. Errors include situations such as missing operands or undefined tags.

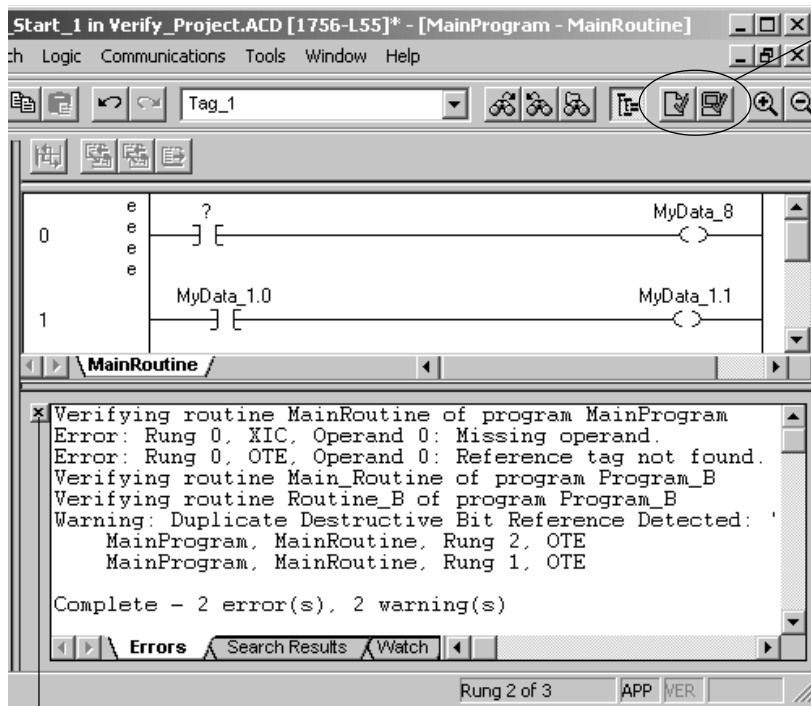
duplicate destructive bit detection – determine if other logic (bit instruction, OREF, ST assignment) also clears or sets the value of a bit that you use in a OTE, ONS, OSF, or OSR instruction. RSLogix 5000 software detects duplicate destructive bits only if *all* of the following conditions are met:

1. You enable duplicate destructive bit detection. (It's off by default.)
2. You use the bit in a ladder logic OTE, ONS, OSF, or OSR instruction.
3. Another logic element such as a bit instruction, OREF, or ST assignment also references that same bit and can change its value.

If you *do not* use a bit in an OTE, ONS, OSF, or OSR instruction, the software does *not* detect any duplicate destructive bits, even if they exist.

By default, duplicate destructive bit detection is turned off.

To verify a routine or project:



1. Choose a verify option:

Verify routine
in view

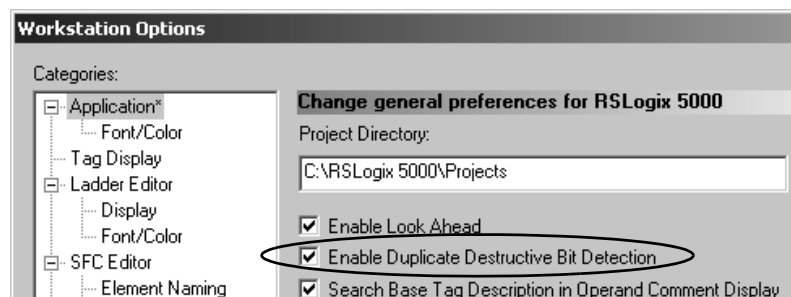
Verify entire
project

2. Go to an error or warning:

To go to:	Do this:
specific error or warning	double-click the error or warning.
cycle through the list of errors and warnings	Press [F4].

3. To close the *Errors* tab, click here.

4. To enable duplicate destructive bit detection (it's off by default), choose *Tools* ⇒ *Options*.



Guidelines for Tags

Use the following guidelines to create tags for a Logix5000 project:

Guideline:	Details:									
<div><input type="checkbox"/> 1. Create user-defined data types.</div>	<p>User-defined data types (structures) let you organize your data to match your machine or process. A user-defined data type provides these advantages:</p> <ul style="list-style-type: none">• One tag contains all the data related to a specific aspect of your system. This keeps related data together and easy to locate, regardless of its data type.• Each individual piece of data (member) gets a descriptive name. This automatically creates an initial level of documentation for your logic.• You can use the data type to create multiple tags with the same data lay-out. <p>For example, use a user-defined data type to store all the parameters for a tank, including temperatures, pressures, valve positions, and preset values. Then create a tag for each of your tanks based on that data type.</p>									
<div><input type="checkbox"/> 2. Use arrays to quickly create a group of similar tags.</div>	<p>An array creates multiple instances of a data type under a common tag name.</p> <ul style="list-style-type: none">• Arrays let you organize a block of tags that use the same data type and perform a similar function.• You organize the data in 1, 2, or 3 dimensions to match what the data represents. <p>For example, use a 2 dimension array to organize the data for a tank farm. Each element of the array represents a single tank. The location of the element within the array represents the geographic location of the tank.</p> <p>Important: Minimize the use of BOOL arrays. Many array instructions <i>do not</i> operate on BOOL arrays. This makes it more difficult to initialize and clear an array of BOOL data.</p> <ul style="list-style-type: none">• Typically, use a BOOL array for the bit-level objects of a PanelView screen.• Otherwise, use the individual bits of a DINT tag or an array of DINTs.									
<div><input type="checkbox"/> 3. Take advantage of program-scoped tags.</div>	<p>If you want multiple tags with the same name, define each tag at the program scope (program tags) for a different program. This lets you re-use both logic and tag names in multiple programs.</p> <p>Avoid using the same name for both a controller tag and a program tag. Within a program, you cannot reference a controller tag if a tag of the same name exists as a program tag for that program.</p> <p>Certain tags must be controller scope (controller tag).</p> <table><tr><th>If you want to use the tag:</th><th>Then assign this scope:</th></tr><tr><td>in more than one program in the project</td><td rowspan="4">controller scope (controller tags)</td></tr><tr><td>in a Message (MSG) instruction</td></tr><tr><td>to produce or consume data</td></tr><tr><td>to communicate with a PanelView terminal</td></tr><tr><td>none of the above</td><td>program scope (program tags)</td></tr></table>	If you want to use the tag:	Then assign this scope:	in more than one program in the project	controller scope (controller tags)	in a Message (MSG) instruction	to produce or consume data	to communicate with a PanelView terminal	none of the above	program scope (program tags)
If you want to use the tag:	Then assign this scope:									
in more than one program in the project	controller scope (controller tags)									
in a Message (MSG) instruction										
to produce or consume data										
to communicate with a PanelView terminal										
none of the above	program scope (program tags)									

Guideline:	Details:										
<input type="checkbox"/> 4. For integers, use the DINT data type.	<p>To increase the efficiency of your logic, minimize the use of SINT or INT data types. Whenever possible, use the DINT data type for integers.</p> <ul style="list-style-type: none">• A Logix5000 controller typically compares or manipulates values as 32-bit values (DINTs or REALs).• The controller typically converts a SINT or INT value to a DINT or REAL value before it uses the value.• If the destination is a SINT or INT tag, the controller typically converts the value back to a SINT or INT value.• The conversion to or from SINTs or INTs occurs automatically with no extra programming. But it takes extra execution time and memory.										
<input type="checkbox"/> 5. Limit a tag name to 40 characters.	<p>Here are the rules for a tag name:</p> <ul style="list-style-type: none">• only alphabetic characters (A-Z or a-z), numeric characters (0-9), and underscores (_)• must start with an alphabetic character or an underscore• no more than 40 characters• no consecutive or trailing underscore characters (_)• not case sensitive										
<input type="checkbox"/> 6. Use mixed case.	<p>Although tags are not case sensitive (upper case <i>A</i> is the same as lower case <i>a</i>), mixed case is easier to read.</p> <table><tr><th>These tags are easier to read:</th><th>Than these tags:</th></tr><tr><td>Tank_1</td><td>TANK_1</td></tr><tr><td>Tank1</td><td>TANK1</td></tr><tr><td></td><td>tank_1</td></tr><tr><td></td><td>tank1</td></tr></table>	These tags are easier to read:	Than these tags:	Tank_1	TANK_1	Tank1	TANK1		tank_1		tank1
These tags are easier to read:	Than these tags:										
Tank_1	TANK_1										
Tank1	TANK1										
	tank_1										
	tank1										
<input type="checkbox"/> 7. Consider the alphabetical order of tags.	<p>RSLogix 5000 software displays tags of the same scope in alphabetical order. To make it easier to monitor related tags, use similar starting characters for tags that you want to keep together.</p>										

Starting each tag for a tank with *Tank* keeps the tags together.

Tag Name
Tank_North
Tank_South
...

Otherwise, the tags may end up separated from each other.

Tag Name
North_Tank
...
...
...
South_Tank

← other tags that start with the letters *o*, *p*, *q*, etc.

Notes:

Document a Project

Using This Chapter

Use this chapter to document your RSLogix 5000 project. This makes the system easier to debug, maintain, and troubleshoot.

If you want to:	See page:
Describe a User-Defined Data Type	4-2
Add Rung Comments	4-4
Enter and Edit Rung Comments Using Microsoft® Excel	4-5
Add Comments to a Function Block Diagram or SFC	4-7
Add Comments to Structured Text	4-9

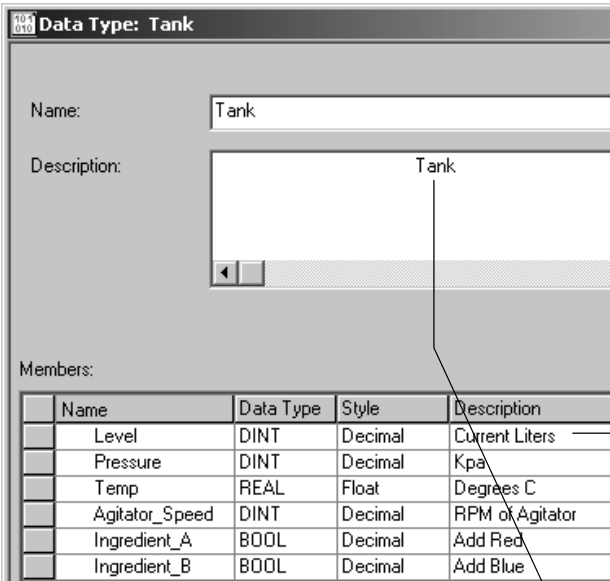
Describe a User-Defined Data Type



RSLogix 5000 software
13.0 or later

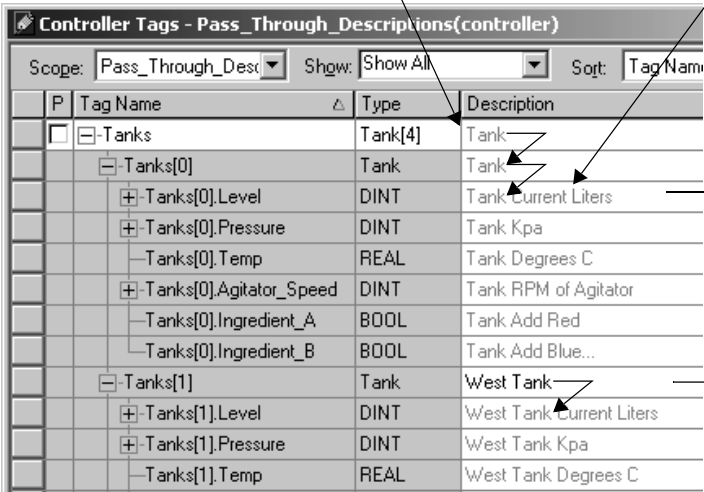
RSLogix 5000 software lets you automatically build descriptions out of the descriptions in your user-defined data types. This greatly reduces the amount of time you have to spend documenting your project.

As you organize your user-defined data types, keep in mind the following features of RSLogix 5000 software:



pass through of descriptions – When possible, RSLogix 5000 software looks for an available description for a tag, element, or member:

- Descriptions in user-defined data types ripple through to the tags that use that data type.
- Description of an array tag ripples through to the elements and members of the array.



append description to base tag – RSLogix 5000 software automatically builds a description for each member of a tag that uses a user-defined data type. It starts with the description of the tag and then adds the description of the member from the data type.

paste pass-through description – Use the data type and array description as a basis for more specific descriptions. In this example, Tank became West Tank.

RSLogix 5000 software uses different colors for descriptions:

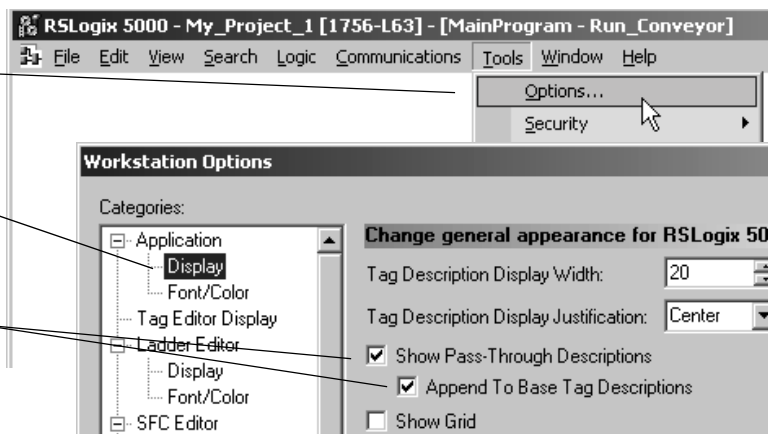
A description in this color:	Is a:
gray	pass-through description
black	manually entered description

Turn Pass-Through and Append Descriptions On or Off

1. In RSLogix 5000 software, choose *Tools* ⇒ *Options*.

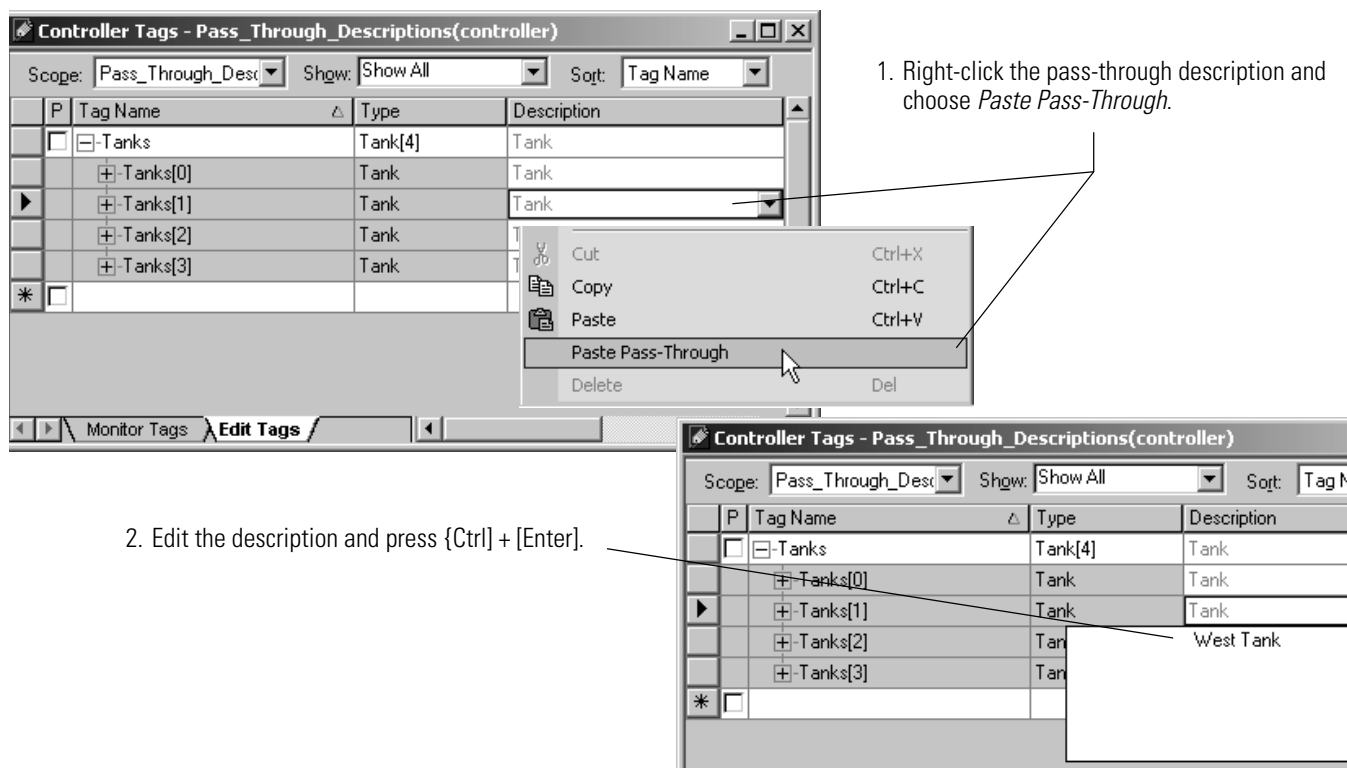
2. Select the *Application* ⇒ *Display*.

3. Turn on (check) or turn off (uncheck) the desired options.



Paste a Pass-Through Description

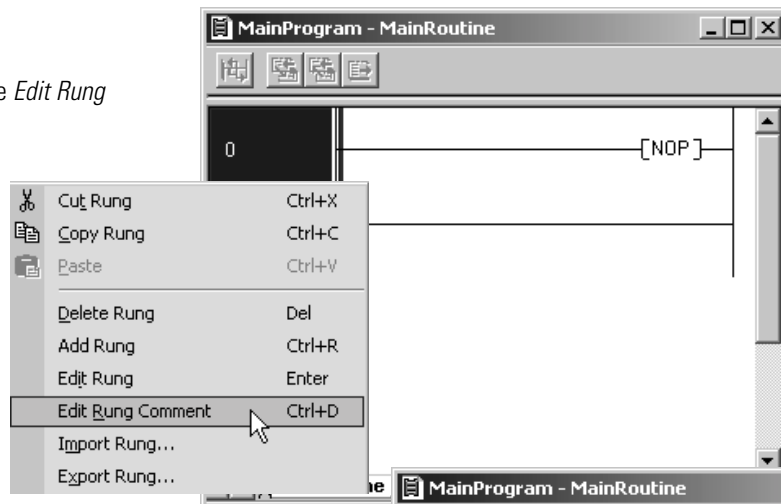
To use a pass-through description as the starting point for a more specific description:



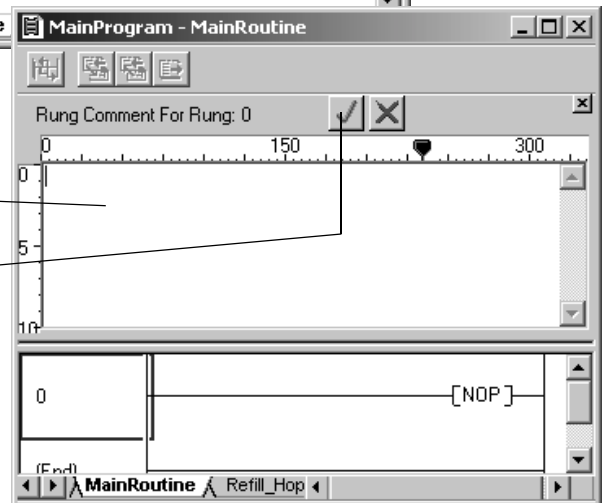
Add Rung Comments

Use a rung comment to describe the operation of a rung of ladder logic. You can also start the routine with a rung that contains only a No Operation (NOP) instruction. Add a comment to this initial rung that describes the routine in general.

1. Right-click the rung and choose *Edit Rung Comment*.



2. Type your comments.



3. Close the entry window.

Enter and Edit Rung Comments Using Microsoft® Excel



RSLogix 5000 software
13.0 or later

You can also use spreadsheet software such as Microsoft Excel to create and edit rung comments. This lets you take advantage of the editing features in the spreadsheet software.

IMPORTANT

Rung comments export in the CSV (comma delimited) format. Make sure you keep that format when you save and close the export file.

Export the Existing Comments

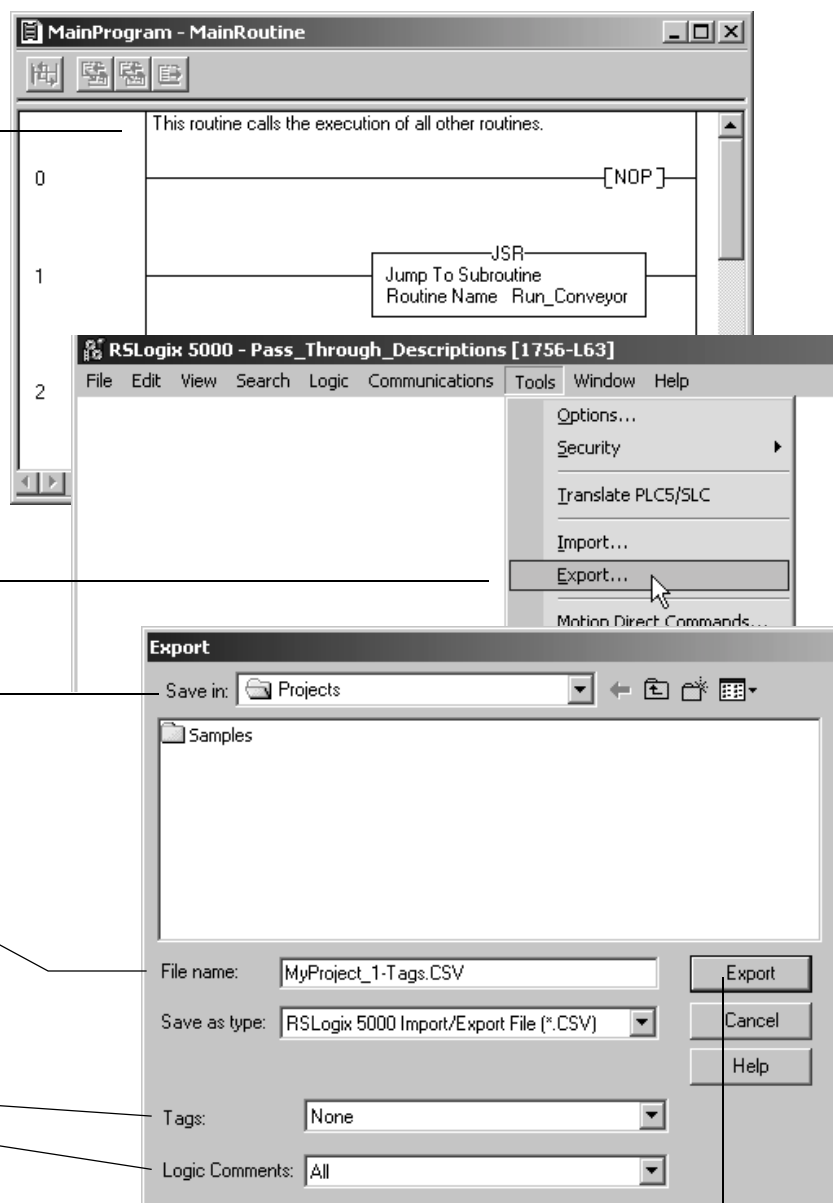
1. In RSLogix 5000 software, add at least 1 rung comment. This helps to format the export file.

2. Choose *Tools* ⇒ *Export*.

3. Note the location and name of the export file.

4. Choose what to export.

5. Export.



Edit the Export File

1. In Microsoft Excel software, open the export file.
2. Enter rung comments in the following format:

MyProject_1-Tags.CSV						
	A	B	C	D	E	F
7	TYPE	SCOPE	ROUTINE	COMMENT	OWNING_ELEMENT	LOCATION
8	RCOMMENT	MainProgram	MainRoutine	This routine calls the execution of all other routines.	NOP()	0
9	RCOMMENT	MainProgram	MainRoutine	If the conveyor is not turning on or off, check this routine.		1
10						

RCOMMENT	program that contains the rung	routine that contains the rung	comments for the rung	leave blank	rung number
----------	--------------------------------	--------------------------------	-----------------------	-------------	-------------

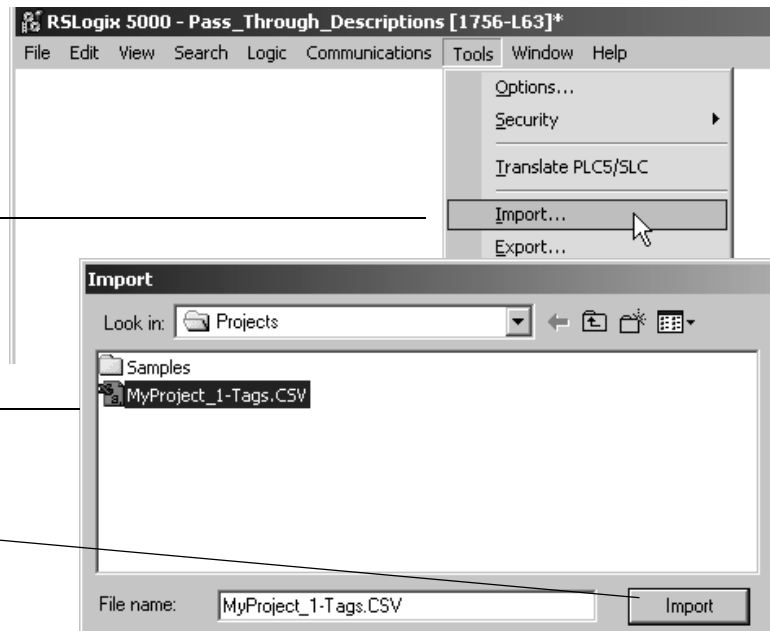
3. Save and close the file. (Keep it in the CSV format.)

Import the New Comments

1. In RSLogix 5000 software, choose *Tools* ⇒ *Import*.

2. Select the file that has the comments you entered (i.e., the export file).

3. Import.



Check the *Errors* tab for the results of the import operation. To refresh the view of the ladder logic and see the comments, close and open the routine.

```
Totals:
  0 tag(s) created
  0 tag(s) overwritten on collision
  0 description(s) imported
  1 new comment(s) imported
  0 comment(s) overwritten on collision
Complete - 0 error(s), 0 warning(s)
```



Errors Search Results Watch

Add Comments to a Function Block Diagram or SFC

Use Text boxes to add notes about the diagram or chart in general or a specific element. Or use a text box to capture information that you will use later on as you develop the project.

Set the Word Wrap Option

Use the word wrap option to control the width of the text box as you type. You set the option for function block diagrams and SFC independent of each other.

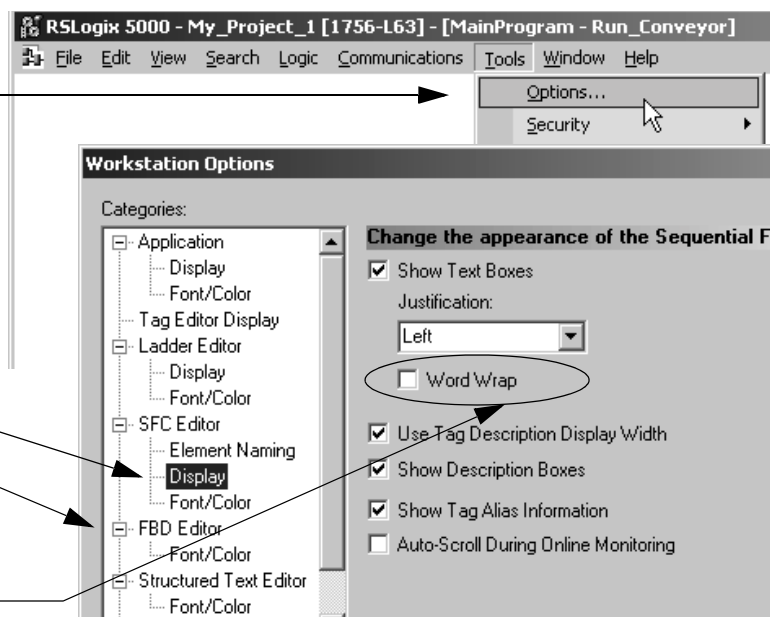
If you want text boxes to:	Then choose this option:
Automatically grow to the width of the longest line of text in the box.	<input type="checkbox"/> Word Wrap
<div>Turns conveyor on and off based on start and stop buttons. If both start and stop are on, the stop button overrides the start button.</div> 	
Retain a fixed width and wrap the text. You can always manually resize the box.	<input checked="" type="checkbox"/> Word Wrap
<div>Turns conveyor on and off based on start and stop buttons. If both start and stop are on, the stop button overrides the start button.</div> 	

To set the word wrap option:

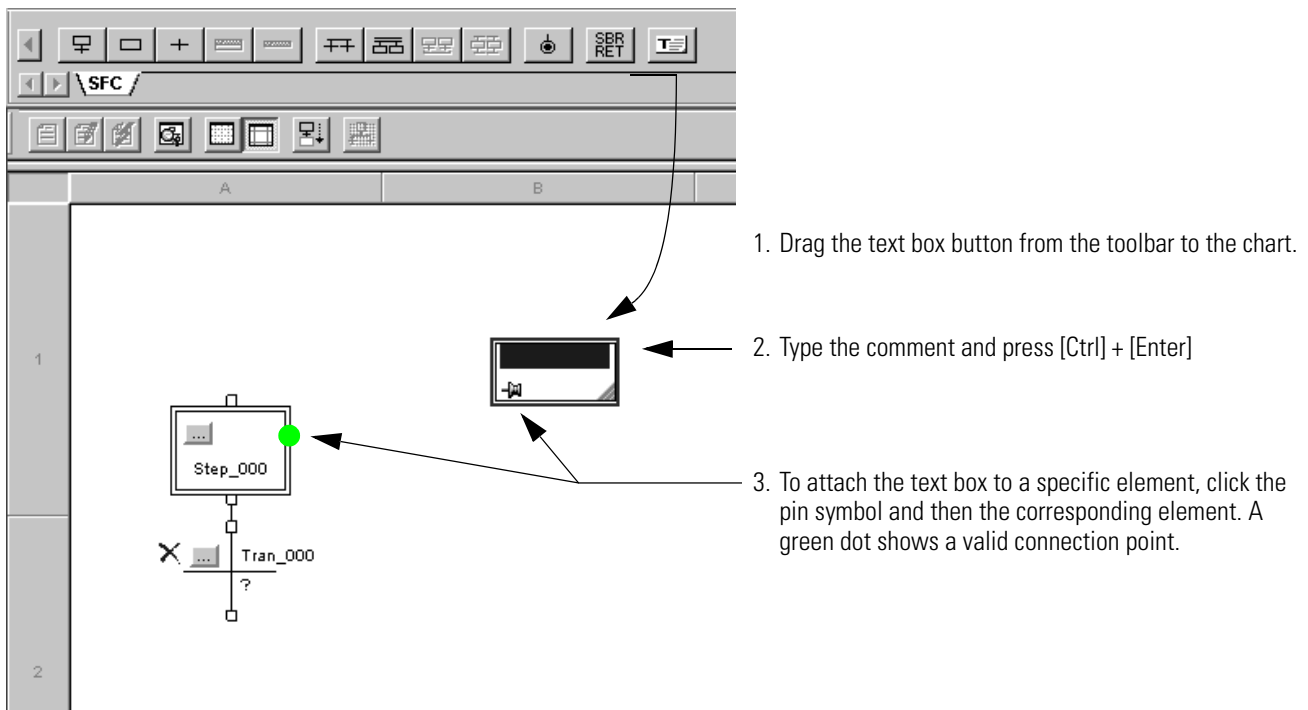
1. In RSLogix 5000 software, choose *Tools* ⇒ *Options*.

2. Select the editor.

3. Select or clear the word wrap option.



Add a Text Box



Add Comments to Structured Text

To make your structured text easier to interpret, add comments. Comments:

- let you use plain language to describe how your structured text works
- download to the controller and upload from the controller
- *do not* affect the execution of the structured text

To add comments to your structured text:

To add a comment:	Use one of these formats:
on a single line	<code>//comment</code>
at the end of a line of structured text	<code>(*comment*)</code> <code>/*comment*/</code>
within a line of structured text	<code>(*comment*)</code> <code>/*comment*/</code>
that spans more than one line	<code>(*start of comment . . . end of comment*)</code> <code>/*start of comment . . . end of comment*/</code>

For example:

Format:	Example:
<code>//comment</code>	<p>At the beginning of a line <code>//Check conveyor belt direction</code> <code>IF conveyor_direction THEN...</code></p> <p>At the end of a line <code>ELSE //If conveyor isn't moving, set alarm light</code> <code>light := 1;</code> <code>END_IF;</code></p>
<code>(*comment*)</code>	<p><code>Sugar.Inlet[:=]1;(*open the inlet*)</code></p> <p><code>IF Sugar.Low (*low level LS*)& Sugar.High (*high level LS*) THEN...</code></p> <p><code>(*Controls the speed of the recirculation pump. The speed depends on the temperature in the tank.*)</code> <code>IF tank.temp > 200 THEN...</code></p>
<code>/*comment*/</code>	<p><code>Sugar.Inlet:=0;/*close the inlet*/</code></p> <p><code>IF bar_code=65 /*A*/ THEN...</code></p> <p><code>/*Gets the number of elements in the Inventory array and stores the value in the Inventory_Items tag*/</code> <code>SIZE(Inventory,0,Inventory_Items);</code></p>

Notes:

Go Online to the Controller

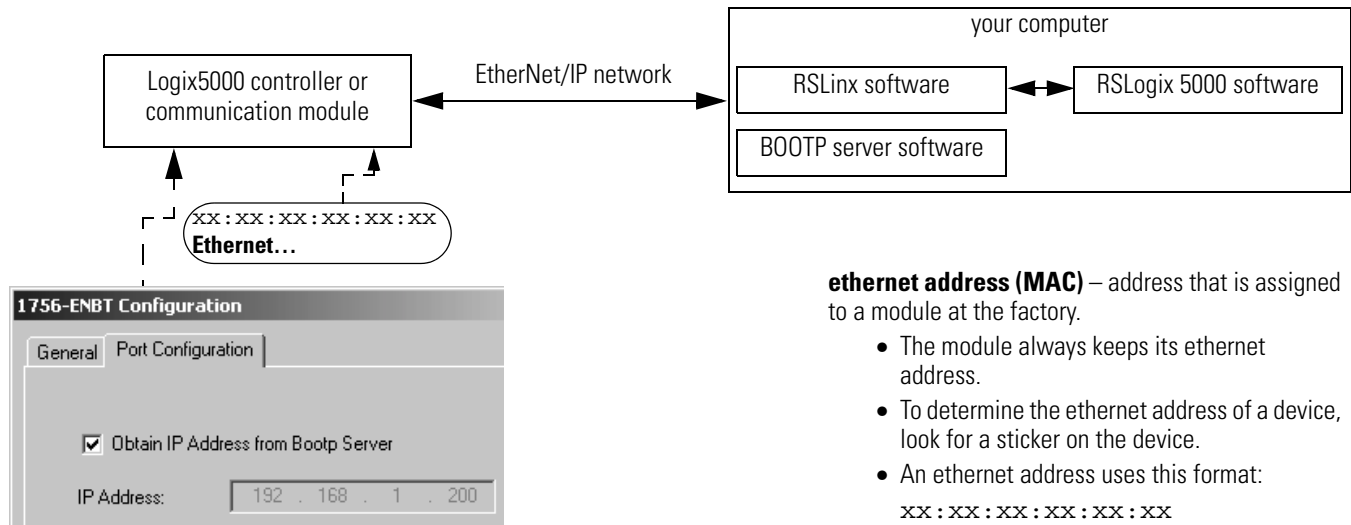
Using This Chapter

Use this chapter to access the project in the controller so you can monitor, edit, or troubleshoot the controller.

To:	See page:
Establish EtherNet/IP Communication with the Controller	5-2
Go Online to a Controller	5-6

Establish EtherNet/IP Communication with the Controller

RSLink® software handles communication between Logix5000 controllers and your software programs, such as RSLogix 5000 software. To communicate with a controller (e.g., download, monitor data), configure RSLink software for the required communication.



ethernet address (MAC) – address that is assigned to a module at the factory.

- The module always keeps its ethernet address.
- To determine the ethernet address of a device, look for a sticker on the device.
- An ethernet address uses this format:
xx : xx : xx : xx : xx : xx

IP address – address that you assign to a module for communication over a specific ethernet network. An IP address uses this format:

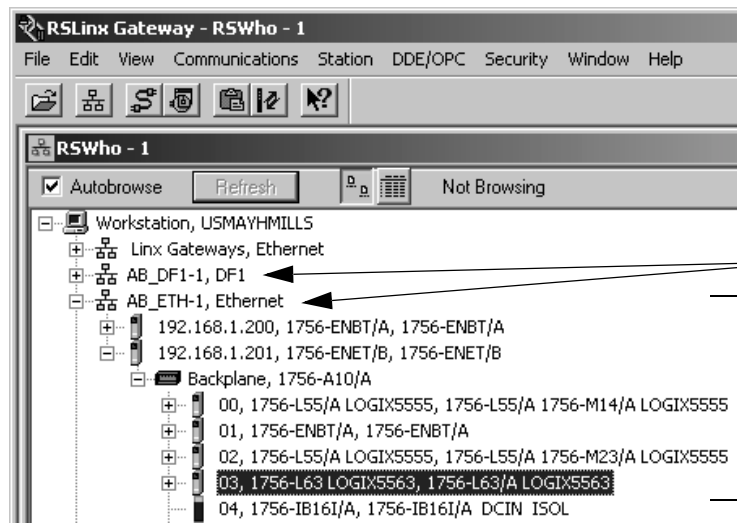
xxx . xxx . xxx . xxx

BOOTP – configure a device to request an IP address over an ethernet network from a BOOTP server. Out of the box, Allen-Bradley EtherNet/IP devices are configured for BOOTP.

BOOTP server – software program that receives BOOTP requests from ethernet devices and assigns IP addresses. RSLink software revision 2.40 and later includes BOOTP server software.

driver – establish communication over a specific network.

path – communication route to a device. To define a path, you expand a driver and select the device.



Equipment and Information That You Need

1. Depending on your controller, you may need a communication module or daughter card:

If you have this controller:	Then install this:	In this location:
1756 ControlLogix controller	1756-ENBT 10/100 Mbps EtherNet/IP Bridge module	open slot in the same chassis as the controller
1769-L35E CompactLogix controller	no additional communication module or card is required.	
1794 FlexLogix controller	1788-ENBT communication daughter card	open slot in the controller

2. For the EtherNet/IP device (controller, bridge module, or daughter card), obtain the following:

Obtain this:	From this source:
ethernet address	sticker on the device
IP address	network administrator
subnet mask	
gateway address (may not be required)	

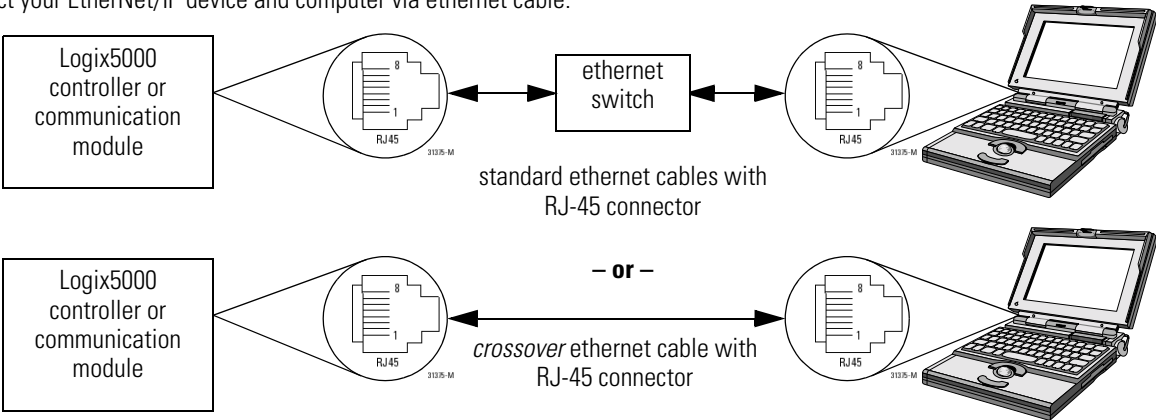
Connect Your EtherNet/IP Device and Computer

WARNING



If you connect or disconnect the communications cable with power applied to this module or any device on the network, an electrical arc can occur. This could cause an explosion in hazardous location installations.

Connect your EtherNet/IP device and computer via ethernet cable.



Assign an IP Address to the Controller or Communication Module

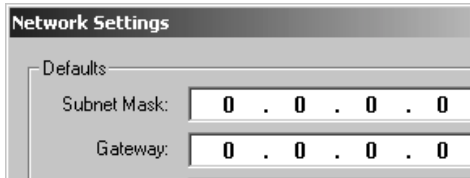
If you *do not* have a serial connection to the controller...

1. Start BOOTP server software:

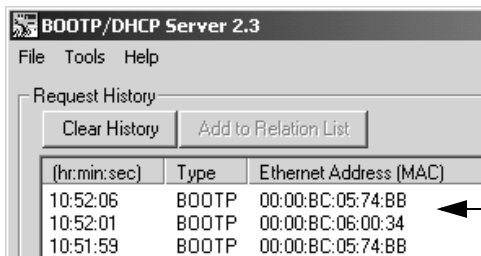
Start ⇒ Programs ⇒ Rockwell Software ⇒ BOOTP-DHCP Server ⇒ BOOTP-DHCP Server

– or –

Start ⇒ Programs ⇒ Rockwell Software ⇒ RSLinx Tools ⇒ BOOTP-DHCP Server.



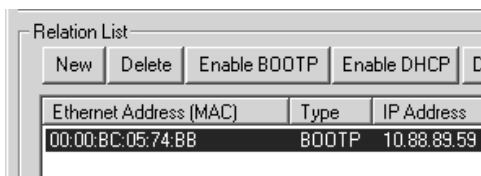
2. If this is the first time you are using the software, type the subnet mask and gateway (if required) for your network and then choose **OK**.



3. Double click the ethernet address of the controller/communication module.



4. Type the IP address and choose **OK**.




5. In the Relation List (lower section), select the device and choose **Disable BOOTP/DHCP**.

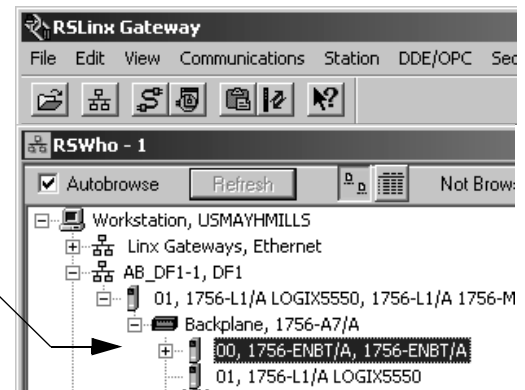
This lets the device keep the address even after a power cycle.

6. When you close the BOOTP server software, you are prompted to save your changes.

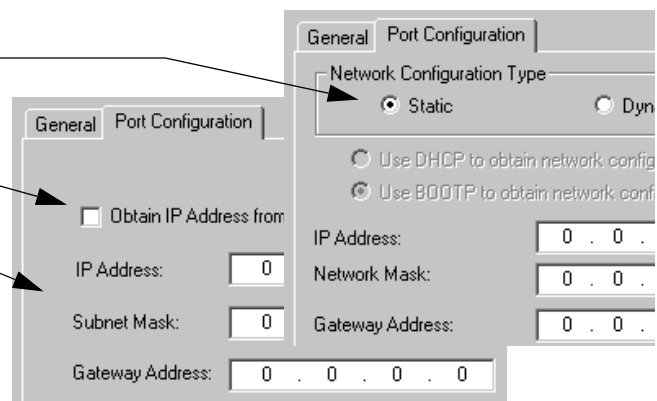
- If you want a record of the IP address that you assigned to the device, save the changes.
- Regardless of whether you save the changes, the device keeps the IP address.


If you have a serial connection to the controller...

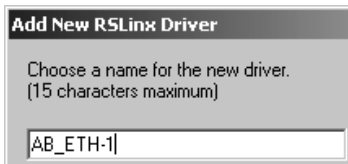
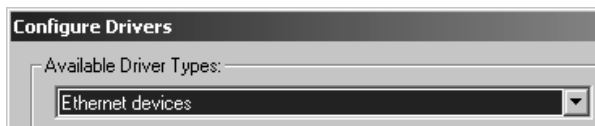
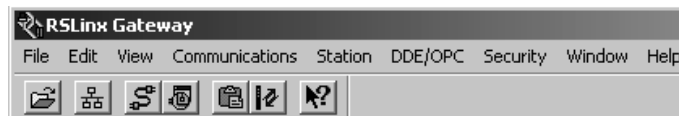
1. Start RSLinx software.
2. Click .
3. Browse to the Ethernet/IP device.
To open a level, click the + sign.
4. Right-click the device and choose *Module Configuration*.
5. Click the Port Configuration tab.



6. Depending on your device, either:
 - Select the *Static* button.
 - Clear (uncheck) the *Obtain IP Address from Bootp Server* check box.
7. Type the:
 - IP address
 - subnet mask
 - gateway address (if required).
8. Choose **OK** and then **Yes** (yes—change IP address).

**Configure an Ethernet Driver**

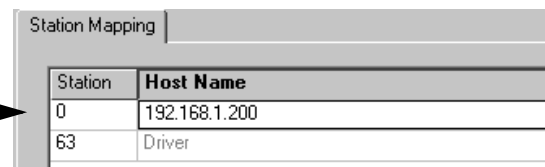
1. Start RSLinx software.
2. Click .



3. Select *Ethernet devices* and choose **Add New...**

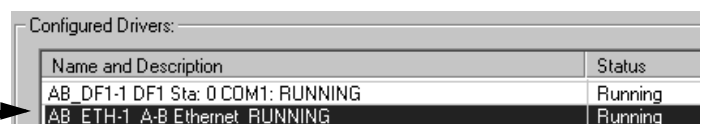
4. Accept the default name.

5. Type the IP address of the controller or communication module.



6. Choose **OK**

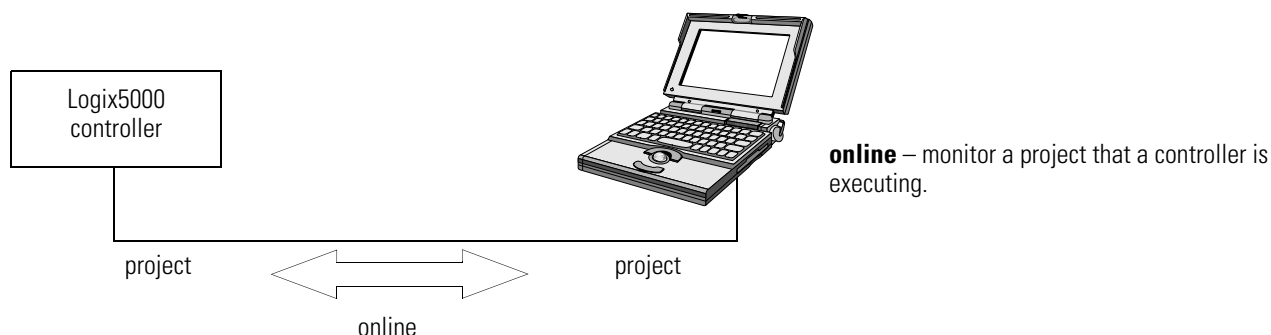
Driver is successfully configured and running.



Go Online to a Controller

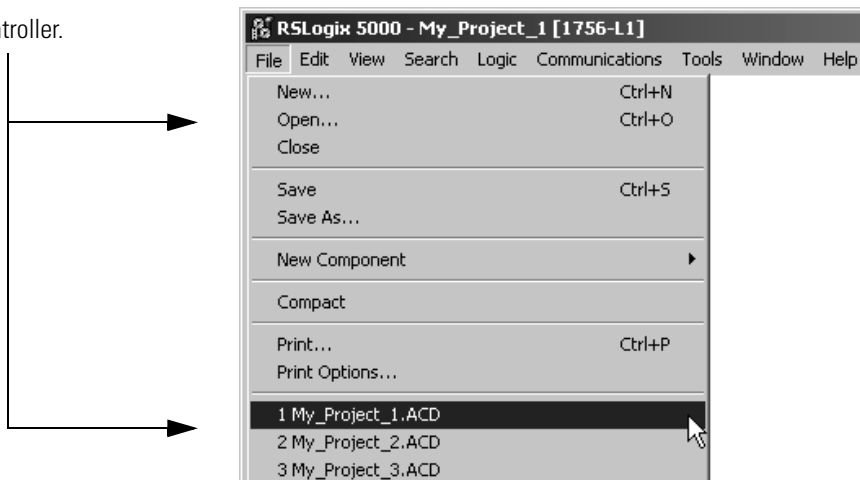
To monitor a project that is executing in a controller, go online with the controller. The procedure that you use depends on whether you have a copy of the project on your computer.

If Your Computer Has the Project For the Controller...



online – monitor a project that a controller is executing.

1. Open the RSLogix 5000 project for the controller.



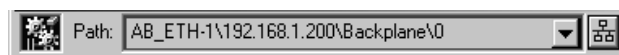
2. Define the path to the controller:

a. Click .

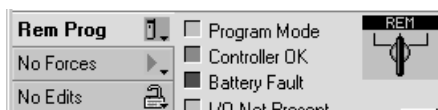
b. Select the controller.

- To open a level, click the + sign.
- If a controller is already selected, make sure that it is the correct controller.

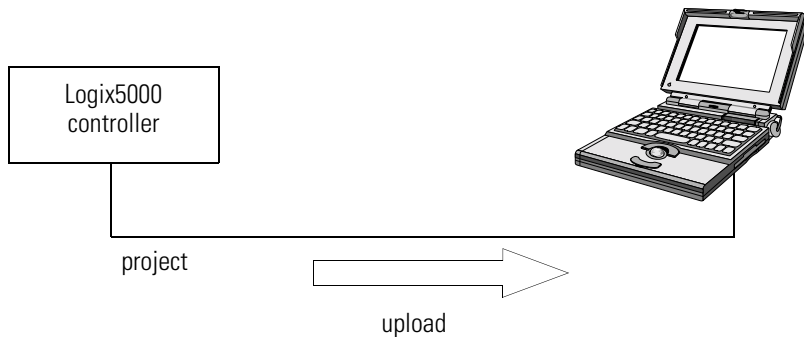
3. Choose .



operating mode of the controller




If Your Computer *Does Not* Have the Project For the Controller...



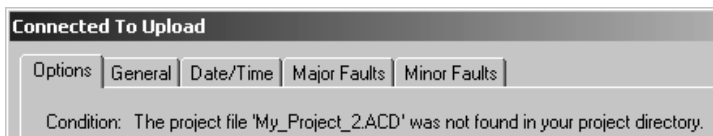
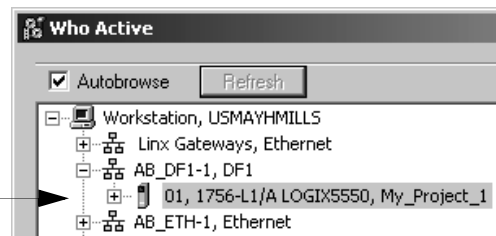
upload – transfer a project from a controller to your computer so you can monitor the project.

1. Define the path to the controller:

- Click .
- Select the controller.
 - To open a level, click the + sign.
 - If a controller is already selected, make sure that it is the correct controller.



2. Choose **Upload...**

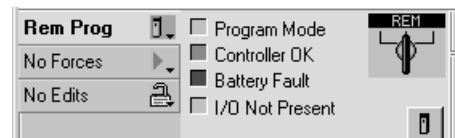


3. Create the project file on your computer:

a. Choose **Select File...**

b. Choose **Select** and then **Yes**

operating mode of the controller



Notes:

Program a Project Online

Using This Chapter

Use this chapter to edit your logic while the controller continues to control your machine or process.

To:	See page:
Edit Logic While Online	6-1
Finalize All Edits in a Program	6-5

Edit Logic While Online

Online edits let you change your logic while your machine or process continues to run.

ATTENTION



Use extreme caution when you edit logic online. Mistakes can injure personnel and damage equipment. Before you edit online:

- Assess how machinery will respond to the changes.
- Notify all personnel of the changes.

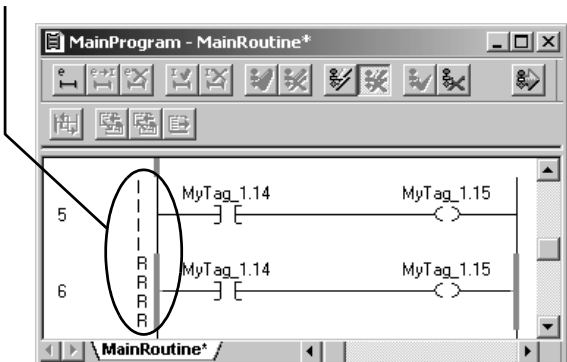
IMPORTANT

When you edit an SFC online:

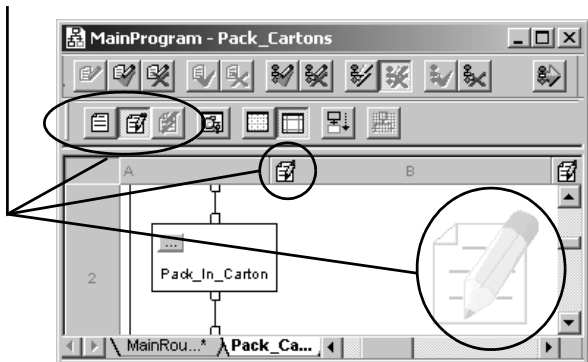
- The SFC resets to the initial step.
- Stored actions turn off.

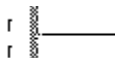



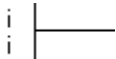
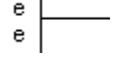


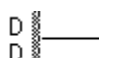

As you perform online edits, RSLogix 5000 software uses markers to show the state of your edits:

relay ladder



function block, structured text, SFC

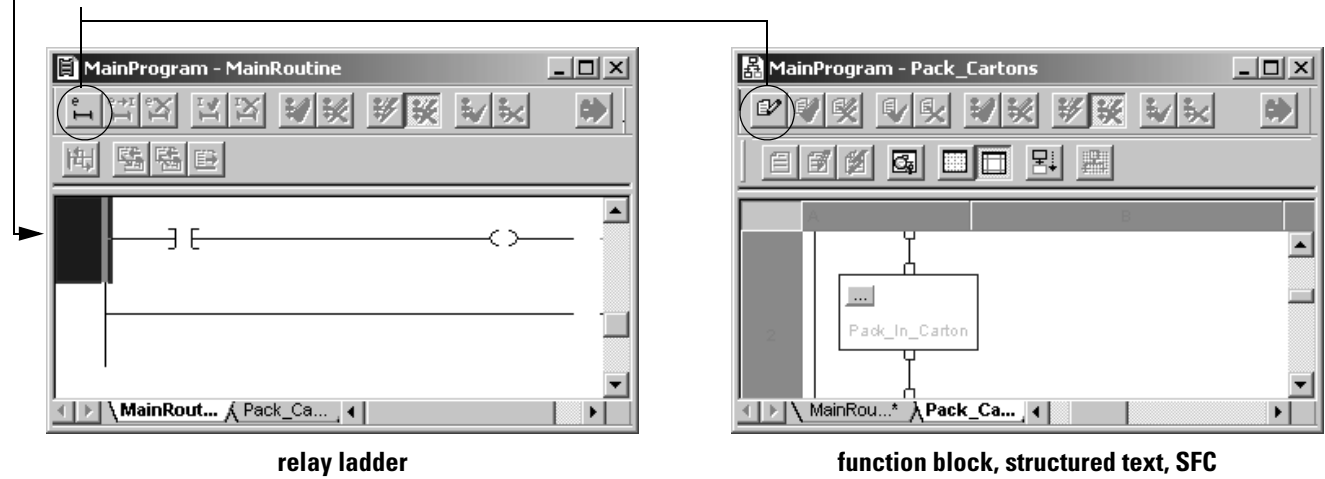


This marker:	Means:	Description:								
relay ladder	 - or - 	<p>original logic</p> <p>When online, RSLogix 5000 software continues to show you the original logic while you edit a copy of the logic (pending edit). A green border or side rail shows which logic the controller is currently running.</p> <p>In function block, structured text, or SFC, use the buttons above the routine to switch between different views.</p>								
function block structured text SFC										
relay ladder	 - or - 	<p>pending edits</p> <p>This is a copy of the original logic for you to edit. Any changes remain on your computer until you accept the edits.</p> <ul style="list-style-type: none">• In relay ladder, you edit individual rungs within a routine.• In function block, structured text, or SFC, you edit an entire routine.								
function block structured text SFC										
relay ladder	 - or - 	<p>test edits</p> <p>When you accept your pending edits, the software downloads them to the controller and marks them as test edits but the controller continues to execute the original logic. You then manually switch execution to the test edits or back to the original logic (test and untest the edits).</p>								
function block structured text SFC		<table><tr><th>If you:</th><th>Then:</th></tr><tr><td>test the edits</td><td><ul style="list-style-type: none">• Execution switches to the test edits (all test edits execute).• Outputs in the original logic stay in their last state unless executed by the test edits (or other logic).• In an SFC, the chart resets to the initial step and stored actions turn off.</td></tr><tr><td>untest the edits</td><td><ul style="list-style-type: none">• Execution switches back to the original logic.• Outputs in the test edits stay in their last state unless executed by the original logic (or other logic).• In an SFC, the chart resets to the initial step and stored actions turn off.</td></tr><tr><td>assemble the edits</td><td>The test edits permanently replace the original logic.</td></tr></table> <p>In relay ladder, if you delete a rung the software immediately marks it as a test edit (upper-case "D" character).</p>	If you:	Then:	test the edits	<ul style="list-style-type: none">• Execution switches to the test edits (all test edits execute).• Outputs in the original logic stay in their last state unless executed by the test edits (or other logic).• In an SFC, the chart resets to the initial step and stored actions turn off.	untest the edits	<ul style="list-style-type: none">• Execution switches back to the original logic.• Outputs in the test edits stay in their last state unless executed by the original logic (or other logic).• In an SFC, the chart resets to the initial step and stored actions turn off.	assemble the edits	The test edits permanently replace the original logic.
If you:	Then:									
test the edits	<ul style="list-style-type: none">• Execution switches to the test edits (all test edits execute).• Outputs in the original logic stay in their last state unless executed by the test edits (or other logic).• In an SFC, the chart resets to the initial step and stored actions turn off.									
untest the edits	<ul style="list-style-type: none">• Execution switches back to the original logic.• Outputs in the test edits stay in their last state unless executed by the original logic (or other logic).• In an SFC, the chart resets to the initial step and stored actions turn off.									
assemble the edits	The test edits permanently replace the original logic.									

Start a Pending Edit

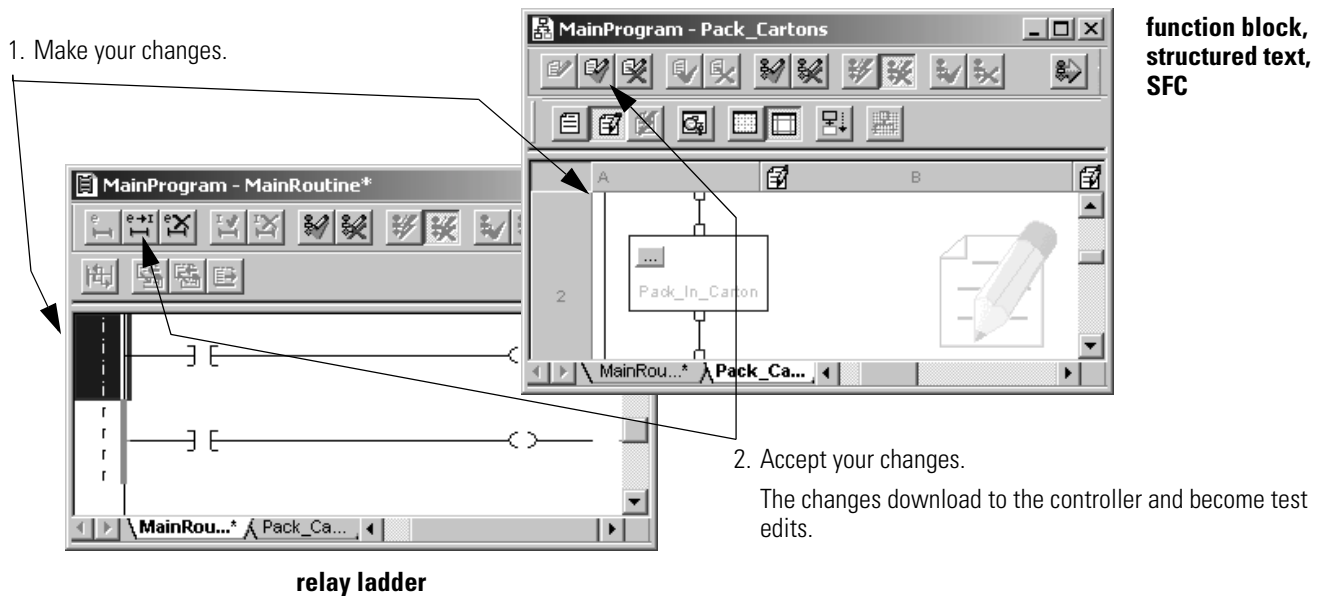
1. For relay ladder, click (select) the rung that you want to edit.

2. Start a pending edit.



Make and Accept Your Edits

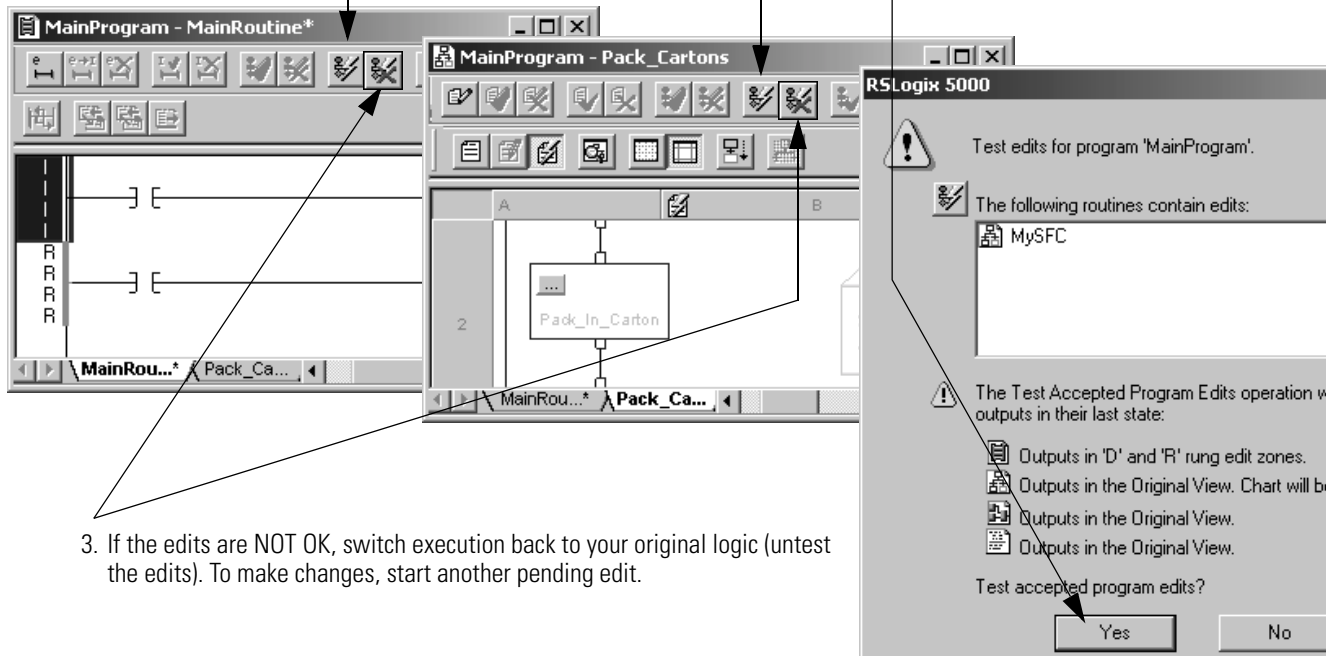
1. Make your changes.



Test the Edits

1. Test the edits to see if they execute as intended.

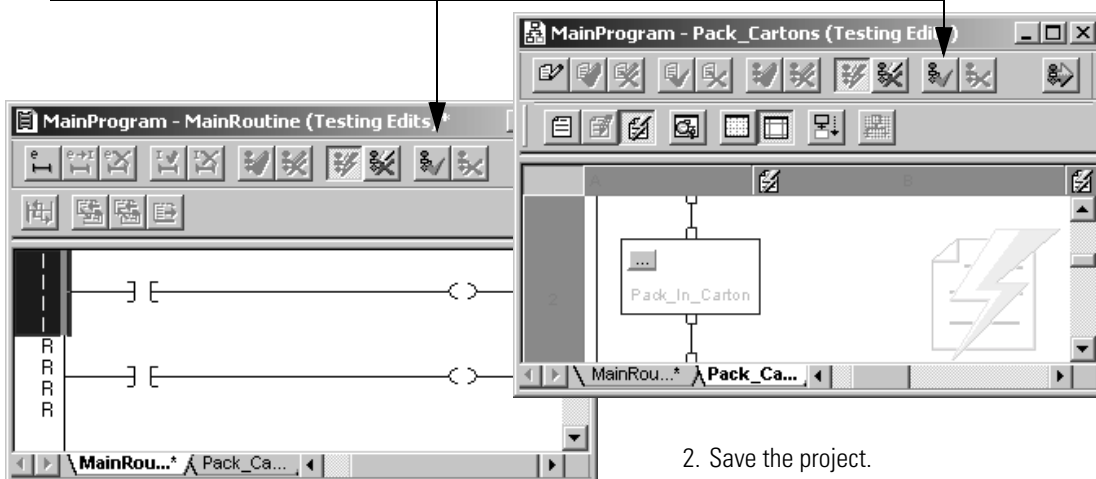
2. Yes—test the edits.



Assemble and Save the Edits

1. Assemble the edits.

The edits become permanent and the original logic is removed.



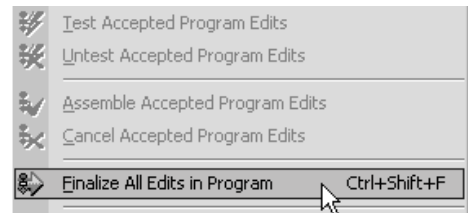
Finalize All Edits in a Program



RSLogix 5000 software
13.0 or later

The *Finalize All Edits in Program* option lets you make an online change to your logic *without* testing the change.

Finalize All Edits in Program



ATTENTION



Use extreme caution when you edit logic online. Mistakes can injure personnel and damage equipment. Before you edit online:

- Assess how machinery will respond to the changes.
- Notify all personnel of the changes.

When you choose *Finalize All Edits in Program*:

- All edits in the program (pending and test), immediately download to the controller and begin execution.
- The original logic is permanently removed from the controller.
- Outputs that were in the original logic stay in their last state unless executed by the new logic (or other logic).
- If your edits include an SFC:
 - The SFC resets to the initial step.
 - Stored actions turn off.

To use the *Finalize All Edits in Program* option:

1. Start a pending edit.
2. Make your change.
3. Choose *Finalize All Edits in Program*.

Notes:

Troubleshoot the Controller

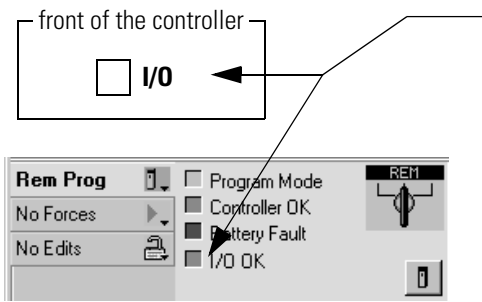
Using This Chapter

Use this chapter to obtain basic diagnostic information about your system and perform basic troubleshooting tasks.

If:	Then:	See page:
there is a problem with several of the devices in your system, communication with an I/O module may have failed.	Troubleshoot I/O Communication	7-2
your entire process unexpectedly shut down, the controller may have experienced a major fault.	Clear a Major Fault	7-4
you want to find a specific element (tag, instruction, etc.) within a project	Search a Project	7-5
you want to browse the project for a specific element (tag, instruction, etc.)	Browse Logic	7-7
you want to: <ul style="list-style-type: none">• override input data• override logic• check wiring to an output device	Force an I/O Value	7-8
you want to sample the data of one or more tags over at a specific period.	Create and Run a Trend (Histogram)	7-11
you want to see the scan time of a task or program.	View Scan Time	7-13

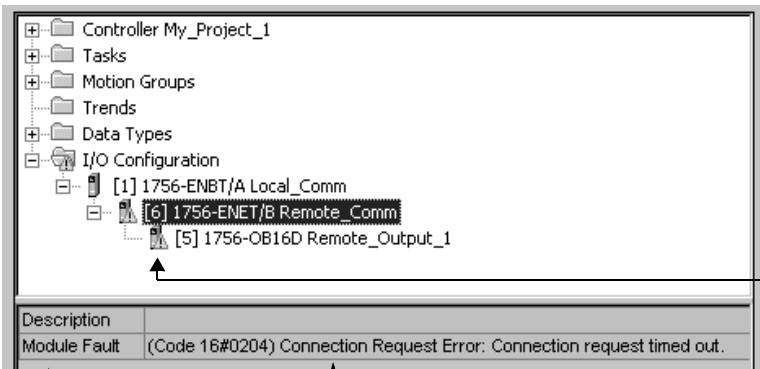
Troubleshoot I/O Communication


If there is a problem with several of the devices in your system, communication with an I/O module may have failed.



Status of I/O communication

If:	Then:
off	Either: <ul style="list-style-type: none"> There are <i>no</i> modules in the I/O configuration of the controller. The controller does <i>not</i> contain a project (controller memory is empty).
solid green	The controller is communicating with all the modules in its I/O configuration.
flashing green	One or more modules in the I/O configuration of the controller are <i>not</i> responding.



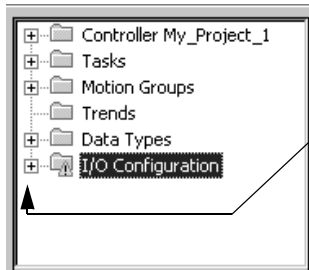
A  over a module means that the controller is *not* communicating with the module.

connection – communication link between 2 devices, such as between a controller and I/O module, PanelView terminal, or another controller. Logix5000 controllers use connections to communicate with the modules in its I/O configuration.

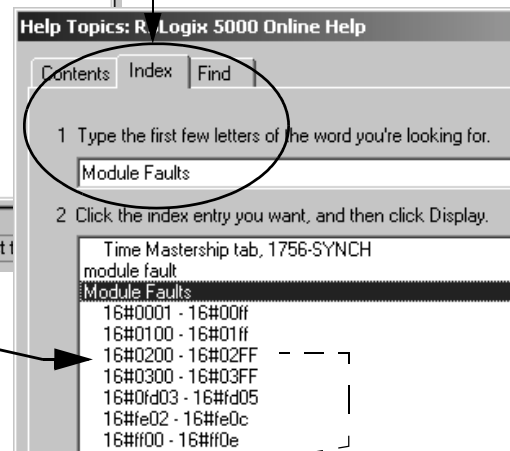
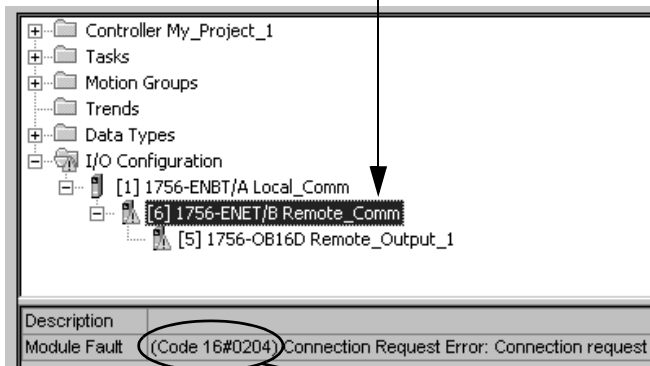
module fault – communication with a module has failed.

TIP

Troubleshoot communication modules first. A faulted communication module effects the modules that are under it.



1. Go online with the controller.
2. If necessary, click the + signs of the I/O Configuration tree to show the faulted modules
3. Select the faulted module.
4. Choose *Help* ⇒ *Contents*.
5. Click the *Index* tab and type module faults.

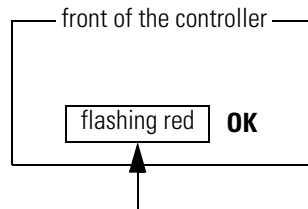


6. Select the corresponding module fault information and choose **Display**

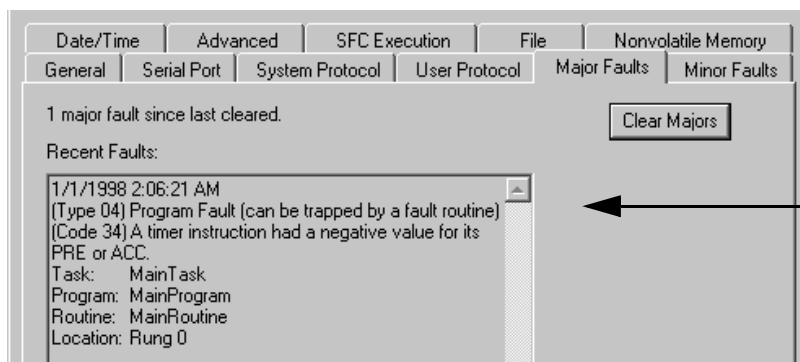
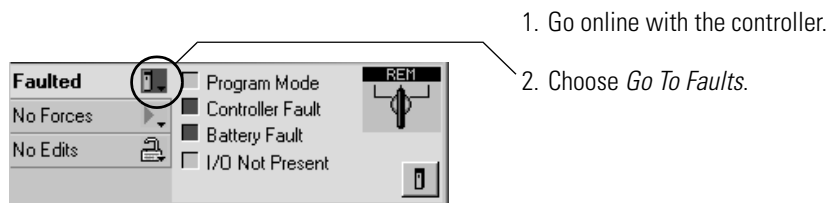
Module Faults: 16#0200 - 16#02ff		
Code:	String:	Explanation and Possible Causes/Solutions:
16#0203	Connection timed out.	The connection to this module has been interrupted causing a loss of communication. <ul style="list-style-type: none"> Ensure that the module has not been removed and is still functioning and is receiving power. For FLEX I/O modules, ensure that the correct terminal block is in use. Ensure that the network connection to this module has not been interrupted. Call Technical Support <p>Note: If a connection to an output module times out and the output module supports Fault Mode and the output module is still functioning, its outputs will transition to the configured Fault Mode.</p>
16#0204	Connection Request Error: Connection request timed out.	The controller is attempting to make a connection to the module and the module is not responding. The controller is not able to communicate with the module. <ul style="list-style-type: none"> Ensure that the module has not been removed and is still functioning and is receiving power. For FLEX I/O modules, ensure that the correct terminal block is in use. Ensure you have entered the correct slot number.

Clear a Major Fault

If your entire process unexpectedly shut down, the controller *may* have experienced a major fault.



major fault – the controller detected a fault condition that is severe enough for it to shut down.



3. Use this information to correct the cause of the fault.

For more information about a fault code, see *Logix5000 Controllers System Reference*, publication 1756-QR107.

4. After you correct the cause of the fault, choose **Clear Majors**

Search a Project



You can find an element of your logic (tag, instruction, comment, etc.) based on the characters that you search for:

To find a:	Specify:	Example:
tag	full or partial tag name	MyTag_1
comment/description	text within the comment/description	fan
instruction	mnemonic of the instruction	OTE
instruction and tag	mnemonic and tag	OTE MyTag_1

Search for All Occurrences of a Tag, Instruction, etc.

1. Open the RSLogix 5000 project that you want to search
2. Choose *Search* ⇒ *Find*.
3. Specify the search criteria:

a. Type the characters to find


To browse for a tag, click , select the tag, and choose .


To select a bit number, click the ▼.

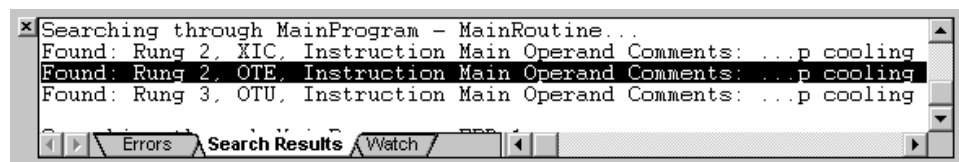
0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31

b. Choose *Text Only*.

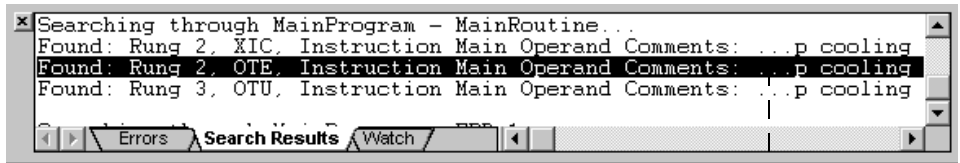
c. Choose *All Routines*.

d. Select each language and check the options in which to search.
To display this section of the dialog box, choose .

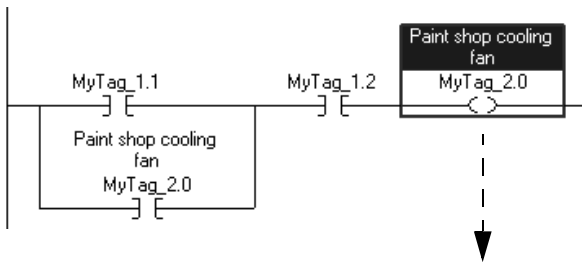
4. Choose .



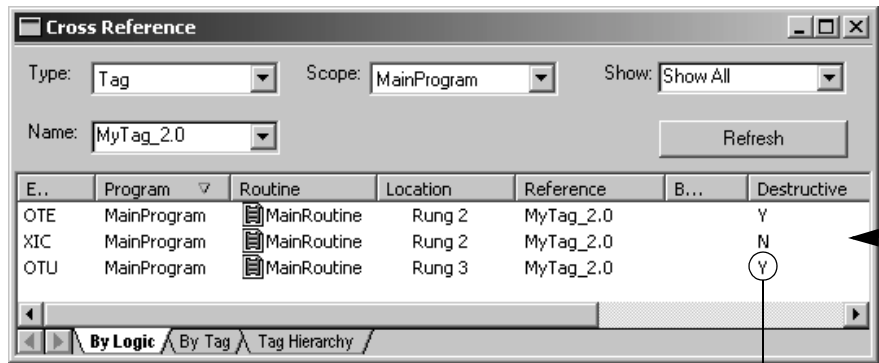
Go to an Instruction



1. To go to an instruction, double-click it.



2. To show a list of cross-references to a tag, right-click and choose *Go To Cross Reference...*



3. To go to an instruction, double-click it.

A "Y" means this instruction changes the value of the tag.

Browse Logic



RSLogix 5000 software
13.0 or later

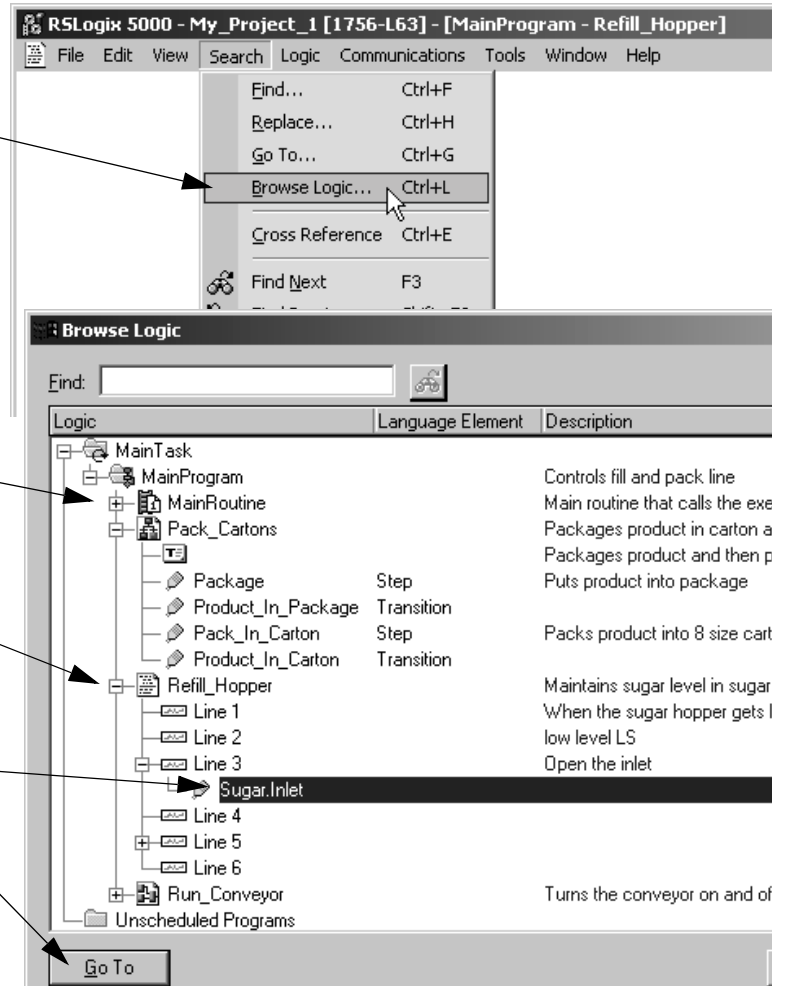
To browse the logic of a routine for a specific item (instruction, element, tag, comment, etc.), use the Browse Logic window.

1. In RSLogix 5000 software, choose *Search* ⇒ *Browse Logic*.

2. To expand an entry and see its contents, either:
 - Double-click the entry.
 - Click the + sign.
 - Right-click the entry and choose *Expand All*.

3. To collapse an entry and hide its contents, either:
 - Double-click the entry.
 - Click the - sign.

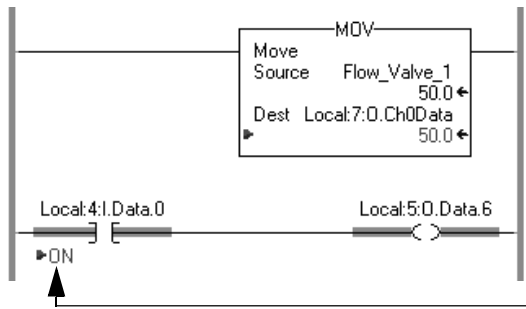
4. To go to the location of a element in logic, select the element and choose *Go To*.



Force an I/O Value

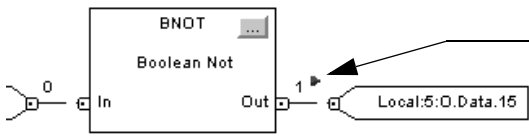
Use a force to override input data or logic when you need to:

- test and debug your logic
- check wiring to an output device
- temporarily keep your process functioning when an input device has failed

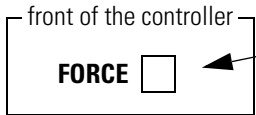


force – override a value from an input device or logic

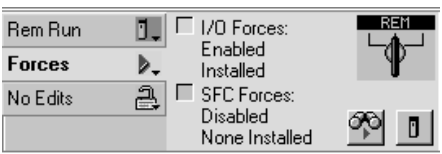
- Forcing an input tag overrides the value from the input device.
- Forcing an output tag overrides your logic and sends the force value to the output device.



When forces are in effect (enabled), a ► appears next to the forced element.



Status of I/O forces



If:	Then:
off	<ul style="list-style-type: none">• No tags contain I/O force values.• I/O forces are inactive (disabled).
flashing amber	<ul style="list-style-type: none">• One or more tags contain a force value.• I/O forces are inactive (disabled).• When you enable I/O forces, <i>all</i> existing I/O forces take effect.
solid amber	<ul style="list-style-type: none">• I/O forces are active (enabled).• Force values may or may not exist.• When you install (add) a force, it immediately takes effect.

If you want to:	Then:
override a value	Install an I/O Force (Force an I/O Value)
stop an individual force but leave other forces enabled and in effect	Remove an Individual Force
stop all I/O forces but leave the I/O forces in the project	Disable All I/O Forces

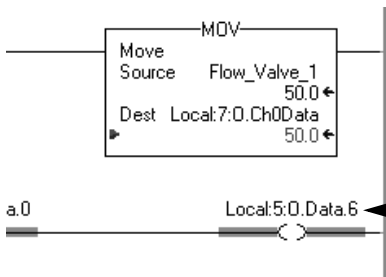
ATTENTION



Forcing can cause unexpected machine motion that could injure personnel. Before you install, disable, or remove a force, determine how the change will effect your machine or process and keep personnel away from the machine area.

- Enabling I/O forces causes input, output, produced, or consumed values to change.
- If you remove an individual force, forces remain in the enabled state.
- If forces are enabled and you install a force, the new force immediately takes effect.

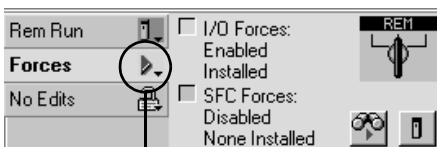
Install an I/O Force (Force an I/O Value)



1. Go online with the controller and open the routine that contains the tag that you want to force.
2. Right-click the tag and choose *Monitor...*
3. If necessary, click the + sign of the tag to show the value that you want to force (e.g., BOOL value of a DINT tag).

Tag Name	Value	Force Mask
+ Local:4:C	{...}	{...}
- Local:4:I	{...}	Forced
+ Local:4:I.Fault	2#00...	
- Local:4:I.Data	2#0...	2#...._...
Local:4:I.Data.0	1	1
Local:4:I.Data.1	0	

4. Install the force value:



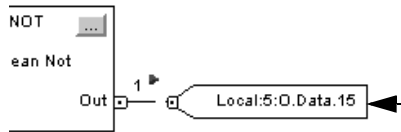
To force a:

Do this:

- | | |
|-----------------------|---|
| BOOL value | Right-click the tag and choose <i>Force ON</i> or <i>Force OFF</i> . |
| integer or REAL value | In the <i>Force Mask</i> column for the tag, type the value to which you want to force the tag and press [Enter]. |

5. Choose *I/O Forcing* ⇒ *Enable All I/O Forces*. and choose (yes—enable I/O forces).

Remove an Individual Force

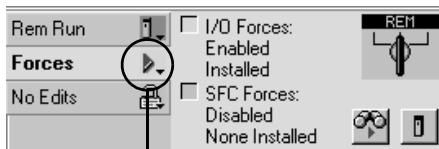


1. Go online with the controller and open the routine that contains the tag that you want to force.
2. Right-click the tag and choose *Monitor...*
3. If necessary, click the + sign of the tag to show its members (e.g., BOOL value of a DINT tag).

Tag Name	Value	Force Mask
+ Local:4:C	{...}	{...}
- Local:4:I	{...}	Forced
+ Local:4:I.Fault	2#00...	
- Local:4:I.Data	2#0...	2#...._...
Local:4:I.Data.0	1	1
Local:4:I.Data.1	0	

4. Right-click the tag and choose *Remove Force*.

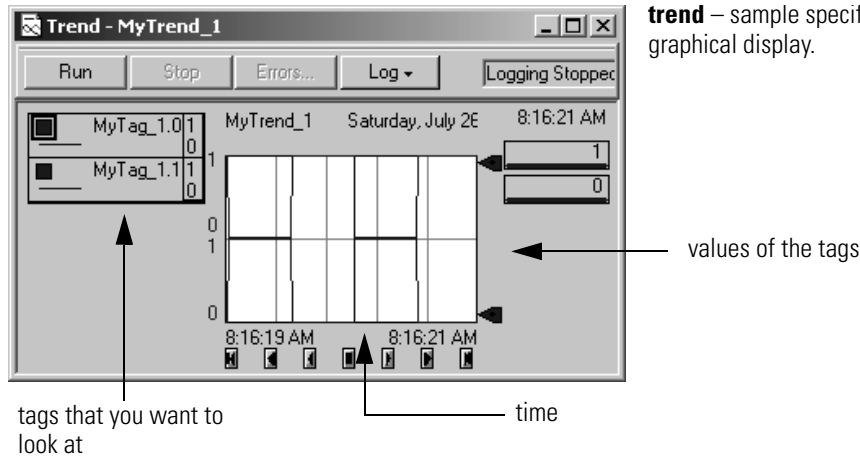
Disable All I/O Forces



1. Go online with the controller.
2. Choose *I/O Forcing* ⇒ *Disable All I/O Forces*. and choose (yes—disable I/O forces).

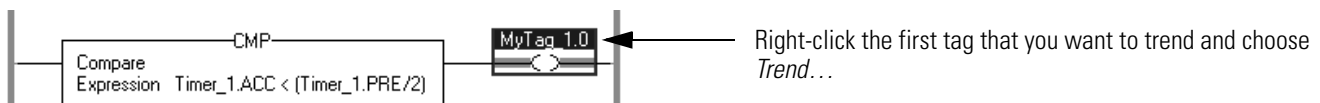
Create and Run a Trend (Histogram)

Trends let you view sampled tag data over a period of time on a graphical display. Tag data is sampled by the controller and then displayed as point(s) on a trend chart.

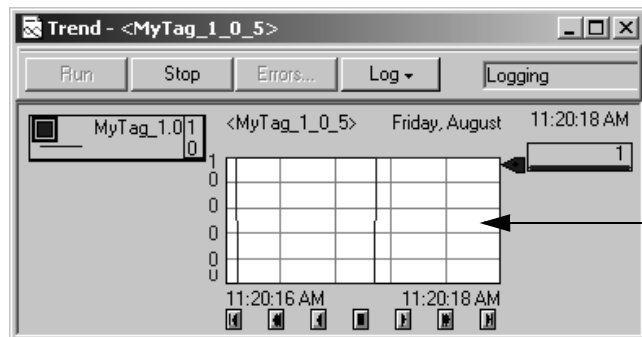


trend – sample specific tags over time and show the data on a graphical display.

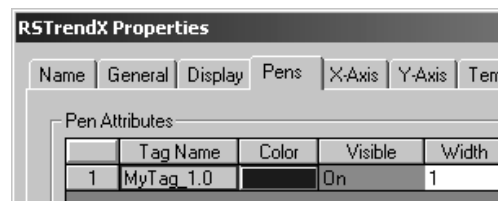
Run a Trend for a Tag



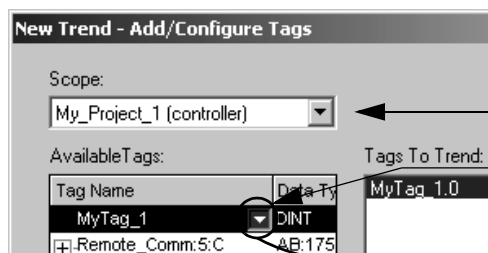
Add More Tags to the Trend



1. Right-click the chart and choose *Chart Properties*.
2. Click the *Pens* tab.



3. Choose **Add/Configure Tags**
4. Select a tag to add and choose **Add -->**



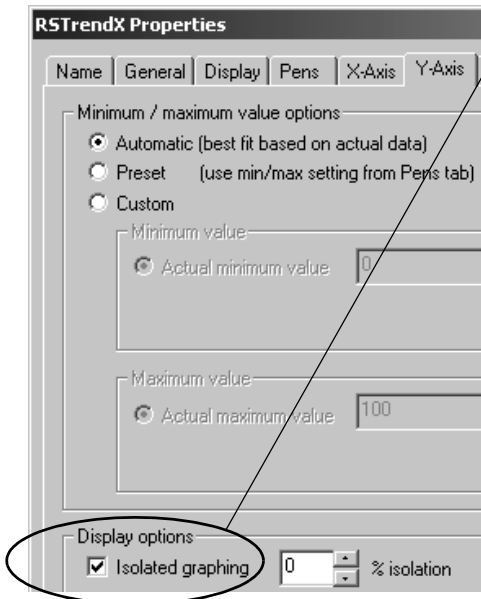
- To change the scope, select a scope.
- To select a bit number, click the ▼.

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31

5. When you have added the required tags, choose **OK**

Add More Tags to the Trend (continued)

6. Click the Y-Axis tab.

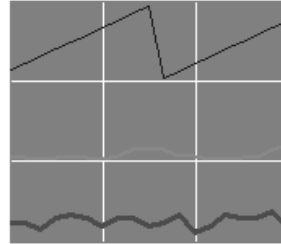


7. Choose the type of graphing.

isolated graphing

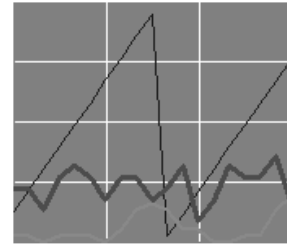
Plots each pen in a separate band of a TrendX chart.

Isolated graphing



☒ Isolated graphing

Non isolated graphing



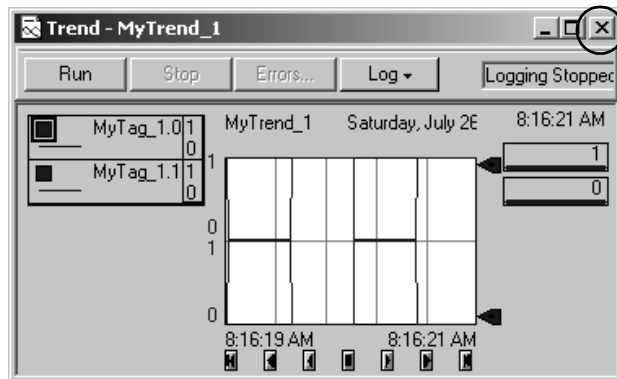
☐ Isolated graphing

► Select isolated or non-isolated graphing on the TrendX dialog box - Y-Axis tab.

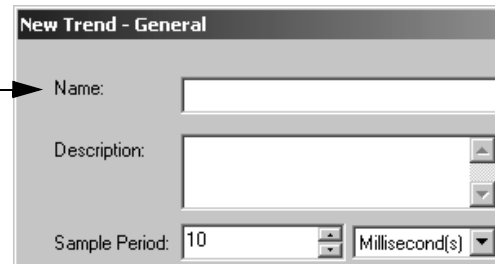
8. Choose **OK**

9. To resume the trend, choose **Run**

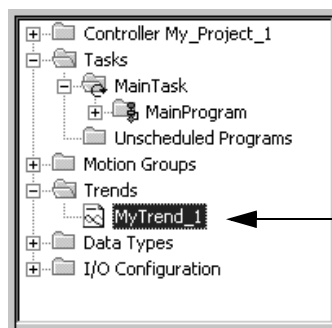
Optional—Save the Trend



1. When you close the trend, you have the option save the trend for future use.



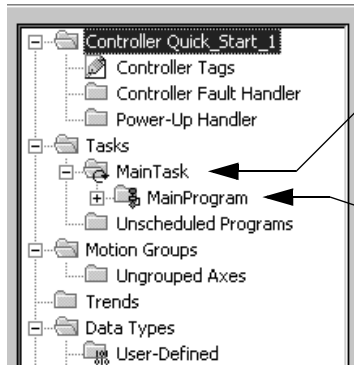
2. Type a name for the trend and choose **Finish**



trend

View Scan Time

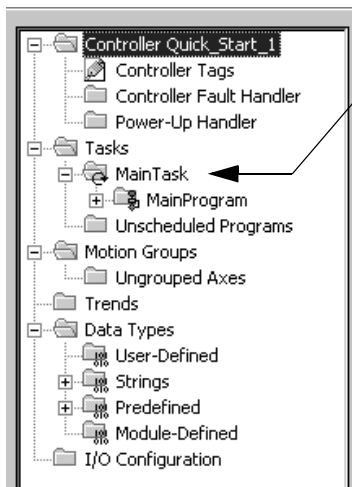
A Logix5000 controller provides two types of scan times. Each serves a different purpose:



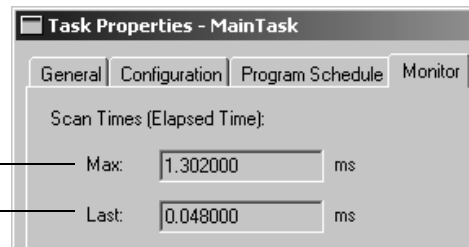
elapsed time (task scan time) – time that has elapsed from the start of a task to the end of the task, in milliseconds. The elapsed time of a task includes the time that the task is interrupted to service communications or other tasks.

execution time (program scan time) –time to execute the logic of a program (its main routine and any subroutines that the main routine calls), in microseconds. The scan time of a program includes only the execution time of the logic. It *does not* include any interrupts.

View Task Scan Time

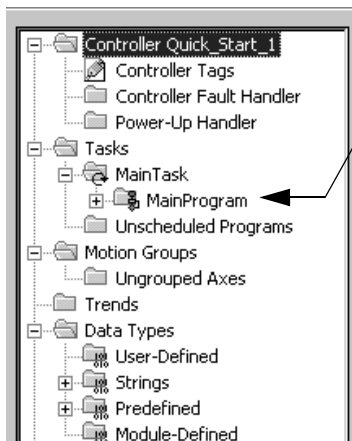


1. Right-click and choose *Properties*.
2. Click the *Monitor* tab.

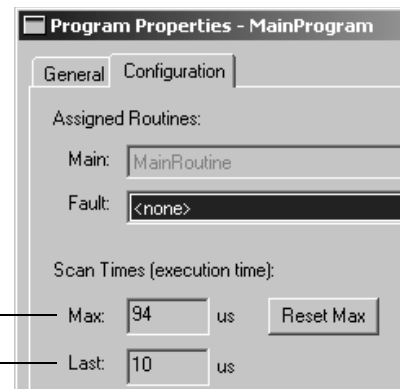


elapsed time of the last execution of this task
maximum elapsed time of the task

View Program Scan Time



1. Right-click and choose *Properties*.
2. Click the *Configuration* tab.



maximum execution time of this program
execution time of the last execution of this program

Notes:

A**alias tags**

use 1-13

array

create 2-5
organize 3-22
use of 2-5

ASCII text

enter logic using 3-2

assume data available indicator

use of 3-9

B**BOOTP**

use of 5-2

browse

logic 7-7

C**clear**

major fault 7-4

comment

add to function block diagram 4-7
add to rung 4-4, 4-5
add to SFC 4-7
add to structured text 4-9
search for 7-5

communicate

with controller via EtherNet/IP network 5-2
with controller via serial cable 1-15

communication

fault 7-2

configure

controller 1-2, 2-11
driver for EtherNet/IP communication 5-2
driver for serial communication 1-15
I/O module 1-4, 2-12
task 2-2
trend 7-11

continous task

execution 2-2

controller

communicate via EtherNet/IP network 5-2
communicate via serial cable 1-15
configure 1-2, 2-11
download project 1-17
faulted 7-4
go online with 5-6

mode 1-19
monitor 5-8
monitor execution 5-6
revision 1-17

controller organizer

add I/O module 1-4
navigate 1-2
open routine 1-7

controller-scope tags

when to use 2-3

create

program 2-3
project 1-2
routine 2-7
sheet 3-9
text box 4-7
trend 7-11

D**data**

I/O module 1-5
trend 7-11

description

rung 4-4, 4-5
search for 7-5
tag 4-2
user-defined data type 4-2

document

function block diagram 4-7
rung 4-4, 4-5
SFC 4-7
structured text 4-9
tag 4-2
user-defined data type 4-2

download

project 1-17

driver

configure for EtherNet/IP communication 5-2
configure for serial communication 1-15

duplicate destructive bit detection

use of 3-20

E**elapsed time**

task 7-13

enter

function block diagram 3-9
ladder logic 3-2
logic while online 6-1, 6-5
rung comment 4-4

- SFC 3-16
- structured text 3-14
 - comments 4-9

errors

- check routine for 3-20

EtherNet/IP network

- assign IP address 5-2
- communicate with controller 5-2

execution

- choose controller mode 1-19
- task 2-2
- time 7-13

export

- ladder logic 3-6
- rung comment 4-5

F**faceplate**

- add 3-12

fault

- controller 7-4
- I/O module 7-2

file

- See array

finalize all edits in program 6-5**find**

- See search

firmware

- update during download 1-17

force

- I/O value 7-8

function block diagram

- create sheet 3-9
- document 4-7
- edit online 6-1, 6-5
- enter 3-9
- resolve loop 3-9
- use for 2-7

function block instruction

- use of faceplate 3-12

H**histogram**

- See trend

I**I/O device**

- access data 1-5

I/O module

- add to project 1-4

- address format 1-5
- communication failure 7-2
- configure 1-4, 2-12
- faulted 7-2
- force value 7-8

import

- ladder logic 3-6
- rung comment 4-5

instruction

- search for 7-5

IP address

- assign to module 5-2

L**ladder logic**

- add rung comment 4-4, 4-5
- edit online 6-1, 6-5
- enter 3-2
- export 3-6
- import 3-6
- use for 2-7
- use of quick keys 3-2

library of logic

- create and use 3-6

logic

- check for errors 3-20
- edit online 6-1, 6-5

M**main routine**

- assign 2-10
- use of 2-7

major fault

- clear 7-4

mode

- controller 1-19

monitor

- controller 5-6
- project in controller 5-8

N**name**

- guidelines for tag 3-22
- limitations 1-2

O**online**

- edit logic 6-1, 6-5
- finalize all edits 6-5

with controller 5-6

open

routine 1-7

operand

assign 3-18

P

pass-through description 4-2

period

define for task 2-2

periodic task

execution 2-2

program

assign main routine 2-10

create 2-3

finalize all edits 6-5

scan time 7-13

program mode 1-19

programming language

choose 2-7

RSLogix 5000 software 2-8

program-scope tags

when to use 2-3

project

create 1-2

download 1-17

monitor in controller 5-6

organize routines 2-7

upload 5-8

verify 3-20

Q

quick keys

enter ladder logic 3-2

R

revision

controller firmware 1-17

routine

check for errors 3-20

create 2-7

create tag 3-18

edit logic online 6-1, 6-5

import ladder logic 3-6

open 1-7

organize 2-7

program ladder logic 3-2

program using a function block diagram
3-9

program using an SFC 3-16

program using structured text 3-14

run mode 1-19

rung comment

add 4-4, 4-5

export/import 4-5

S

scan time

view 7-13

scope

choose for tag 2-3

guidelines 3-22

search

browse 7-7

comments or descriptions 7-5

instruction 7-5

tag 7-5

sequential function chart

See SFC

serial communication

with controller 1-15

SFC

document 4-7

edit online 6-1, 6-5

enter 3-16

use for 2-7

sheet

use of 3-9

source protection

use of 1-7

structure

create 2-5

organize 3-22

structured text

document 4-9

edit online 6-1, 6-5

enter 3-14

use for 2-7

subroutine

See routine

T

tag

create 3-18

description 4-2

force value 7-8

format 3-18

guidelines 3-22

I/O module 1-5

organize 2-5, 3-22

reuse of names 2-3

- scope 2-3
- search for 7-5
- trend value 7-11

task

- configure 2-2
- scan time 7-13

test mode 1-19

text box

- add to function block diagram 4-7
- add to SFC 4-7

trend

- create and run 7-11

troubleshoot

- check wiring to output device 7-8
- communication with I/O module 7-2
- entire system is shut down 7-4
- override logic 7-8

- see data history 7-11
- several devices not responding 7-2

U

update

- controller firmware 1-17

upload

- project 5-8

user-defined data type

- create 2-5
- use of 2-5

V

verify

- project 3-20



How Are We Doing?

Your comments on our technical publications will help us serve you better in the future.
Thank you for taking the time to provide us feedback.

You can complete this form and mail (or fax) it back to us or email us at
RADocumentComments@ra.rockwell.com

Pub. Title/Type Logix5000™ Controllers Quick Start

Cat. No. 1756 ControlLogix®, 1769 CompactLogix™, 1789 SoftLogix™, 1794 FlexLogix™, PowerFlex® 700S with DriveLogix™
Pub. No. 1756-QS001B-EN-P Pub. Date March 2004 Part No. 957867-44

Please complete the sections below. Where applicable, rank the feature (1=needs improvement, 2=satisfactory, and 3=outstanding).

Overall Usefulness	1	2	3	How can we make this publication more useful for you?
Completeness (all necessary information is provided)	1	2	3	Can we add more information to help you?
				procedure/step illustration feature
				example guideline other
				explanation definition
Technical Accuracy (all provided information is correct)	1	2	3	Can we be more accurate?
				text illustration
Clarity (all provided information is easy to understand)	1	2	3	How can we make things clearer?
Other Comments				You can add additional comments on the back of this form.

Your Name _____
Your Title/Function _____
Location/Phone _____

Would you like us to contact you regarding your comments?
___ No, there is no need to contact me
___ Yes, please call me
___ Yes, please email me at _____
___ Yes, please contact me via _____

Return this form to: Rockwell Automation Technical Communications, 1 Allen-Bradley Dr., Mayfield Hts., OH 44124-9705
Fax: 440-646-3525 Email: RADocumentComments@ra.rockwell.com

PLEASE FASTEN HERE (DO NOT STAPLE)

Other Comments

PLEASE FOLD HERE

PLEASE REMOVE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 18235 CLEVELAND OH

POSTAGE WILL BE PAID BY THE ADDRESSEE



**Rockwell
Automation**

1 ALLEN-BRADLEY DR
MAYFIELD HEIGHTS OH 44124-9705



Microsoft, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Rockwell Automation Support

Rockwell Automation provides technical information on the web to assist you in using our products. At <http://support.rockwellautomation.com>, 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://support.rockwellautomation.com>.

Installation Assistance

If you experience a problem with a hardware module within the first 24 hours of installation, please review the information that's contained in this manual. You can also contact a special Customer Support number for initial help in getting your module up and running:

United States	1.440.646.3223 Monday – Friday, 8am – 5pm EST
Outside United States	Please contact your local Rockwell Automation representative for any technical support issues.

New Product Satisfaction Return

Rockwell tests all of our 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:

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

www.rockwellautomation.com

Corporate Headquarters

Rockwell Automation, 777 East Wisconsin Avenue, Suite 1400, Milwaukee, WI, 53202-5302 USA, Tel: (1) 414.212.5200, Fax: (1) 414.212.5201

Headquarters for Allen-Bradley Products, Rockwell Software Products and Global Manufacturing Solutions

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444

Europe: Rockwell Automation SA/NV, Vorstlaan/Boulevard du Souverain 36-BP 3A/B, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

Asia Pacific: Rockwell Automation, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Headquarters for Dodge and Reliance Electric Products

Americas: Rockwell Automation, 6040 Ponders Court, Greenville, SC 29615-4617 USA, Tel: (1) 864.297.4800, Fax: (1) 864.281.2433

Europe: Rockwell Automation, Brühlstraße 22, D-74834 Elztal-Dallau, Germany, Tel: (49) 6261 9410, Fax: (49) 6261 17741

Asia Pacific: Rockwell Automation, 55 Newton Road, #11-01/02 Revenue House, Singapore 307987, Tel: (65) 351 6723, Fax: (65) 355 1733



Allen-Bradley

Logix5000™ Controllers Quick Start

Quick Start