Logix5000 Controllers

Allen-Bradley

Catalog Numbers 1756 ControlLogix, 1756 GuardLogix,
1768 CompactLogix, 1768 Compact GuardLogix,
1769 CompactLogix, 1789 SoftLogix, PowerFlex with DriveLogix

Quick Start

Allen-Bradley • Rockwell Software

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.rockwellautomation.com/literature/) describes some important differences between solid state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

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

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

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

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

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

WARNING ⚠	Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.
IMPORTANT	Identifies information that is critical for successful application and understanding of the product.
ATTENTION ⚠	Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence
SHOCK HAZARD ⚡	Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.
BURN HAZARD ⚠	Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.

Summary of Changes

This version of the quick start corresponds to revision 18 of the Logix5000 controller firmware.

Change	Page
Descriptions of controller modes	32
Language switching	108
Additional information for finalizing edits in larger projects	124

Notes:

About This Publication

Use this manual to get started programming and maintaining Logix5000 controllers.

This manual describes the necessary tasks to do the following.
- establish communication with a Logix5000 controller
- program a Logix5000 controller
- perform online maintenance tasks such a search and edit logic, run a histogram, clear faults, and force I/O values.

Required Software

To complete this quick start, the following software is required:
- RSLogix 5000 software, version 18 or later
- RSLinx Classic software, version 2.51

Additional Resources

Resource	Description
Logix5000 Controllers System Reference, publication 1756-QR107	Look up abbreviated information and procedures regarding programming languages, instructions, communications, and status
Logix5000 Controllers Design Considerations Reference, publication 1756-RM094	Design and optimize a controller application.
Logix5000 Controllers Common Procedures, publication 1756-PM001	Program a Logix5000 controller—detailed and comprehensive information
• Logix5000 Controllers General Instructions Reference Manual, publication 1756-RM003 • Logix5000 Controllers Process and Drives Instructions Reference Manual, publication 1756-RM006 • Logix5000 Controllers Motion Instruction Set Reference Manual, publication MOTION-RM001	Program a specific Logix5000 programming instruction
Logix5000 Controllers Import/Export Reference Manual, publication 1756-RM084	Import or export a Logix5000 project or tags from or to a text file
• 1768 CompactLogix Controller Quick Start and User Manual, publication 1768-UM001 • 1769 CompactLogix System User Manual, publication 1769-UM007 • ControlLogix System User Manual, publication 1756-UM001 • DriveLogix Controller User Manual, publication 20D-UM002 • GuardLogix Controllers User Manual, publication 1756-UM020 • SoftLogix5800 System User Manual, publication 1789-UM002	Integrate a specific Logix5000 controller within a system of controllers, I/O modules, and other devices
EtherNet/IP Modules in Logix5000 Control Systems User Manual, publication ENET-UM001	Control devices over an EtherNet/IP network
ControlNet Modules in Logix5000 Control Systems User Manual, publication CNET-UM001	Control devices over a ControlNet network
DeviceNet Modules in Logix5000 Control Systems User Manual, publication DNET-UM004	Control devices over a DeviceNet network

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

Program and Test a Simple Project

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.

What You Need

You need these items to complete the tasks in this manual:

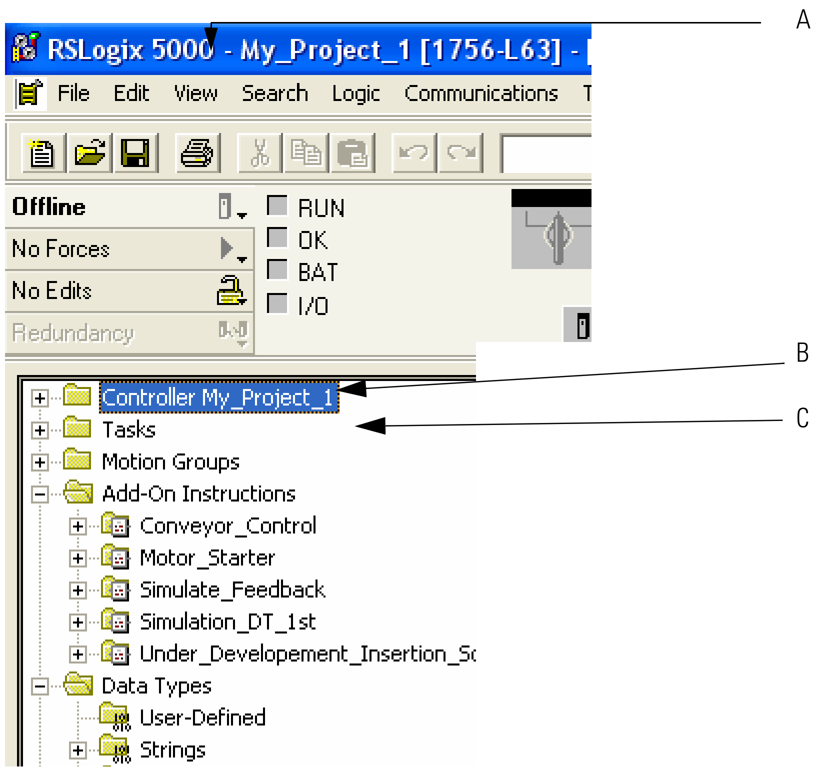
- Personal computer running RSLogix 5000 software, version 16 or later
- A layout of the system for which you are creating a project

Before You Begin

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

In an open project, there is this information:



Item	Description
A	Name of the project. If you rename the project or controller, both names are shown.
B	Name of the controller.
C	The controller organizer is a 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:

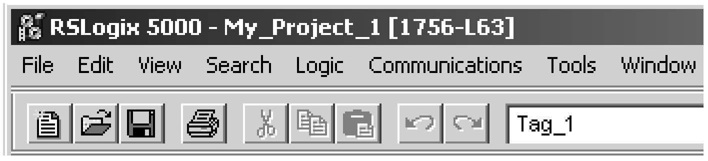
- double-click the folder.
- click the + sign.

Follow These Steps

1. Create a project for the controller (page 14).
2. Add I/O modules (page 15).
3. Look at I/O data (page 17).
4. Enter ladder logic (page 19).
5. Enter a function block diagram (page 21).
6. Assign alias tags for your devices (page 25).
7. Establish a serial connection to the controller (page 28).
8. Download a project to the controller (page 30).
9. Select the operating mode of the controller (page 32).

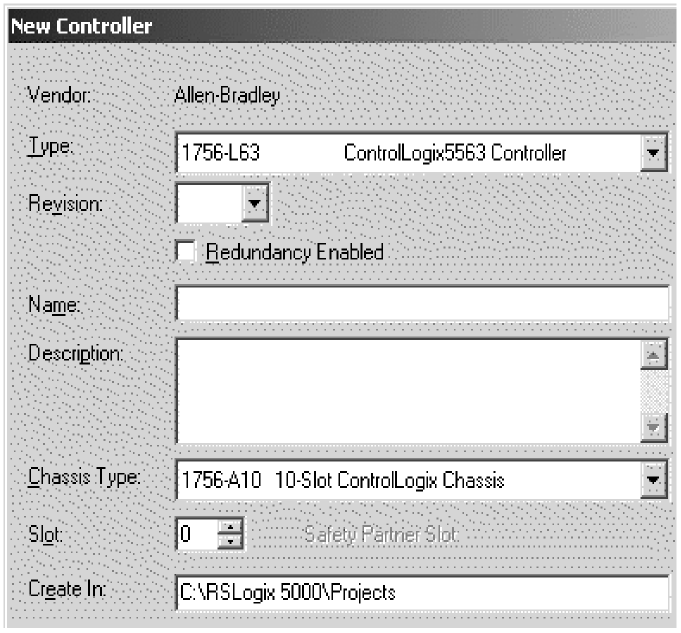
Create a Project for the Controller

1. Start RSLogix 5000 software.


RSLogix 5000 - My_Project_1 [1756-L63]
File Edit View Search Logic Communications Tools Window
Tag_1

2. Click New.

3. Specify the general configuration for the controller.



New Controller

Vendor:	Allen-Bradley
Type:	1756-L63 ControlLogix5563 Controller
Revision:	
	☐ Redundancy Enabled
Name:	
Description:	
Chassis Type:	1756-A10 10-Slot ControlLogix Chassis
Slot:	0 Safety Partner Slot:
Create In:	C:\RSLogix 5000\Projects

Specify these items (some items apply to only certain controllers):
- Type of controller.
- Major revision of firmware for the controller.
- Name for the controller.
- Chassis type for the controller.
- Slot number of the controller.
- The path where the project will be stored.

4. Click 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, and I/O modules. 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

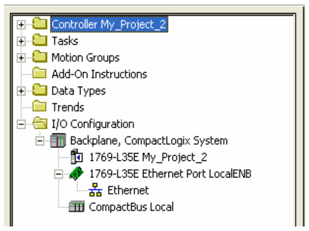
Add Your I/O Modules

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.

TIP The screens shown are representative of three types of controllers; other types are available, but are not shown here.

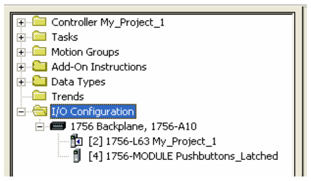
1. Right-click the I/O Configuration folder and choose New Module.

CompactLogix Controller



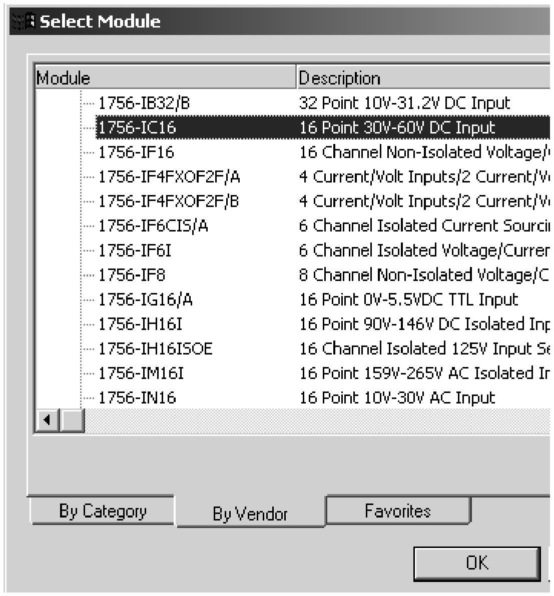
```
Controller My_Project_2
Tasks
Motion Groups
Add-On Instructions
Data Types
Trends
I/O Configuration
  Backplane, CompactLogix System
    1769-L35E My_Project_2
    1769-L35E Ethernet Port LocalENB
      Ethernet
    CompactBus Local
```

ControlLogix Controller



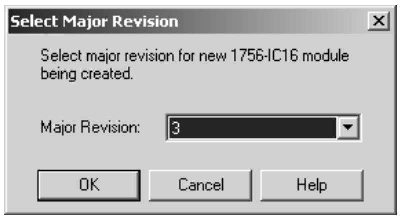
```
Controller My_Project_1
Tasks
Motion Groups
Add-On Instructions
Data Types
Trends
I/O Configuration
  1756 Backplane, 1756-A10
    [2] 1756-L63 My_Project_1
    [4] 1756-MODULE Pushbuttons_Latched
```

2. Select the module and click OK.



Module	Description
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/
1756-IF4FXOF2F/A	4 Current/Volt Inputs/2 Current/V
1756-IF4FXOF2F/B	4 Current/Volt Inputs/2 Current/V
1756-IF6CIS/A	6 Channel Isolated Current Sourci
1756-IF6I	6 Channel Isolated Voltage/Currer
1756-IF8	8 Channel Non-Isolated Voltage/C
1756-IG16/A	16 Point 0V-5.5VDC TTL Input
1756-IH16I	16 Point 90V-146V DC Isolated Inp
1756-IH16ISOE	16 Channel Isolated 125V Input Se
1756-IM16I	16 Point 159V-265V AC Isolated Ir
1756-IN16	16 Point 10V-30V AC Input

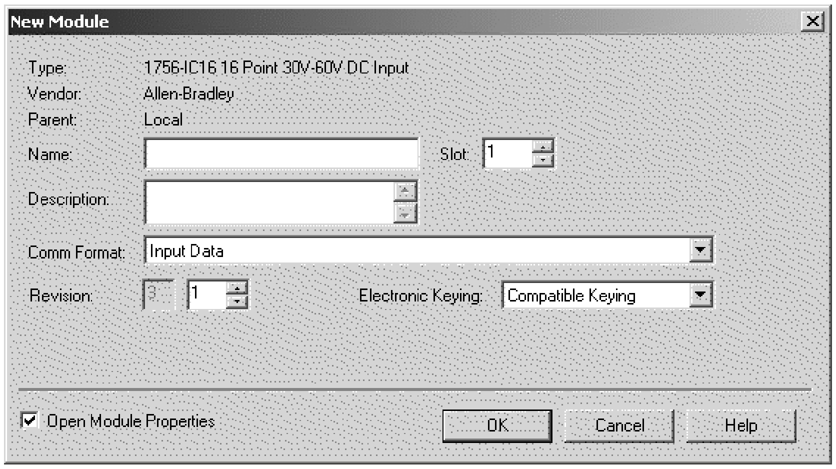
3. From the Major Revision pull-down menu, choose the revision of the module.



Select major revision for new 1756-IC16 module being created.

Major Revision: 3

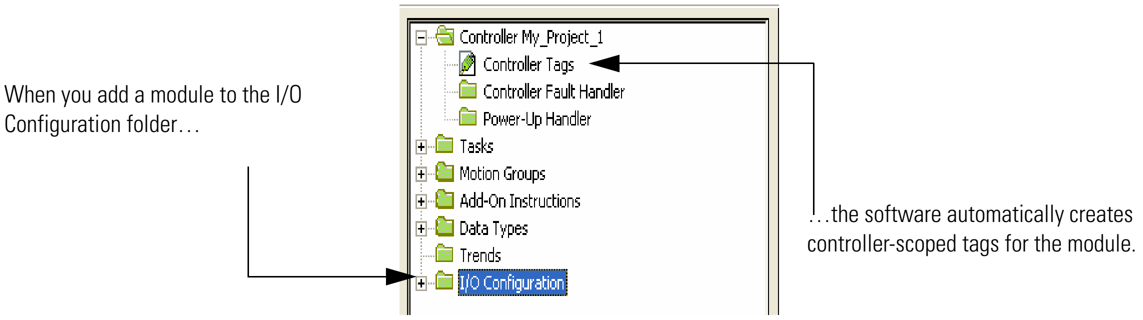
4. Define the module and click OK.



Type: 1756-IC16 16 Point 30V-60V DC Input
Vendor: Allen-Bradley
Parent: Local
Name:
Slot: 1
Description:
Comm Format: Input Data
Revision: 3 1
Electronic Keying: Compatible Keying
☑ Open Module Properties

Look at Your I/O Data

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



When you add a module to the I/O Configuration folder…

…the software automatically creates controller-scoped tags for the module.

An I/O address follows this format.

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

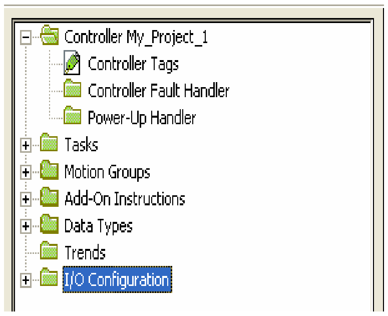
▨ = Optional

Where	Is
Location	Network location LOCAL = same chassis or DIN rail as the controller *ADAPTER_NAME* = identifies remote communication adapter or bridge module
Slot	Slot number of I/O module in its chassis or DIN rail

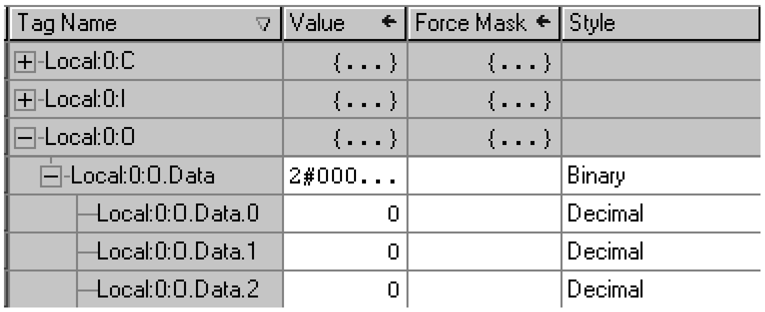
Type	Type of data
	I = input
	O = output
	C = configuration
	S = status

Member	Specific data from the I/O module; depends on what type of data the module can store. • 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.
SubMember	Specific data related to a Member.
Bit	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 Controller Tags and choose Monitor Tags.



The Tag Monitor displays the tags.



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

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.

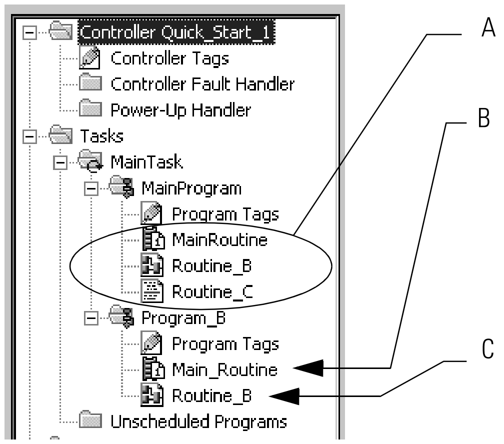
2. To see a value in a different style, select the desired style.

3. To change a value, click the Value cell, type the new value, and click Enter.

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

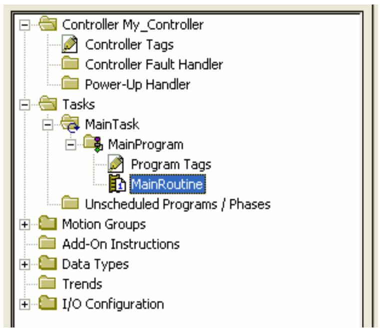
Ladder Logic

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



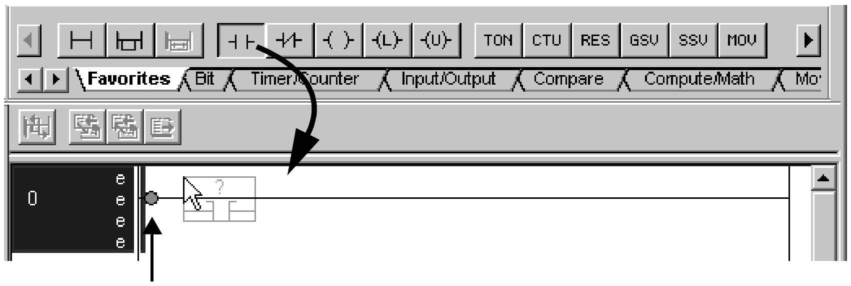
Item	Description
A	A routine provides the executable code (logic) for a program (similar to a program file in a PLC or SLC controller).
B	There is one main routine you assign for each program. • When the program executes, its main routine automatically executes. • Use the main routine to control the execution of the other routines in the program. • To call (execute) another routine (subroutine) within the program, use a Jump to Subroutine (JSR) instruction.
C	A subroutine is 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.

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



Enter Ladder Logic

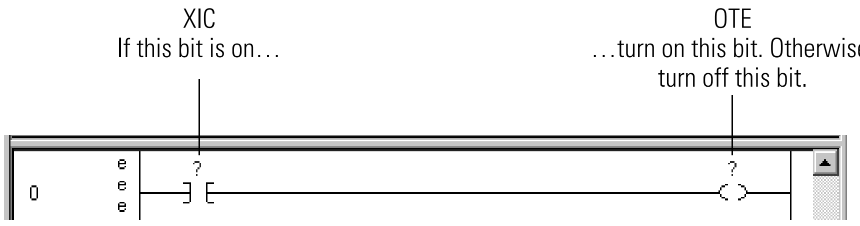
One way to enter logic is to drag buttons from a toolbar to the desired location.



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

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.

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.



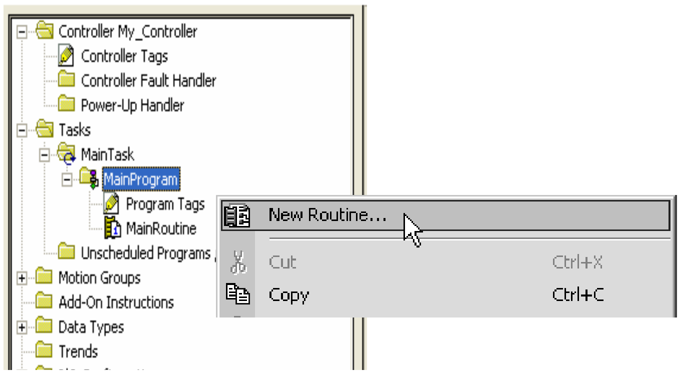
Enter a Function Block Diagram

Follow these steps to add a function block diagram to your project.

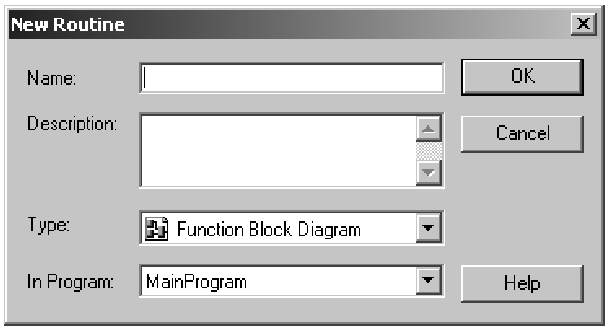
Create a Routine

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.

1. Right-click MainProgram and choose New Routine.



2. Type a name for the routine.



Type: Function Block Diagram

In Program: MainProgram

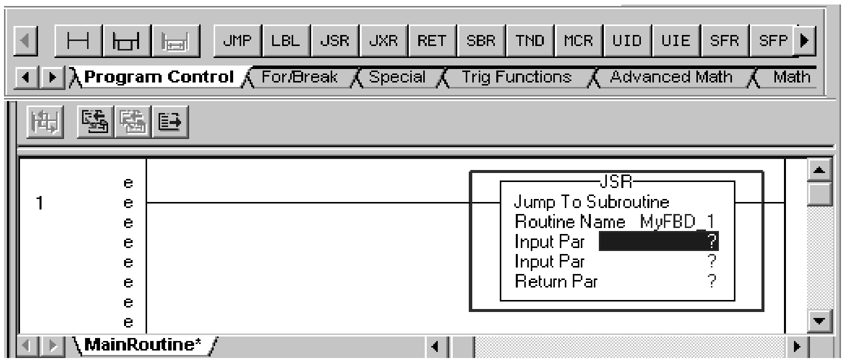
3. Choose the programming language.

4. Click OK.

Call the Routine

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

1. Add a rung.

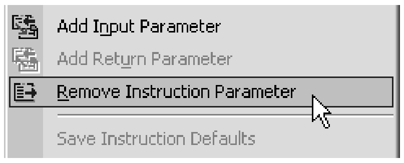


```
JSR
Jump To Subroutine
Routine Name   MyFBD_1
Input Par            ?
Input Par            ?
Return Par           ?
```

2. On the Program Control tab, add a JSR instruction.

3. In the Routine Name field of the JSR instruction, type the name of the routine that you want to execute.

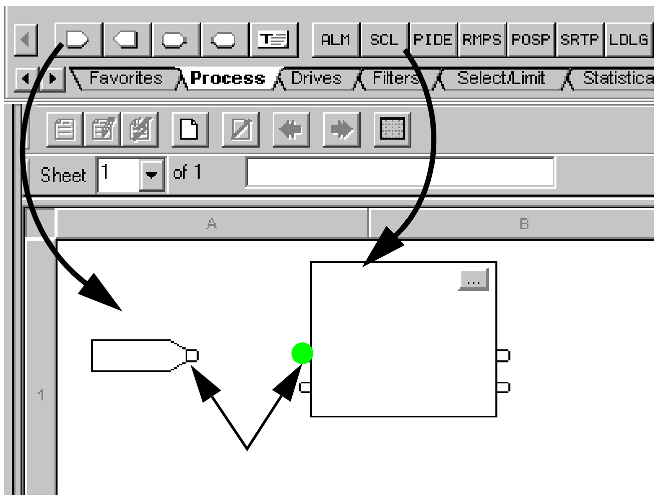
4. To simply call the routine, remove the rest of the parameters for the JSR instruction. To remove a parameter, right-click the parameter and choose Remove Instruction Parameter.



Enter a Function Block Diagram

Enter function block diagram instructions in a function block routine.

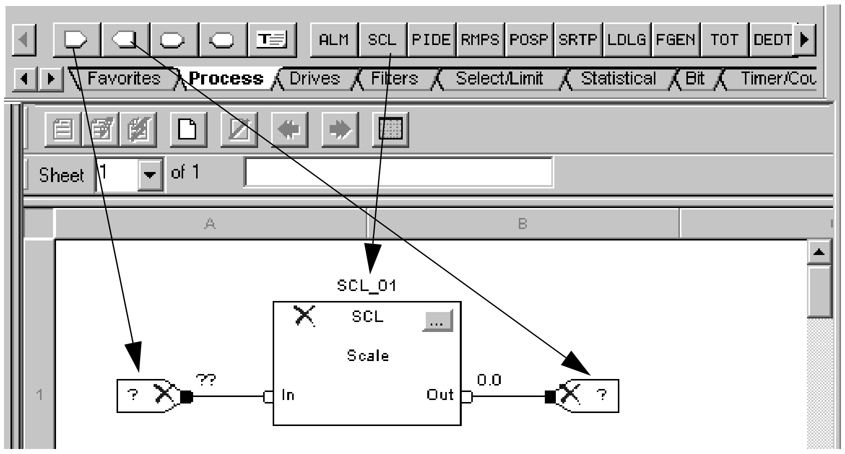
1. Click the tab for the desired instructions.



2. Drag elements from the toolbar to the sheet.
3. To connect elements, click corresponding pins (green dot = valid connection point).

EXAMPLE

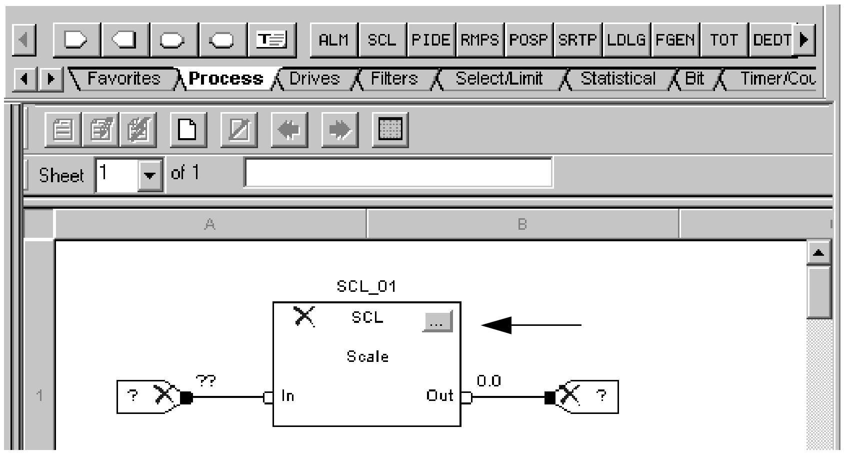
In the following example, an Input Reference (IREF) reads the value of an analog input and sends the value to a Scale (SCL) instruction. The SCL instruction converts the value to engineering uses and sends it to an Output Reference (OREF). The OREF writes the value to an analog output.



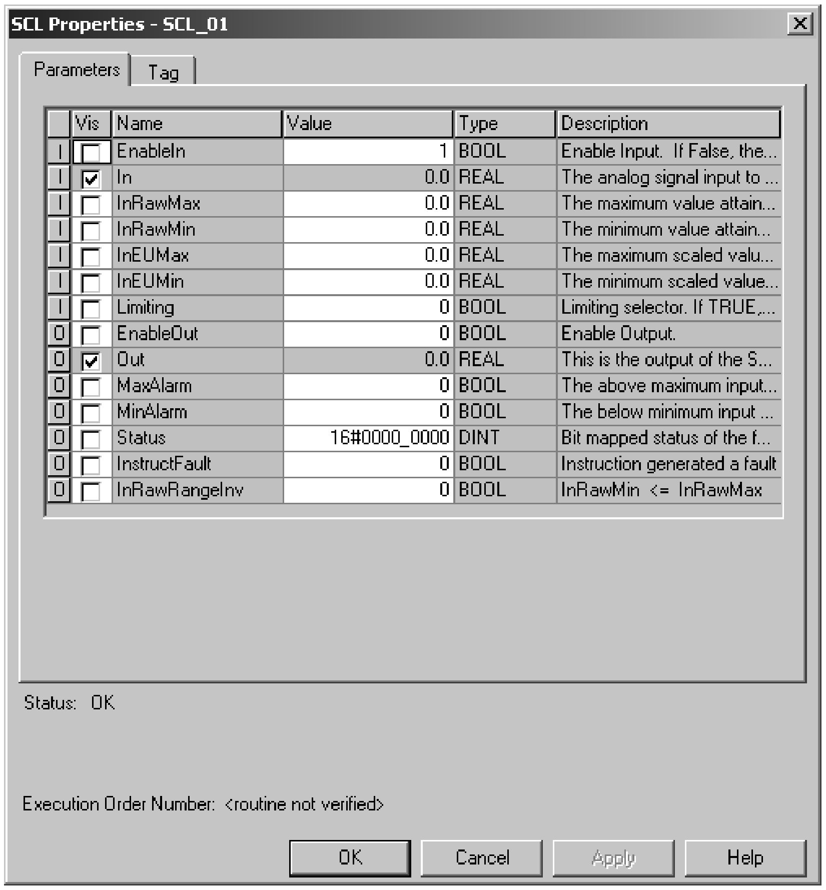
Configure a Function Block Instruction

Assign specific values (parameters) to configure a function block instruction.

1. Click the configuration button.



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



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

Status: OK

Execution Order Number: <routine not verified>

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. Click OK.

Assign Alias Tags for Your Devices

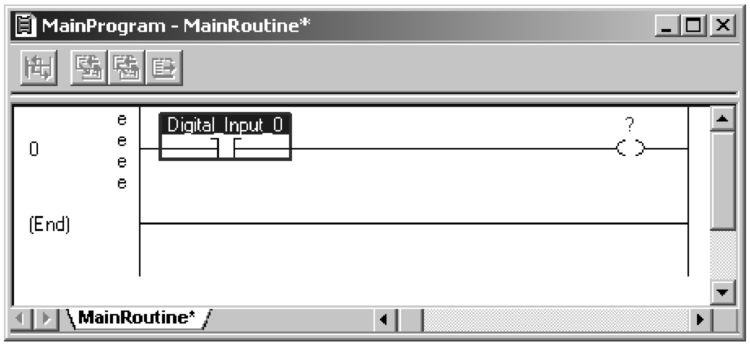
While you can use the input and output tags of a module directly in your logic, it is easier to use alias tags. An alias tag is a tag that represents another tag.

- 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 goes to the new location without editing your logic.

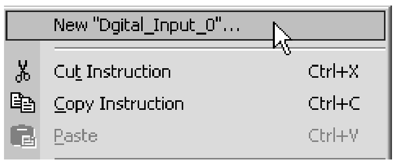
As an option, create tags that describe each device without pointing them to the actual addresses of the devices. Later, convert the tags to aliases for the data of the devices.

1. Enter the logic.

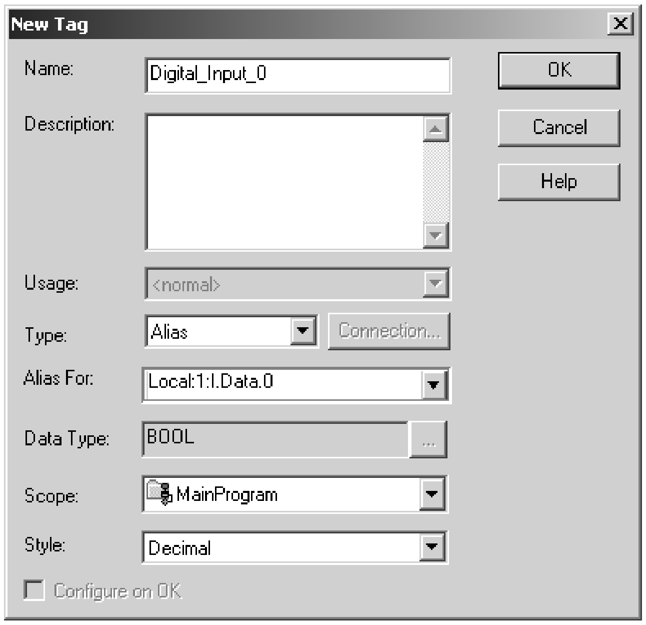
2. Type a descriptive tag name for the device.



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



4. Select Alias from the menu.



New Tag

Name: Digital_Input_0

Description:

Usage: <normal>

Type: Alias

Alias For: Local:1:I.Data.0

Data Type: BOOL

Scope: MainProgram

Style: Decimal

Configure on OK

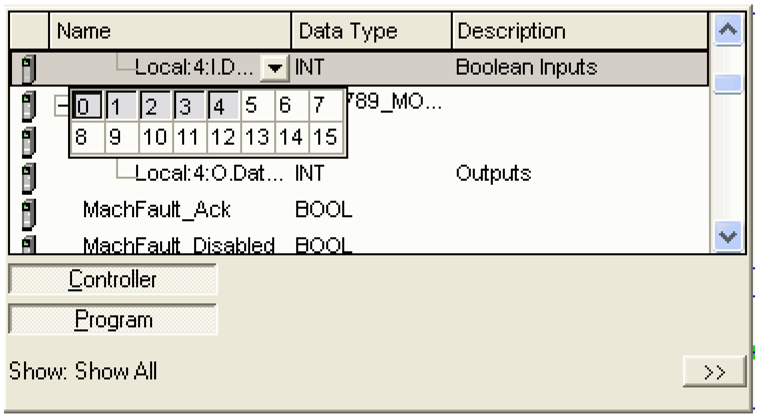
5. Select the tag that this alias tag represents.

6. Select the scope for the alias tag.

7. Click OK.

8. Select the address of the data.

   To select a bit, click the ▼ button.



Name	Data Type	Description
Local:4:I.D...	INT	Boolean Inputs
...789_MO...		
Local:4:O.Dat...	INT	Outputs
MachFault_Ack	BOOL	
MachFault_Disabled	BOOL	

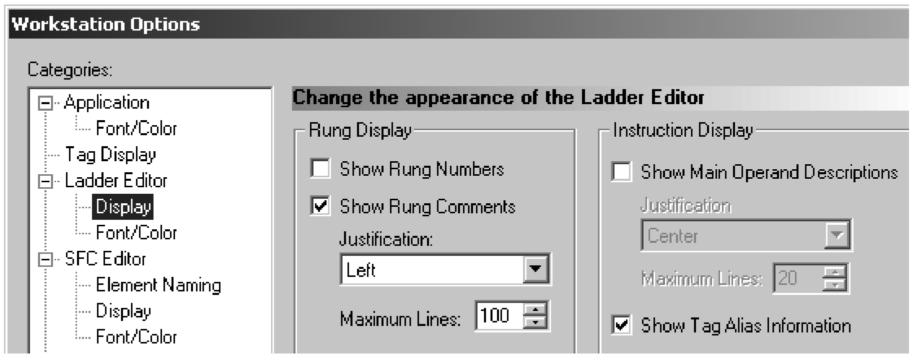
0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15

Show or Hide Alias Information

Show or hide alias information for a tag.

1. From the Tools menu, choose Options.

2. Select the Ladder Editor Display category.



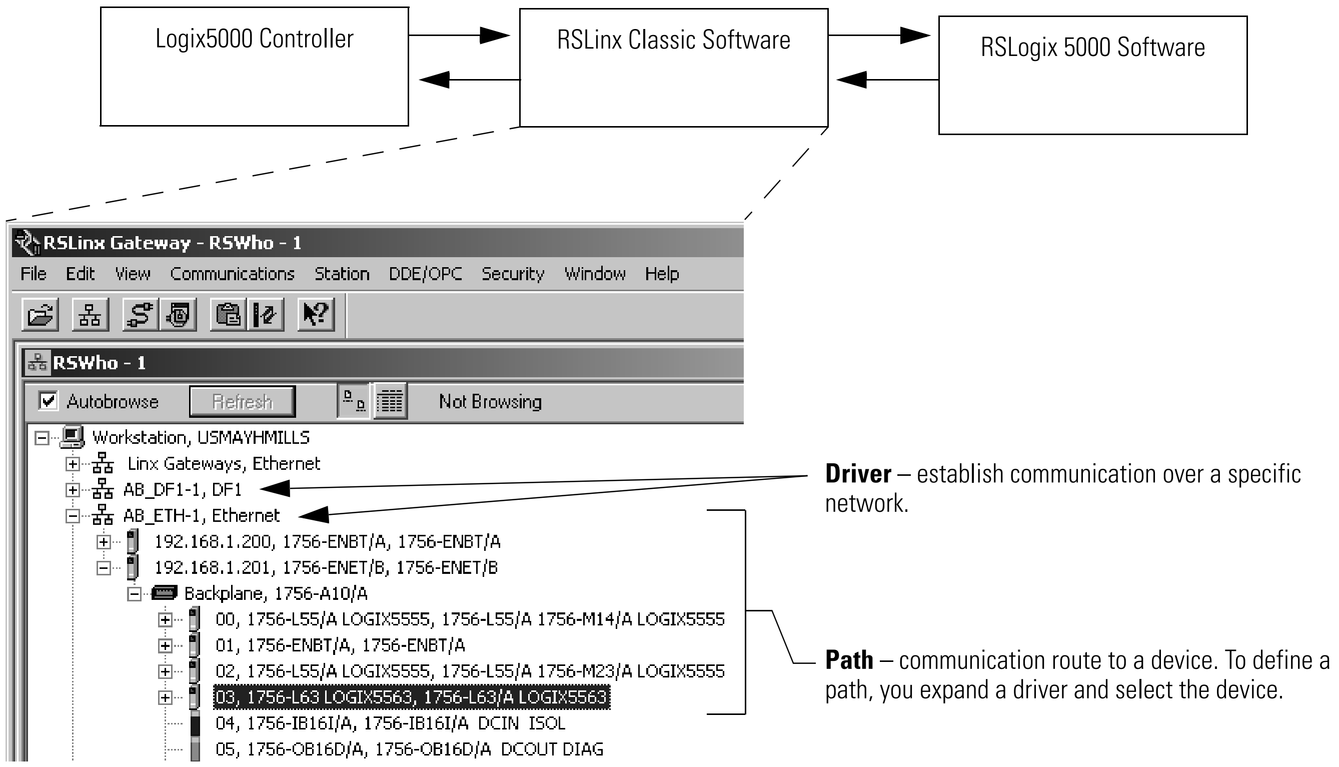
[ ] Show Rung Numbers
[✓] Show Rung Comments
Justification: Left
Maximum Lines: 100
[ ] Show Main Operand Descriptions
Justification: Center
Maximum Lines: 20
[✓] Show Tag Alias Information

3. Check or uncheck Show Tag Alias Information.

4. Click OK.

Establish a Serial Connection to the Controller

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



```
Workstation, USMAYHMILLS
  Linx Gateways, Ethernet
  AB_DF1-1, DF1
  AB_ETH-1, Ethernet
    192.168.1.200, 1756-ENBT/A, 1756-ENBT/A
    192.168.1.201, 1756-ENET/B, 1756-ENET/B
      Backplane, 1756-A10/A
        00, 1756-L55/A LOGIX5555, 1756-L55/A 1756-M14/A LOGIX5555
        01, 1756-ENBT/A, 1756-ENBT/A
        02, 1756-L55/A LOGIX5555, 1756-L55/A 1756-M23/A LOGIX5555
        03, 1756-L63 LOGIX5563, 1756-L63/A LOGIX5563
        04, 1756-IB16I/A, 1756-IB16I/A  DCIN  ISOL
        05, 1756-OB16D/A, 1756-OB16D/A  DCOUT DIAG
```

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.

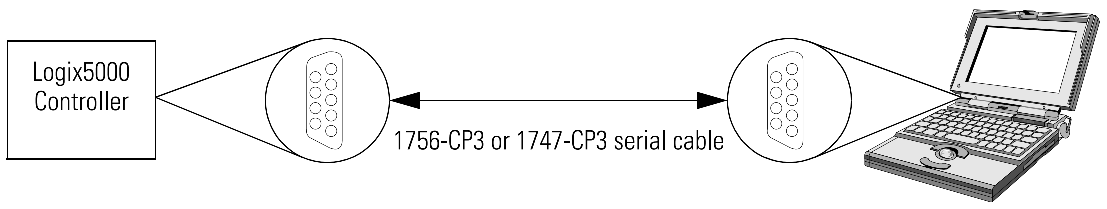
Use a serial cable to establish a point-to-point connection between the serial ports on your computer and controller.

WARNING

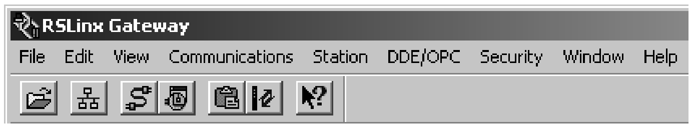
If you connect or disconnect the serial cable with power applied to this module or the serial device on the other end of the cable, an electrical arc can occur. This could cause an explosion in hazardous location installations.

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

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

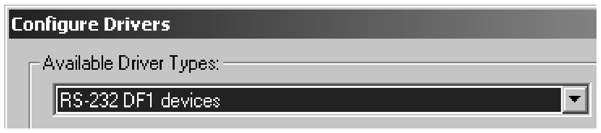


2. Start RSLinx Classic software.


RSLinx Gateway
File Edit View Communications Station DDE/OPC Security Window Help

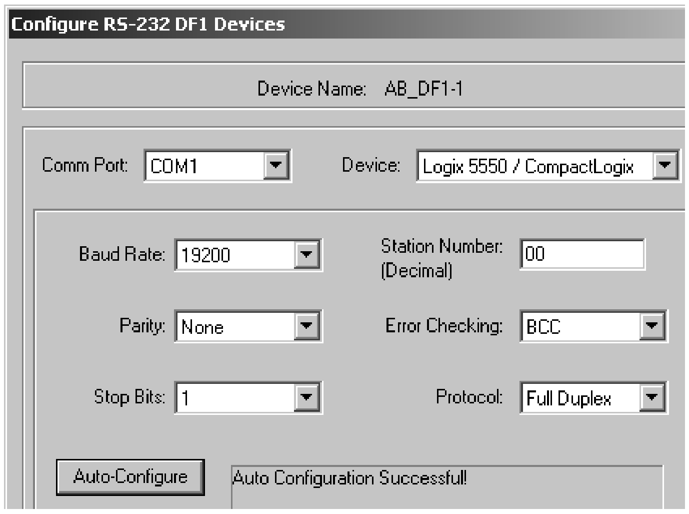
3. Click .

4. Select RS-232 DF1 devices and click Add New... .


Configure Drivers

Available Driver Types: RS-232 DF1 devices

5. From the Comm Port pull-down menu, choose the COM port of your computer.


Configure RS-232 DF1 Devices

Device Name: AB_DF1-1

Comm Port: COM1 Device: Logix 5550 / CompactLogix

Baud Rate: 19200 Station Number: (Decimal) 00

Parity: None Error Checking: BCC

Stop Bits: 1 Protocol: Full Duplex

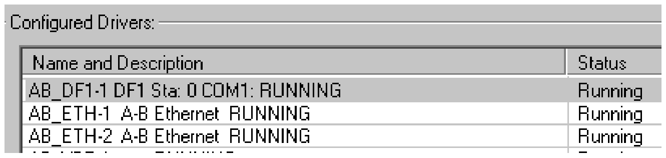
Auto-Configure Auto Configuration Successful!

6. From the Device pull-down menu, choose Logix 5550/CompactLogix.

7. Click Auto-Configure .

8. When the auto-configuration completes, click OK.

   The driver is successfully configured and running.


Configured Drivers:

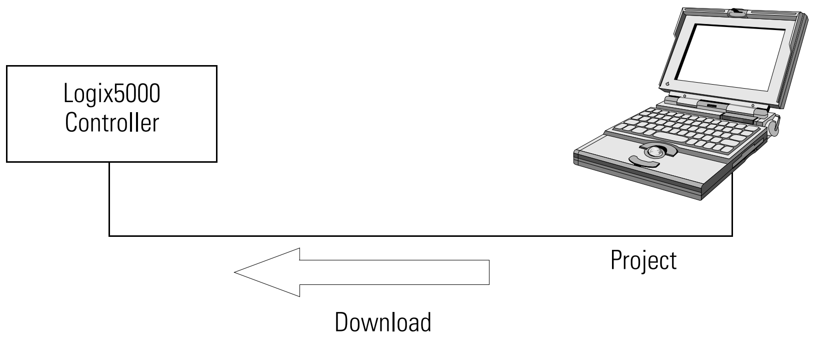
Name and Description	Status
AB_DF1-1 DF1 Sta: 0 COM1: RUNNING	Running
AB_ETH-1 A-B Ethernet RUNNING	Running
AB_ETH-2 A-B Ethernet RUNNING	Running

Download a Project to the Controller

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

ATTENTION ⚠

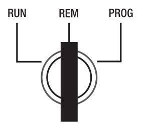
When you download a project or update firmware, all active servo axes are turned off. Before you download a project or update firmware, make sure that this will not cause any unexpected movement of an axis.



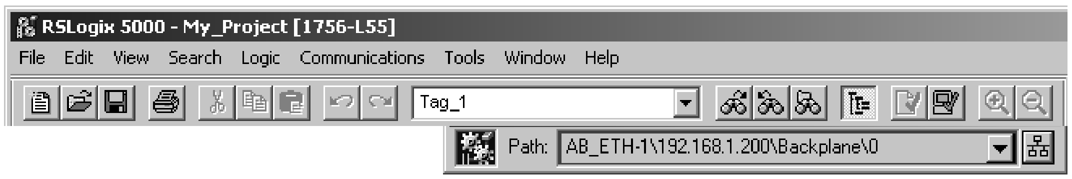
IMPORTANT

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 http://www.ab.com. Choose Product Support. Choose Firmware Updates.

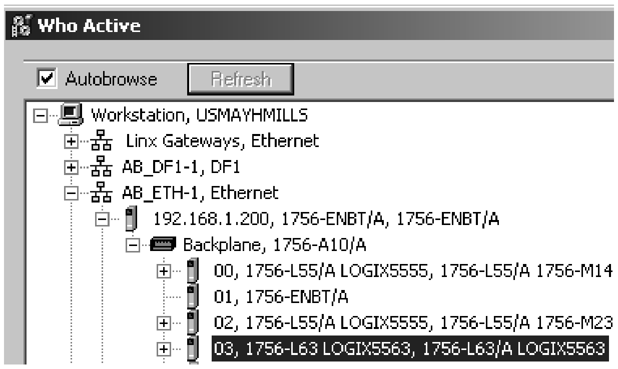
1. Turn the keyswitch of the controller to .

2. Open the RSLogix 5000 project that you want to download.


RSLogix 5000 - My_Project [1756-L55]
File Edit View Search Logic Communications Tools Window Help
Tag_1
Path: AB_ETH-1\192.168.1.200\Backplane\0

3. Click 品.

4. Browse to the controller.



Who Active

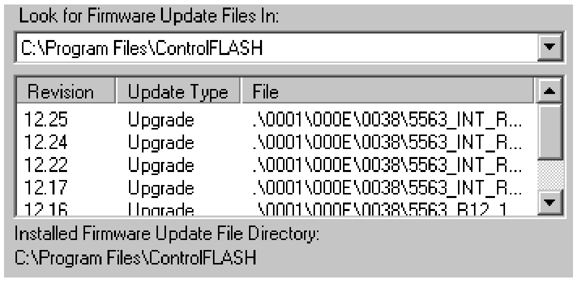
☑ Autobrowse Refresh

```
⊟ Workstation, USMAYHMILLS
  ⊞ Linx Gateways, Ethernet
  ⊞ AB_DF1-1, DF1
  ⊟ AB_ETH-1, Ethernet
    ⊟ 192.168.1.200, 1756-ENBT/A, 1756-ENBT/A
      ⊟ Backplane, 1756-A10/A
        ⊞ 00, 1756-L55/A LOGIX5555, 1756-L55/A 1756-M14
          01, 1756-ENBT/A
        ⊞ 02, 1756-L55/A LOGIX5555, 1756-L55/A 1756-M23
        ⊞ 03, 1756-L63 LOGIX5563, 1756-L63/A LOGIX5563
```

5. To download the project, click Download.

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

1. Choose Update Firmware....



Look for Firmware Update Files In:

C:\Program Files\ControlFLASH

Revision	Update Type	File
12.25	Upgrade	.\0001\000E\0038\5563_INT_R...
12.24	Upgrade	.\0001\000E\0038\5563_INT_R...
12.22	Upgrade	.\0001\000E\0038\5563_INT_R...
12.17	Upgrade	.\0001\000E\0038\5563_INT_R...
12.16	Upgrade	.\0001\000E\0038\5563_R12_1

Installed Firmware Update File Directory:
C:\Program Files\ControlFLASH

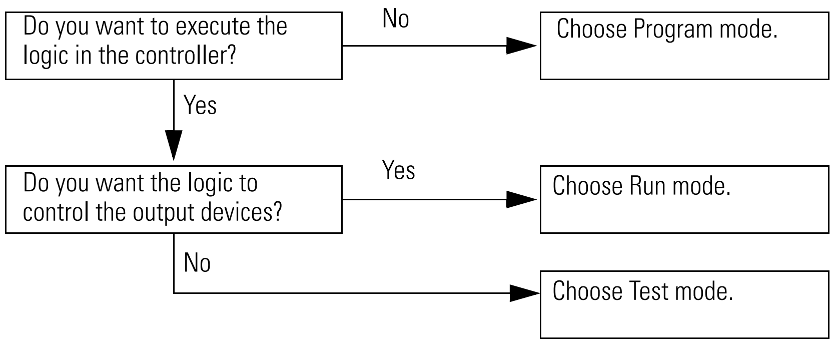
2. Select the revision for the controller.

3. Choose Update... and then Yes.

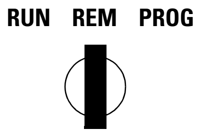
Select the Operating Mode of the Controller

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.

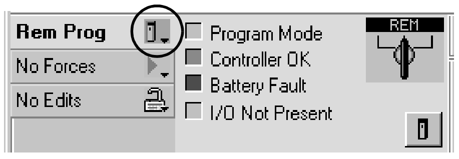

Do you want to execute the logic in the controller?
No
Choose Program mode.
Yes
Do you want the logic to control the output devices?
Yes
Choose Run mode.
No
Choose Test mode.

2. Turn the keyswitch to



3. Go online with the controller.

4. Select the mode.



Mode	Description
Program	Logic is not executing, outputs are not controlled, and editing operations are available. If you are configuring an output module, the owner controller is in Program Mode. Newly received output values are ignored and all outputs will transition to their Program mode state (which you can configure on the Configuration tab). The output module's health LED will flash green when in Program mode. Input modules are always in Run mode and always report back input data to the controller. It does not matter whether the owner controller is in Run or Program mode. The input module's health indicator is always solid green if a connection exists to it.
Run	Logic is executing, inputs are read, logic is scanned, and outputs are controlled by the application program and changes made through the data monitor or the I/O force table. The actual I/O modules accept the output results of the application and set the outputs accordingly. The keyswitch must be in the Remote or Run position.
Test	Logic is executing, inputs are read, logic is scanned, and outputs are controlled by the application program and changes made through the data monitor or the I/O force table. The actual I/O modules will ignore the output results of the application. Some editing operations are restricted. The keyswitch must be in the Remote position. When going into Test mode: • Input modules continue to update in Test mode. • Produce/consume tags continue to update in Test mode. • Test mode places all outputs in the project in the Program mode state (as configured in the Configuration tab for module properties).

Notes:

Organize a Project

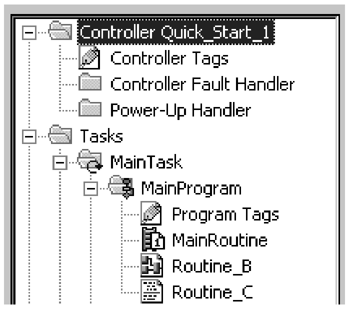
This chapter provides more detailed information on how to organize the program layout and data structures for the controller.

What You Need

You need these items to complete the tasks in this manual:
- Personal computer running RSLogix 5000 software, version 16 or later
- A layout of the system for which you are creating a project

Before You Begin

A new project contains a default task for the execution of your logic. Before you can create programs, you must first configure the task execution. A task defines scheduling and priority information for the execution (scan) of your logic.

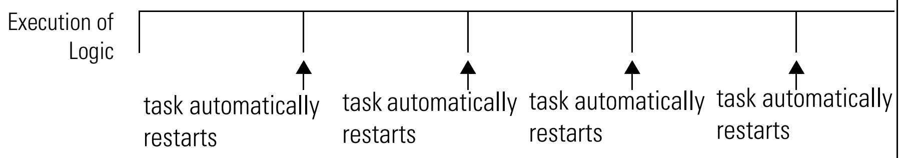
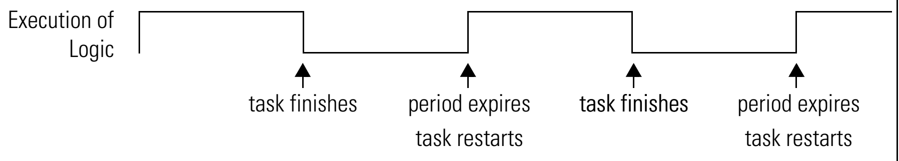


Follow These Steps

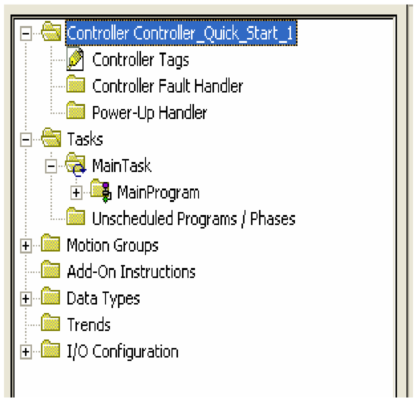
1. Configure the task execution (page 36).
2. Create additional programs (page 38).
3. Create user-defined data types (page 40).
4. Define your routines (page 43).
5. Assign main routines (page 47).
6. Configure the controller (page 48).
7. Configure I/O modules (page 49).

Configure the Task Execution

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

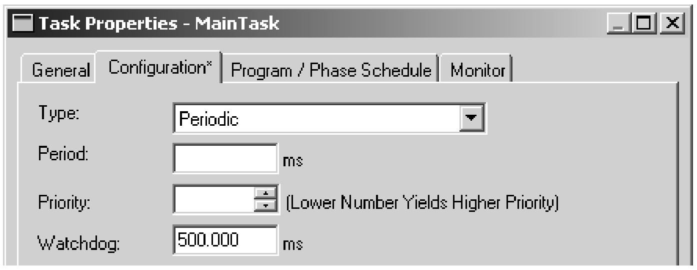
If you want to execute your logic	Then configure the task for this type of execution
All of the time  Execution of Logic task automatically restarts task automatically restarts task automatically restarts task automatically restarts	Continuous This is the default configuration of MainTask.
At a specific period  Execution of Logic task finishes period expires task restarts task finishes period expires task restarts	Periodic You define the period at which the task executes.

1. Right-click MainTask and choose Properties.



2. Click the Configuration tab.

3. From the Type pull-down menu, choose Periodic.



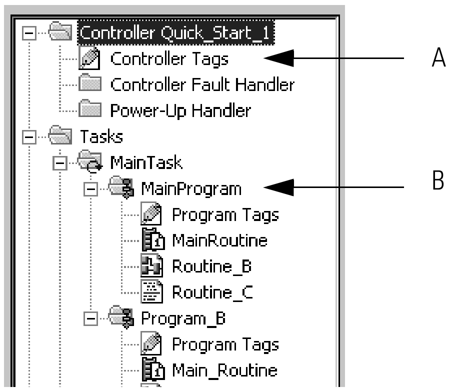
Type: Periodic
Period: ms
Priority: (Lower Number Yields Higher Priority)
Watchdog: 500.000 ms

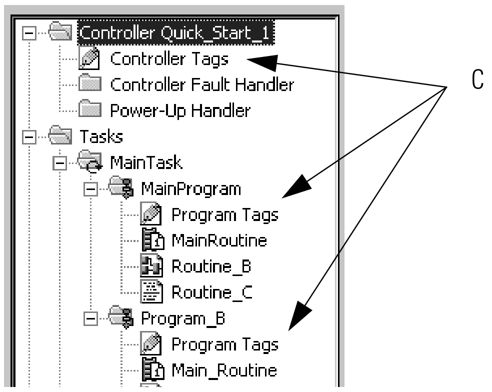
4. Type the period for the task and click OK.

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

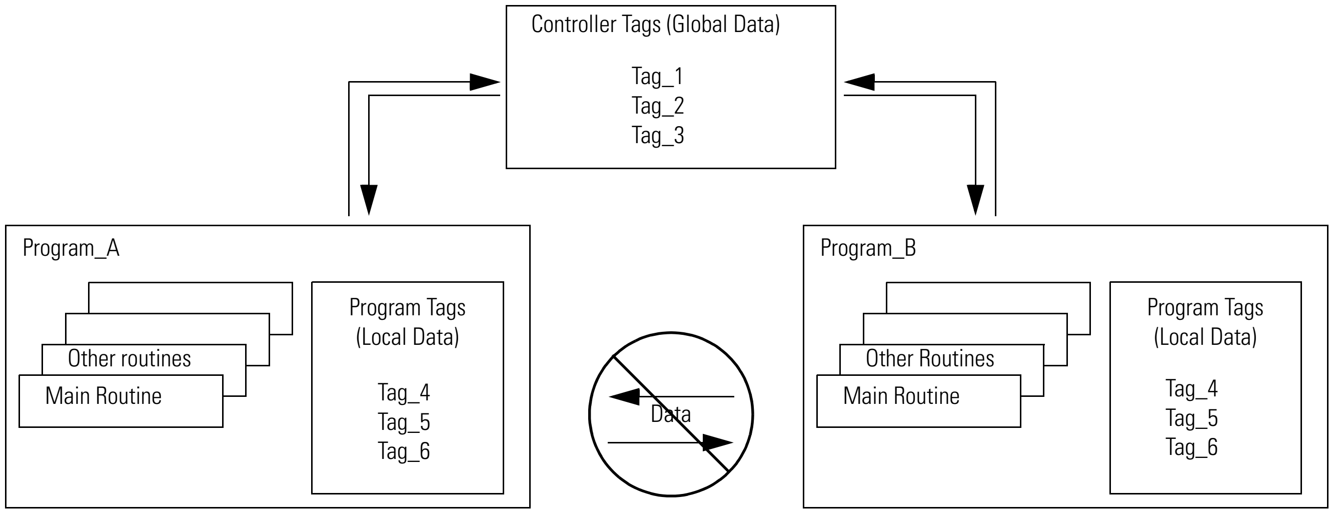
A Logix5000 controller lets you divide your application into multiple programs, each with its own tags (data).





Item	Description
A	Tag stores 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.
B	Program isolates logic and data from other logic and data. Each program contains one or more logic routines as associated data.
C	Scope defines 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.



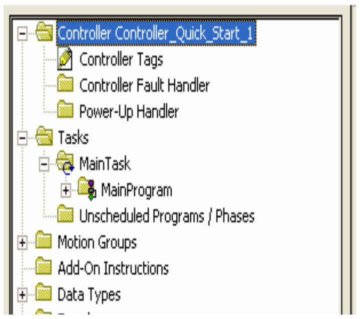
All programs have access to data that is at the controller scope. Data at the program scope is isolated from other programs.

- 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.

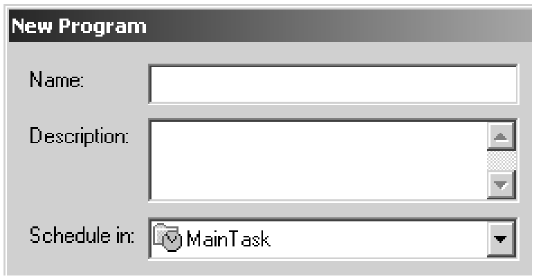
If you have multiple machines, stations, or processes that use identical logic but different data, create a program for each machine, station, or process.

- You can re-use both code and tag names in the programs.
- There is no need to manage conflicting tag names between the programs.

1. Right-click MainTask and choose New Program.



2. Type a name for the program and click OK.



TIP Names follow these conventions:

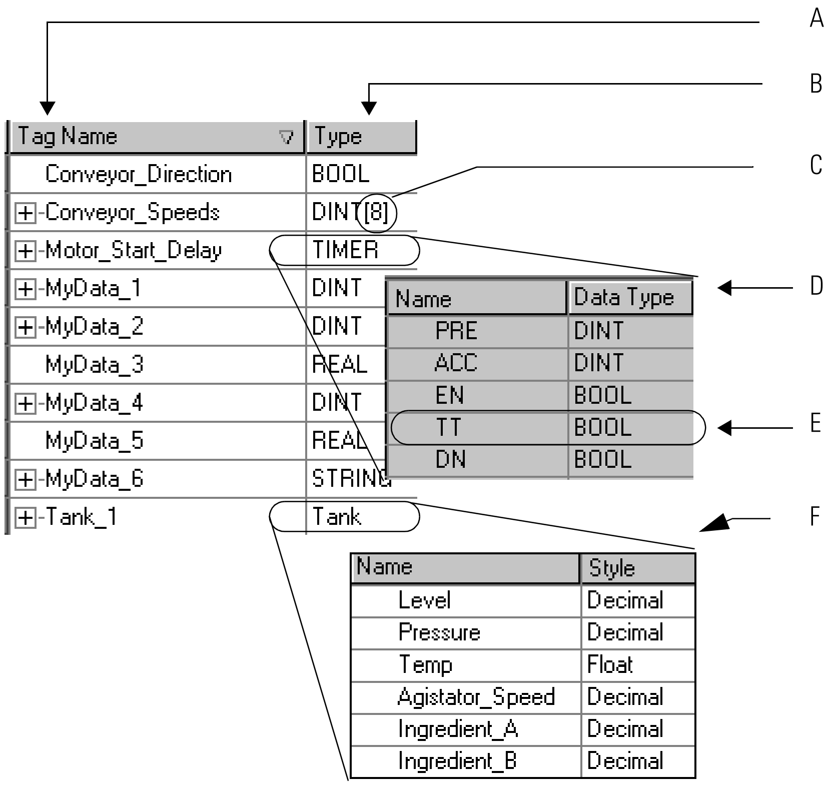
- Only letters, numbers, and underscores (_)
- Must start with a letter or an underscore
- ≤ 40 characters
- No consecutive or trailing underscores
- Not case-sensitive

Certain tags must be controller scope.

If you want to use a tag	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	
In a single program only	Program Tags for the program

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 Name	Type
Conveyor_Direction	BOOL
+-Conveyor_Speeds	DINT[8]
+-Motor_Start_Delay	TIMER
+-MyData_1	DINT
+-MyData_2	DINT
MyData_3	REAL
+-MyData_4	DINT
MyData_5	REAL
+-MyData_6	STRING
+-Tank_1	Tank

Name	Data Type
PRE	DINT
ACC	DINT
EN	BOOL
TT	BOOL
DN	BOOL

Name	Style
Level	Decimal
Pressure	Decimal
Temp	Float
Agistator_Speed	Decimal
Ingredient_A	Decimal
Ingredient_B	Decimal

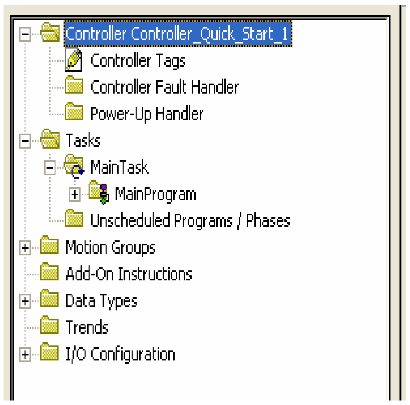
Item	Description
A	Tag stores data. There is no fixed data table or numeric format for data addresses. The tag name is the address. You create the tags that you want to use.
B	Data type defines the type of data that a tag stores, such as a bit, integer, floating-point value, or string.
C	Array defines a block of data (file). The entire block uses the same data type. It can have 1, 2, or 3 dimensions.
D	Structure combines 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 layout.
E	Member describes an individual piece of data within a structure.
F	User-defined data type defines your own data structure. 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
Consider the pass-through of descriptions.	See User-defined Data Type on page 98.
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.
If you include an array as a member, limit the array to a single dimension.	Multi-dimension arrays are not permitted in a user-defined data type.
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**: BOOL BOOL BOOL DINT DINT **Less Efficient**: BOOL DINT BOOL DINT BOOL

Follow these steps to create a user-defined data type.

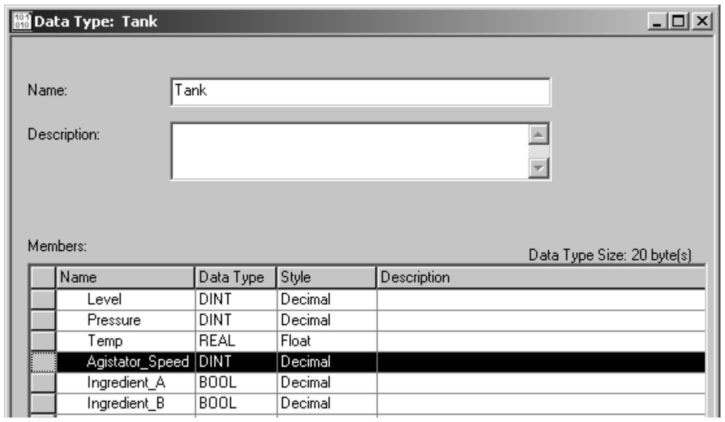
1. Right-click Data Types and choose New Data Type.



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

3. Enter the members.

   As an option, type a description for each member.



Data Type: Tank

Name: Tank

Description:

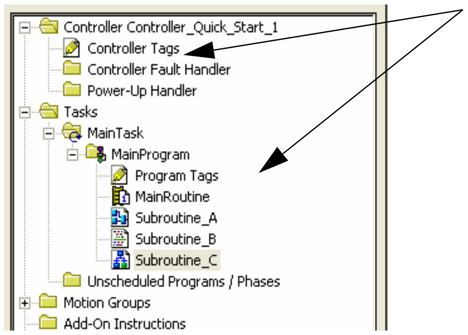
Members: Data Type Size: 20 byte(s)

Name	Data Type	Style	Description
Level	DINT	Decimal	
Pressure	DINT	Decimal	
Temp	REAL	Float	
Agistator_Speed	DINT	Decimal	
Ingredient_A	BOOL	Decimal	
Ingredient_B	BOOL	Decimal	

4. Click OK.

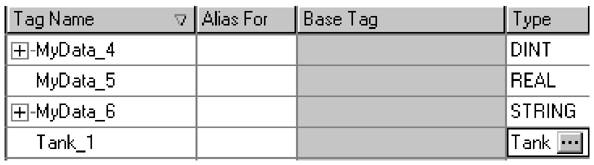
Follow these steps to create a tag that uses a user-defined data type.

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



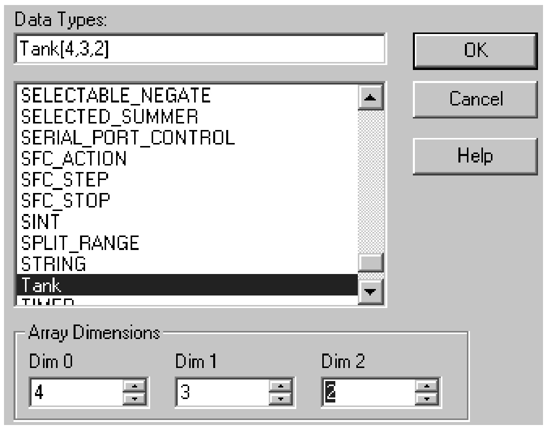
2. Type a name for the tag.

3. Type the name of the user-defined data type.



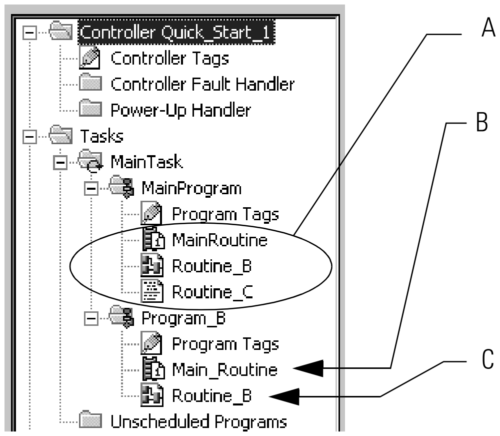
Tag Name	Alias For	Base Tag	Type
⊞-MyData_4			DINT
MyData_5			REAL
⊞-MyData_6			STRING
Tank_1			Tank

4. Do the following if you want the tag to be an array (multiple instances of the data type).
   a. Select the data type and click ….
   b. Specify the array dimensions and click OK.



Define Your Routines

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



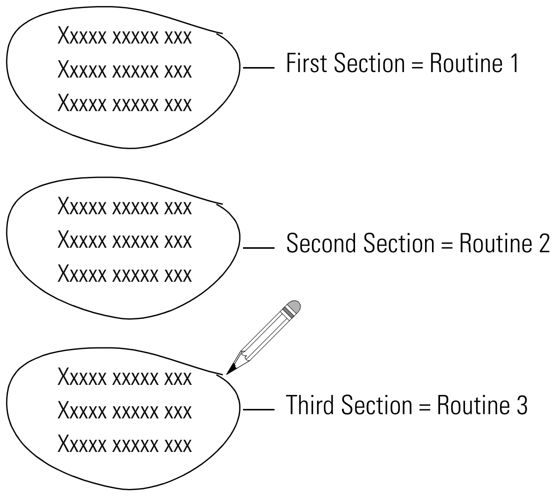
Item	Description
A	Routine provides the executable code (logic) for a program (similar to a program file in a PLC or SLC controller).
B	Main routine is required each program. • When the program executes, its main routine automatically executes. • Use the main routine to control the execution of the other routines in the program. • To call (execute) another routine (subroutine) within the program, use a Jump to Subroutine (JSR) instruction.
C	Subroutine is 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

Follow these steps 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.

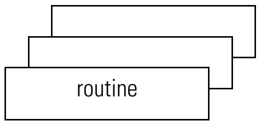
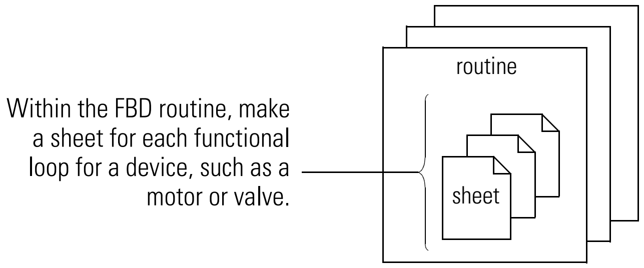
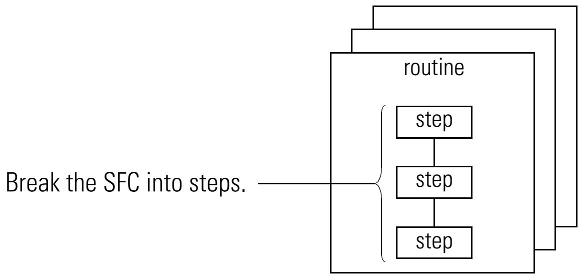
Assign a Programming Language to Each Routine

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	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.	
Continuous process and drive control	Function block diagram (FBD)
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)	 routine sheet Within the FBD routine, make a sheet for each functional loop for a device, such as a motor or valve.	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)	 routine step step step 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.

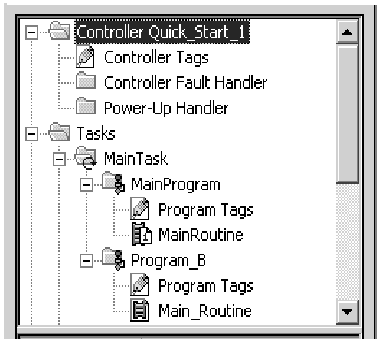
Assign Main Routines

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.

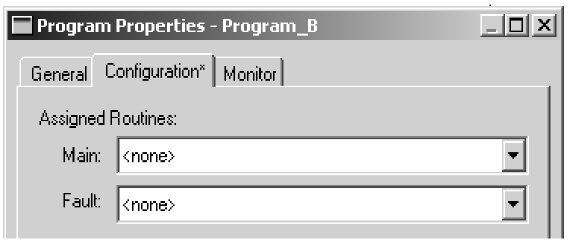
Follow these steps to assign a main routine.

1. Right-click the program folder and choose Properties.



2. Click the Configuration tab.

3. Select the main routine and click OK.

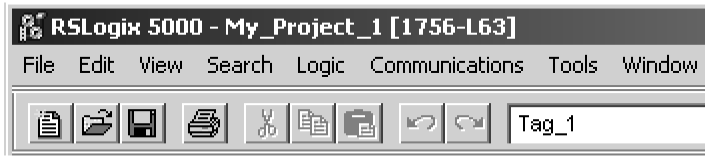


Assigned Routines:
Main: <none>
Fault: <none>

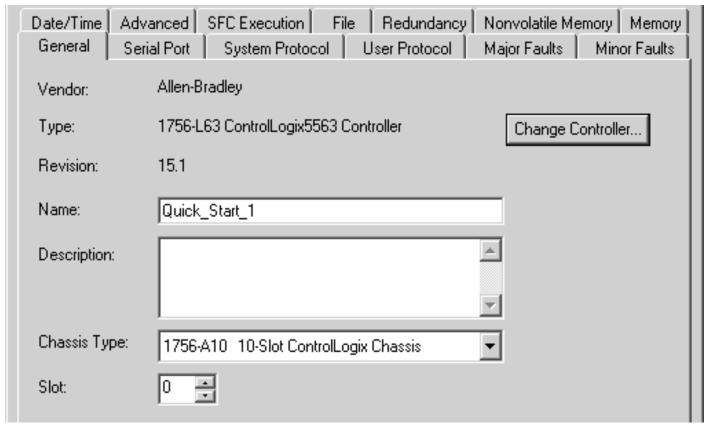
Configure the Controller

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.


RSLogix 5000 - My_Project_1 [1756-L63]
File Edit View Search Logic Communications Tools Window
Tag_1

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

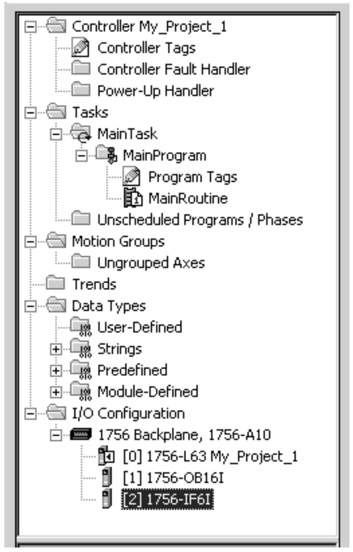


Vendor: Allen-Bradley
Type: 1756-L63 ControlLogix5563 Controller
Revision: 15.1
Name: Quick_Start_1
Description:
Chassis Type: 1756-A10 10-Slot ControlLogix Chassis
Slot: 0

Configure I/O Modules

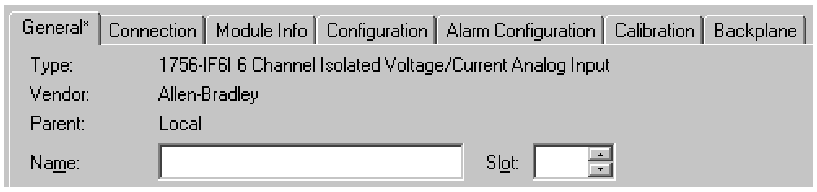
To change the behavior of a module, use the Module Properties window for the module. The configuration options vary from module to module.

1. Right-click the module and choose Properties.



```
Controller My_Project_1
    Controller Tags
    Controller Fault Handler
    Power-Up Handler
Tasks
    MainTask
        MainProgram
            Program Tags
            MainRoutine
    Unscheduled Programs / Phases
Motion Groups
    Ungrouped Axes
Trends
Data Types
    User-Defined
    Strings
    Predefined
    Module-Defined
I/O Configuration
    1756 Backplane, 1756-A10
        [0] 1756-L63 My_Project_1
        [1] 1756-OB16I
        [2] 1756-IF6I
```

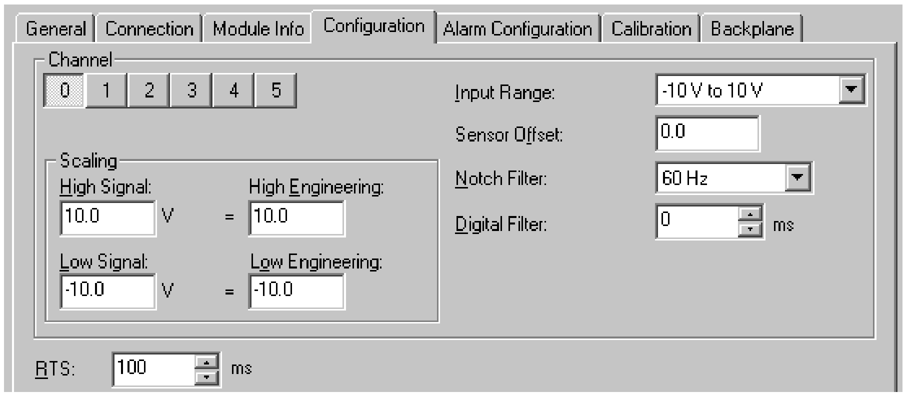
2. To change the name or slot number, use the General tab.



Type: 1756-IF6I 6 Channel Isolated Voltage/Current Analog Input
Vendor: Allen-Bradley
Parent: Local
Name:
Slot:

3. To change the configuration, click the Configuration tab.

   Some modules have several configuration tabs.



Scaling
High Signal: 10.0 V = High Engineering: 10.0
Low Signal: -10.0 V = Low Engineering: -10.0

Input Range: -10 V to 10 V
Sensor Offset: 0.0
Notch Filter: 60 Hz
Digital Filter: 0 ms

RTS: 100 ms

Notes:

Program Add-On Instructions

With version 16 of RSLogix 5000 programming software, you can design and configure sets of commonly used instructions to increase project consistency. Similar to the built-in instructions contained in Logix5000 controllers, these instructions you create are called Add-On Instructions.

With Add-On Instructions, you can:

- insert your own instruction.
- copy an Add-On Instruction definition from another RSLogix 5000 project.
- import an Add-On Instruction definition from another RSLogix 5000 project.

What You Need

You need these items to complete the tasks in this manual:

- Personal computer running RSLogix 5000 software, version 16 or later
- A layout of the system for which you are creating a project

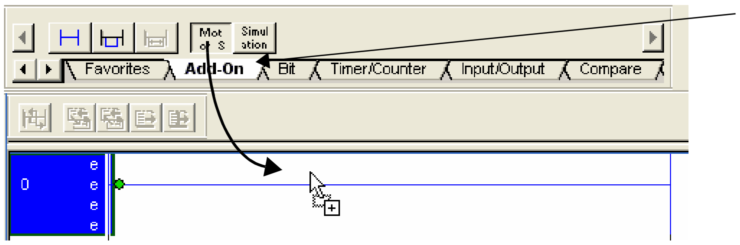
Follow These Steps

4. Insert an Add-On Instruction (page 52).
5. Copy an Add-On Instruction definition (page 53).
6. Import an Add-On Instruction definition (page 54).
7. Access a parameter that is not visible (page 55).
8. Monitor or change the value of a parameter (page 56).
9. View the logic of an Add-On Instruction (page 57).
10. Edit and monitor an Add-On Instruction (page 58).
11. Update an Add-On Instruction to a newer revision (page 58).

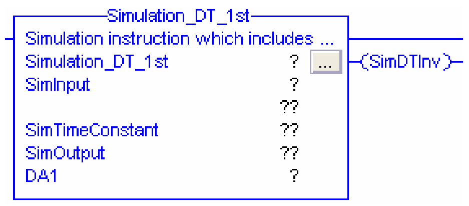
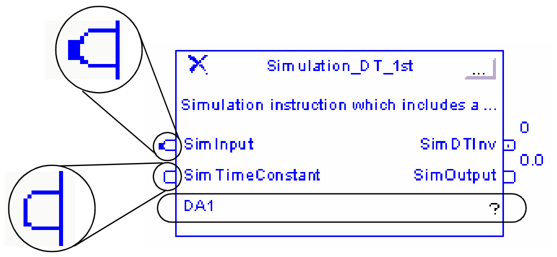
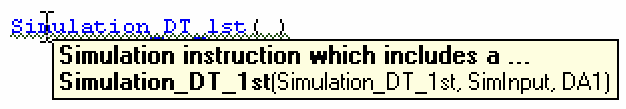
Insert an Add-On Instruction

You can create an Add-On Instruction in a ladder, function block, or structured text routine.

1. Open the routine that will use the instruction.
2. Click the Add-On tab of the Language Element toolbar.
3. Drag the instruction from the toolbar to the routine.



4. Enter the parameters.

Routine Type	Parameters
Ladder  Simulation_DT_1st Simulation instruction which includes ... Simulation_DT_1st ? SimInput ? ?? SimTimeConstant ?? SimOutput ?? DA1 ? (SimDTInv)	Single question mark — Required parameter. Enter a tag. Single and double question marks — Required parameter. Enter a tag. Only double question marks — Not a required parameter. You can either: • leave it blank and use the default value. • if it's an input value, enter a different value.
Function block  Simulation_DT_1st Simulation instruction which includes a ... SimInput SimDTInv 0 SimTimeConstant SimOutput 0.0 DA1 ?	Nub on the end of a pin — Required parameter. Wire the pin to an IREF, OREF, connector, or another block. Single question mark — Required parameter. Enter a tag. No nub on the end of a pin — Not a required parameter.
Structured text  Simulation_DT_1st( ) **Simulation instruction which includes a ...** **Simulation_DT_1st**(Simulation_DT_1st, SimInput, DA1)	The instruction takes only the required parameters. Enter a tag for each parameter.

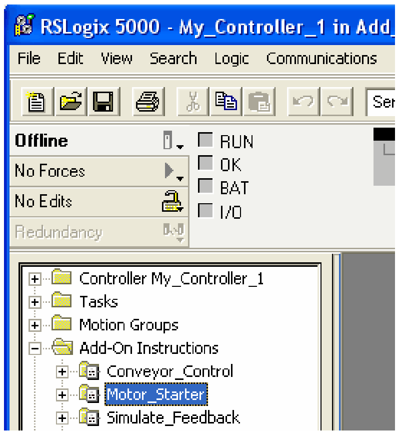
TIP For help with an instruction, select the instruction and then press [F1]. In structured text, make sure the cursor is in the blue instruction name.

Copy an Add-On Instruction Definition

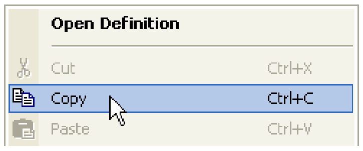
You can copy an Add-On Instruction from within another RSLogix 5000 project.

1. Open the RSLogix 5000 project that has the Add-On Instruction definition.

2. Find the definition in the Add-On Instructions folder.

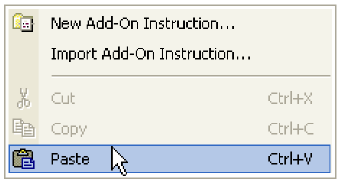


3. Right-click the definition and choose Copy.



4. Go to the project that gets the definition.

5. Right-click the Add-On Instructions folder and choose Paste.



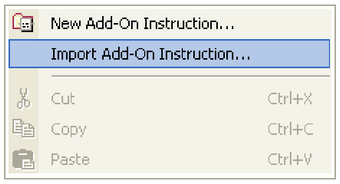
Import an Add-On Instruction Definition

You can add the definition of an Add-On Instruction that was exported from another RSLogix 5000 project.

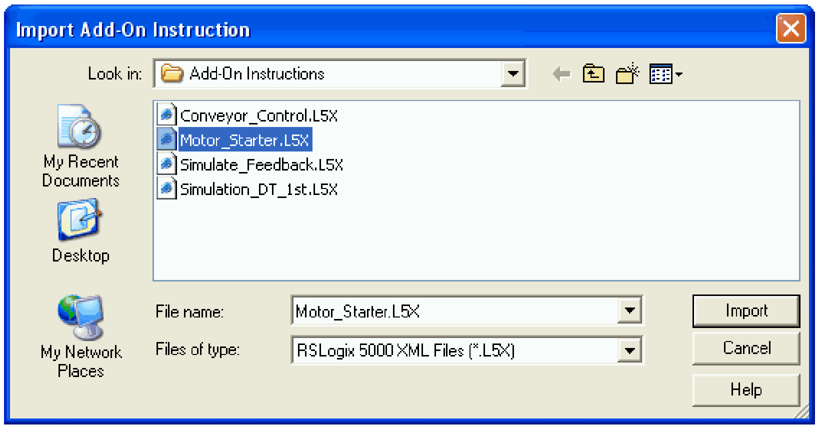
Does the RSLogix 5000 project already have a revision of this Add-On Instruction?

- No — use this procedure to import the instruction.
- Yes — see Update an Add-On Instruction to a Newer Revision on page 58

1. Right-click the Add-On Instructions folder and choose Import Add-On Instruction.



2. Find the instruction.



Conveyor_Control.L5X
Motor_Starter.L5X
Simulate_Feedback.L5X
Simulation_DT_1st.L5X

File name: Motor_Starter.L5X
Files of type: RSLogix 5000 XML Files (*.L5X)

3. Select the instruction and click Import.

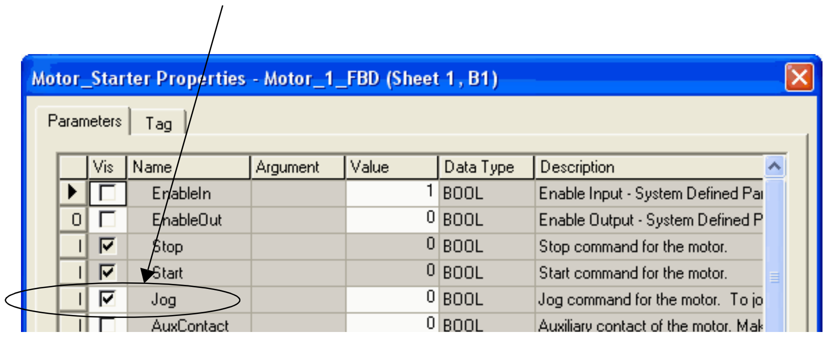
Access a Parameter That Is Not Visible

How you read or write to a parameter of an Add-On Instruction that is not visible depends on the programming language.

Function Block

1. Click the Properties button for the instruction.

2. Check the Vis box for the parameter and click OK.



Motor_Starter Properties - Motor_1_FBD (Sheet 1, B1)

	Vis	Name	Argument	Value	Data Type	Description
▶	☐	EnableIn		1	BOOL	Enable Input - System Defined Pa
0	☐	EnableOut		0	BOOL	Enable Output - System Defined P
I	☑	Stop		0	BOOL	Stop command for the motor.
I	☑	Start		0	BOOL	Start command for the motor.
I	☑	Jog		0	BOOL	Jog command for the motor. To jo
I	☐	AuxContact		0	BOOL	Auxiliary contact of the motor. Mak

3. Wire to the pin for the parameter.

Ladder Logic and Structured Text

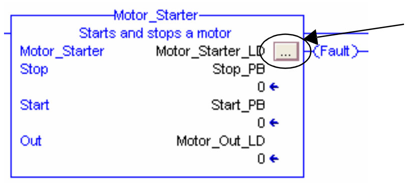
Use another instruction, an assignment, or an expression to read or write to the tag name of the parameter. Use this format for the tag name of the parameter.

```
Add_On_Tag.Parameter
```

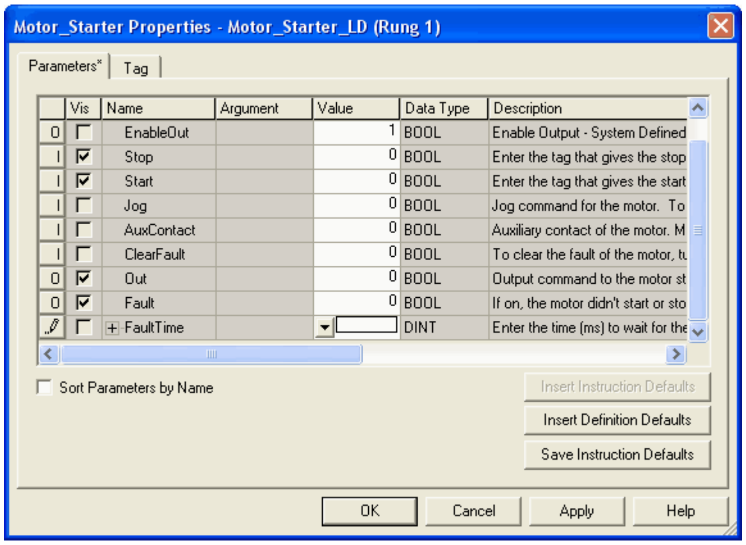
Where	Is
Add_On_Tag	Add-on-defined tag for the add-on instruction
Parameter	Name of the parameter

Monitor or Change the Value of a Parameter of an Add-On Instruction

Use the Properties dialog to monitor or change a parameter value of an Add-On Instruction.

Routine Type	**Parameters**
Ladder logic or function block  Motor_Starter Starts and stops a motor Motor_Starter Motor_Starter_LD ... Fault Stop Stop_PB 0 Start Start_PB 0 Out Motor_Out_LD 0	Click the Properties button for the instruction.
Structured text `Motor_Starter(Motor_Starter_ST,Stop_PB,Start_PB,Motor_Out_ST);`	Right-click the instruction and choose Properties.

1. Click and type the new value.



Motor_Starter Properties - Motor_Starter_LD (Rung 1)

Parameters* | Tag

	Vis	Name	Argument	Value	Data Type	Description
O	☐	EnableOut		1	BOOL	Enable Output - System Defined
I	☑	Stop		0	BOOL	Enter the tag that gives the stop
I	☑	Start		0	BOOL	Enter the tag that gives the start
I	☐	Jog		0	BOOL	Jog command for the motor. To
I	☐	AuxContact		0	BOOL	Auxiliary contact of the motor. M
I	☐	ClearFault		0	BOOL	To clear the fault of the motor, tu
O	☑	Out		0	BOOL	Output command to the motor st
O	☑	Fault		0	BOOL	If on, the motor didn't start or sto
	☐	+ FaultTime			DINT	Enter the time (ms) to wait for the

☐ Sort Parameters by Name

Insert Instruction Defaults
Insert Definition Defaults
Save Instruction Defaults
OK
Cancel
Apply
Help

2. Click Apply.

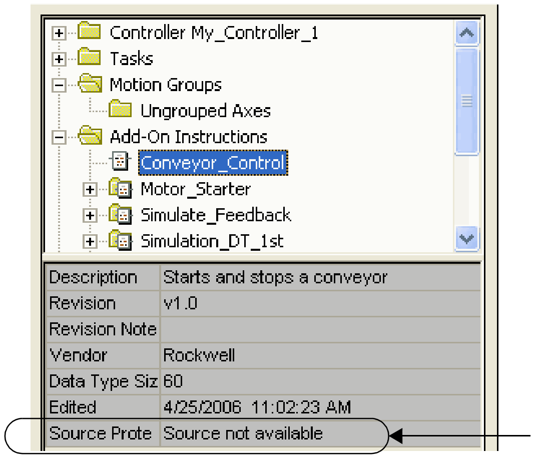
3. Click OK.

View the Logic of an Add-On Instruction

It's possible to protect an Add-On Instruction so that you can't see its logic. You can determine whether an Add-On Instruction is protected.

1. Select the add-on instruction.

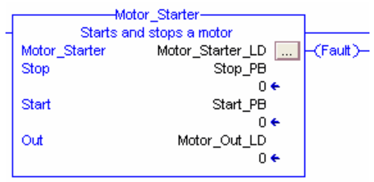
2. Look in the Quick View pane for Source Protection.



Description	Starts and stops a conveyor
Revision	v1.0
Revision Note	
Vendor	Rockwell
Data Type Siz	60
Edited	4/25/2006 11:02:23 AM
Source Prote	Source not available

If it isn't listed, then the routine isn't protected.

To view the logic, right-click the instruction and choose Open Instruction Logic.

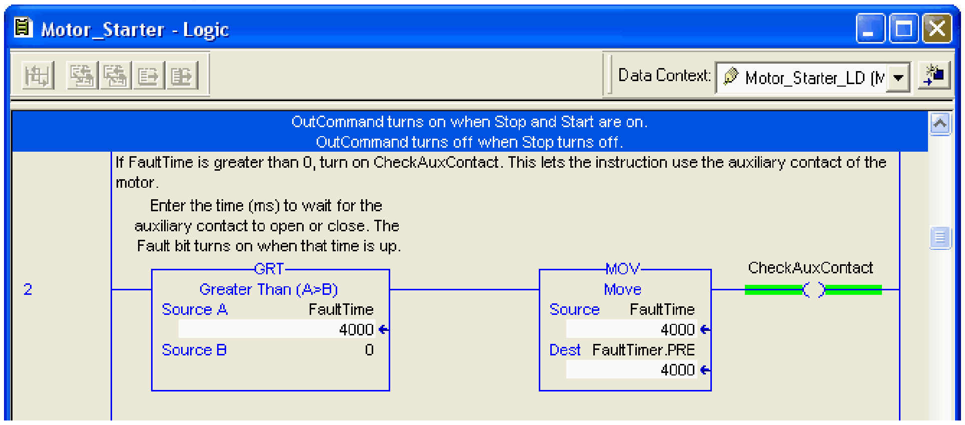


```
Motor_Starter(Motor_St
arter_ST,Stop_PB,Start
_PB,Motor_Out_ST);
```

Edit and Monitor an Add-On Instruction

You can:

- see the logic as it executes.
- see tag values.
- change tag and parameter values.



OutCommand turns on when Stop and Start are on.
OutCommand turns off when Stop turns off.

If FaultTime is greater than 0, turn on CheckAuxContact. This lets the instruction use the auxiliary contact of the motor.

Enter the time (ms) to wait for the auxiliary contact to open or close. The Fault bit turns on when that time is up.

You cannot:

- edit logic online.
- edit logic for just this instruction.

To edit the logic, you must edit the definition.

Update an Add-On Instruction to a Newer Revision

You can update the definition of an Add-On Instruction to a newer revision.

IMPORTANT Before you change the definition of an add-on instruction, make sure the change won't cause problems with existing instances of that instruction. When you change the definition of an add-on instruction, the change affects all the instances of that instruction in your project.

For example, if a project uses a certain Add-On Instruction 5 times, update the definition so that all 5 instances change when you change the definition.

1. Right-click the Add-On Instructions folder and choose Import Add-On Instruction.
2. Find the instruction and choose Import.
3. Decide how to handle the conflict with the existing revision (probably overwrite).
4. Use a cross-reference list to check each use of the instruction in your logic.

Program an Equipment Phase

Use PhaseManager software to create an equipment phase and change the default settings for the equipment phase.

What You Need

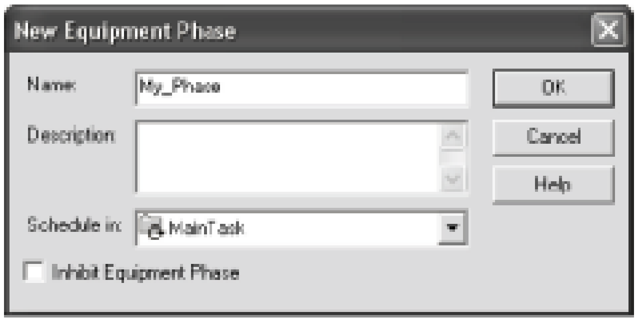
You need these items to complete the tasks in this manual:
- Personal computer running RSLogix 5000 software, version 16 or later
- A layout of the system for which you are creating a project

Follow These Steps

5. Create an equipment phase (page 60).
6. Create a state routine (page 60).
7. Manually step through the states (page 61).
8. Configure the initial state for an equipment phase (page 63).
9. Open the configuration for an equipment phase (page 64).
10. Configure an equipment phase (page 64).

Create an Equipment Phase

1. Right-click Main Task and choose New Equipment Phase.



Name: My_Phase

Description:

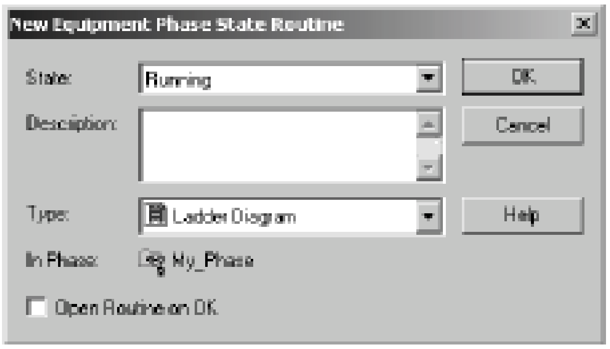
Schedule in: MainTask

Inhibit Equipment Phase

2. Type a name for the equipment phase and click OK.

Create a State Routine

1. Right-click Main Task and choose the equipment phase.
2. Choose New Phase State Routine.
3. Type a name for the state routine.
4. Select the programming language and click OK.



State: Running

Description:

Type: Ladder Diagram

In Phase: My_Phase

Open Routine on OK

Manually Step Through the States

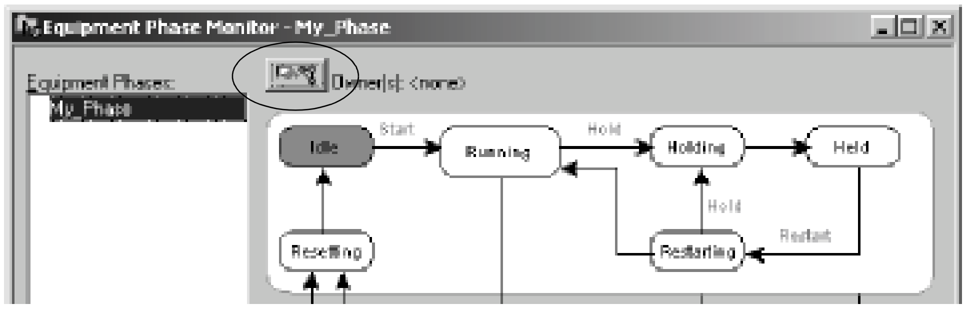
Before you step through states, do the following:

- Download the project to the controller.
- Put the controller in run or remote run mode.

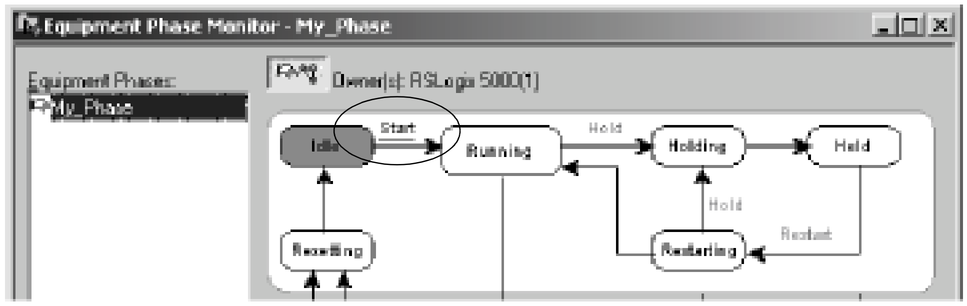
1. Right-click the equipment phase and choose Monitor Equipment Phase.

2. Click the Ownership and then Yes.

   Use this window to step through the states.

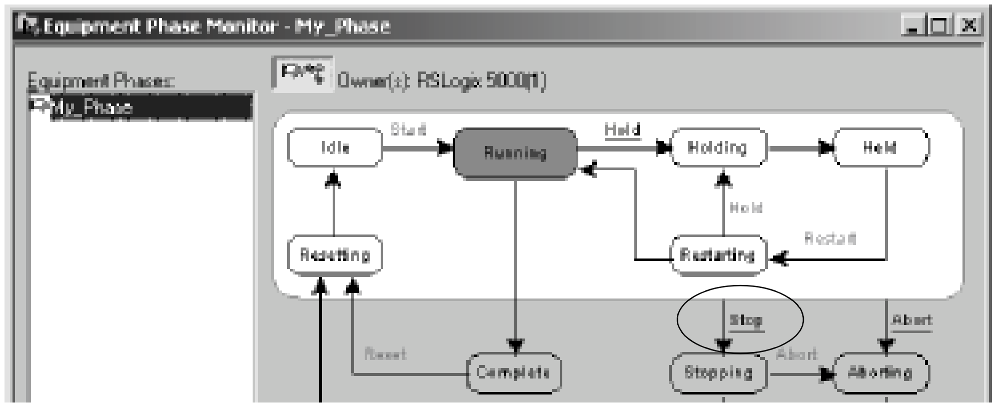


3. Click Start.



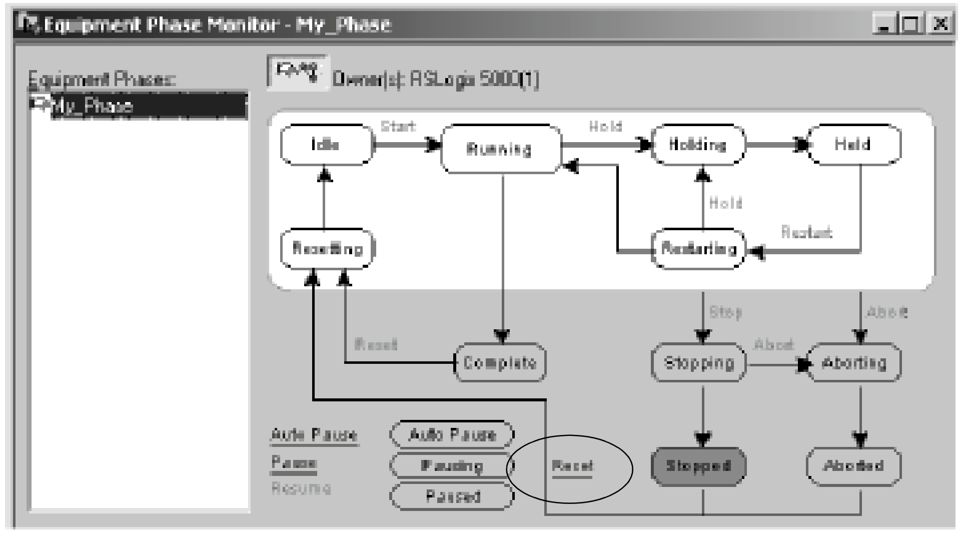
The equipment phase goes to the Running state. Any code in the Running state routine starts running. This is where you put the code for the normal production sequence of your equipment.

4. Click Stop.



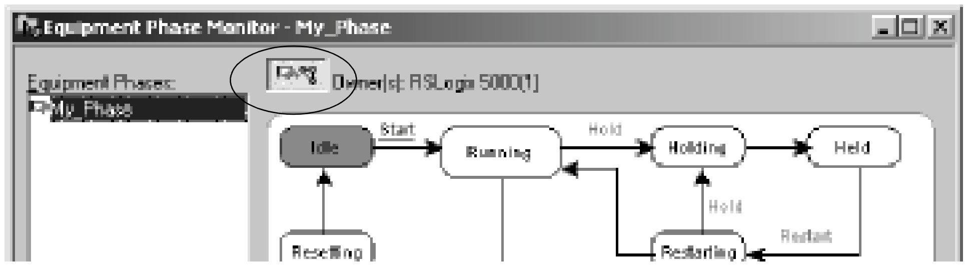
The equipment phase goes to the Stopped state.The Running state routine stops running. The Stopping state routine is optional. Without it, the equipment phase goes directly to the Stopped state.

5. Click Reset.



The equipment phase goes to the Idle state. The Resetting state routine is optional. Without it, the equipment phase goes directly to the Idle state.

6. Click Ownership.



This releases the equipment phase from control by this window.

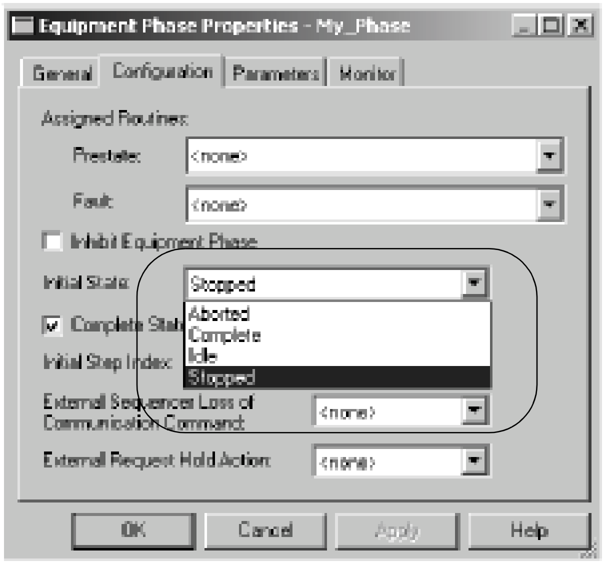
Configure the Initial State for an Equipment Phase

The initial state is the first state to which the equipment phase goes after you apply power.

1. Right-click the equipment phase and choose Properties.

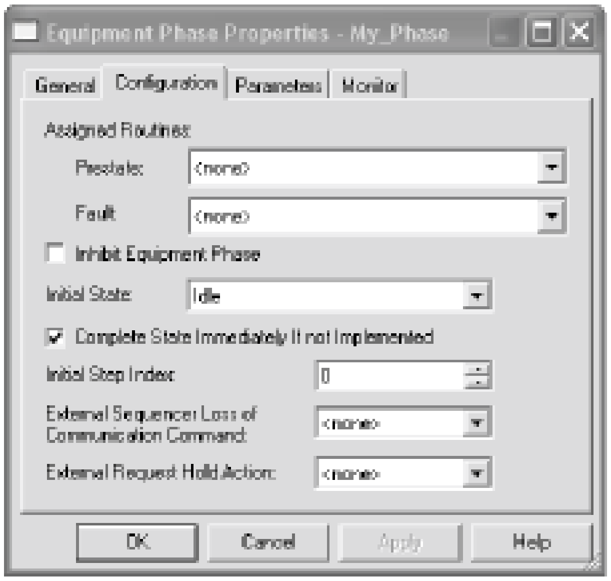
2. Choose the Configuration tab.

3. Choose the Initial State and click OK.



Open the Configuration for an Equipment Phase

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



Equipment Phase Properties - My_Phase

General | Configuration | Parameters | Monitor

Assigned Routines:

Prestate: <none>

Fault: <none>

☐ Inhibit Equipment Phase

Initial State: Idle

☑ Complete State Immediately If not Implemented

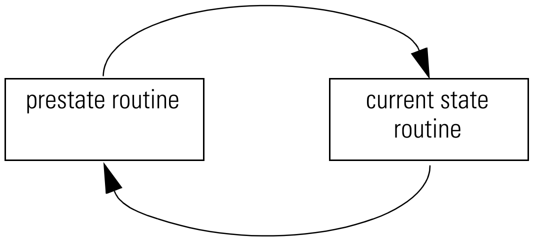
Initial Step Index: 0

External Sequencer Loss of Communication Command: <none>

External Request Hold Action: <none>

Configure an Equipment Phase

Use the following settings to configure an equipment phase.

Setting	Choices
Prestate	 The prestate routine runs all the time, even when the equipment phase is in the idle state. It runs before each scan of a state. Do you want to run a prestate routine? • Yes — Select the routine that you want to run. • No — Leave this box set to <none>

Setting	**Choices**
Fault	A fault routine lets you clear a major fault made by an instruction. Do you want to set up a fault routine for the instructions in this equipment phase? • Yes — Select the routine that you want as your fault routine. • No — Leave this box set to <none>
Inhibit Equipment Phase	Do you want the controller to inhibit this equipment phase? • Yes — Check this box. • No — Leave this box unchecked or uncheck it.
Initial State	Which state do you want the equipment phase to go to when you turn on the controller? • Idle • Complete • Stopped • Aborted
Complete State Immediately If not Implemented	Do you want the equipment phase to skip any states that you aren't using? • Yes — Leave this box checked or check it. • No — Uncheck this box.
Initial Step Index	Are any of the state routines in ladder diagram or structured text? • No — Skip this box. • Yes — Go to the next question. Do any of those state routines use step numbers? • Yes — Type the number for the first step of each state. • No — Skip this box. The tag for the equipment phase has a StepIndex number. The controller resets the StepIndex each time the equipment phase changes states. The controller resets the StepIndex to the number you put in the Initial Step Index box.
External Sequencer Loss of Communication Command	Are you using FactoryTalk Batch software to command this equipment phase? • No — Skip this box. • Yes — Go to the next question. If the controller loses communication with FactoryTalk Batch software, what do you want the equipment phase to do? • Continue in its current state — Select None. • Go to aborting — Select Abort. • Go to holding — Select Hold. • Go to stopping — Select Stop. The equipment phase must still follow the state model. For example, it goes to holding only if it is in running or restarting when communication fails.

Setting	Choices
External Request Hold Action	Are you using any PXRQ instructions? • No — Skip this box. • Yes — Go to the next question. What do you want to do if an equipment phase goes to holding while a PXRQ instruction is in process? • Nothing — Select None. • Stop the request — Select Clear.

Program a Project Offline

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

What You Need

You need these items to complete the tasks in this manual:

- Personal computer running RSLogix 5000 Software, version 16
- A plan for the project you are programming

Before You Begin

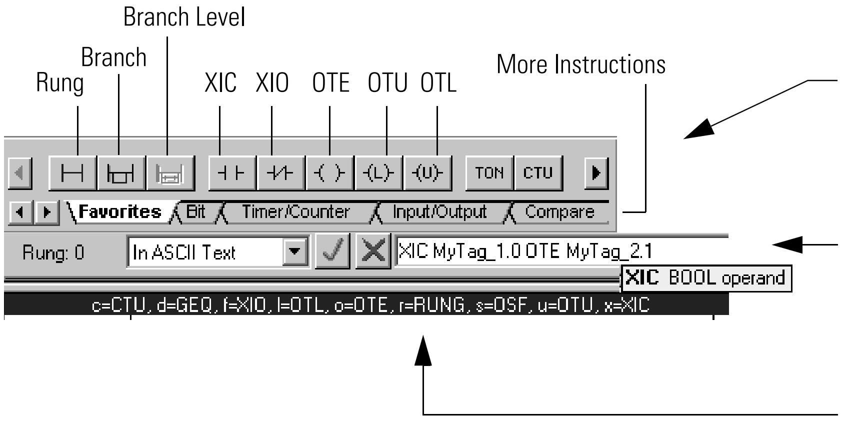
In this chapter, you program the project while offline. Online programming requires additional steps. See chapter Chapter 8, Program a Project Online.

Follow These Steps

1. Enter ladder logic (page 68).
2. Export/import ladder logic (page 73).
3. Enter a function block diagram (page 77).
4. Use a faceplate for a function block (page 81).
5. Enter structured text (page 84).
6. Enter a sequential function chart (page 87).
7. Assign operands (page 89).
8. Verify a project (page 92).
9. Review guidelines for tags (page 94).

Enter Ladder Logic

To enter ladder logic, you have the following options:

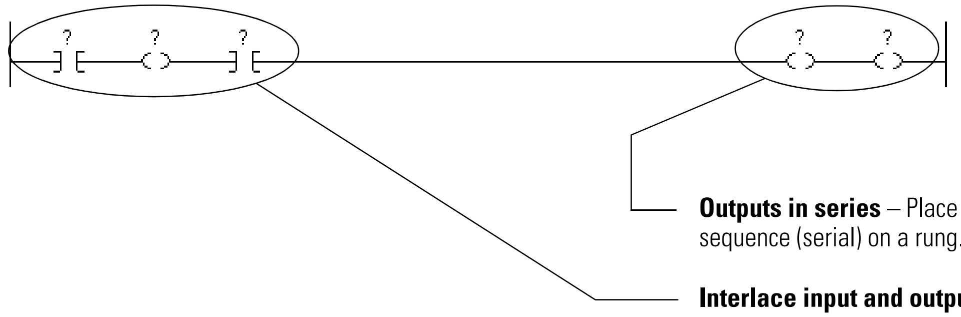


Drag and drop logic elements – Use the Language Element toolbar to drag and drop a rung, branch, or instruction to your routine.

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:

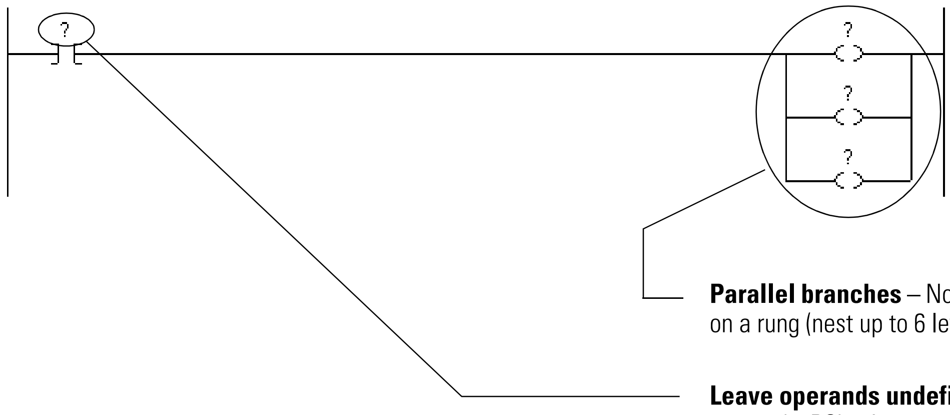
```
mnemonic operand_1 operand_2
```

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.

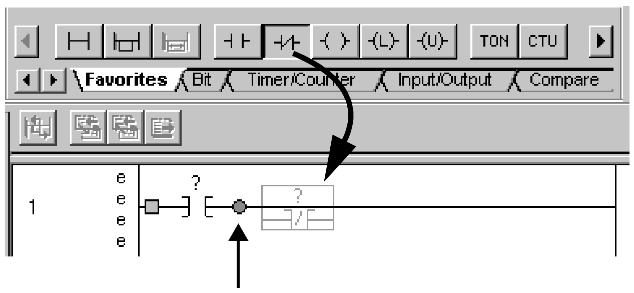


Parallel branches – No limit to the number of parallel branches on a rung (nest up to 6 levels).

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.

Add a Rung or an Instruction

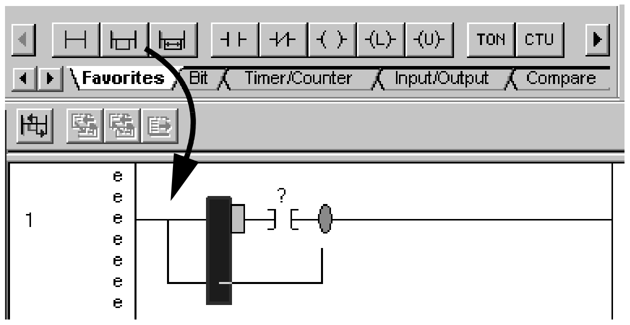
Drag the button for the rung or instruction directly to the desired location. A green dot shows a valid placement location (drop point).



Add a Branch

1. Drag the branch button to where the branch starts.

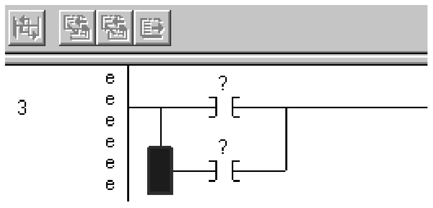
   A green dot shows a valid placement location (drop point).



2. Drag a branch rail to the desired location.

Add a Level to a Branch

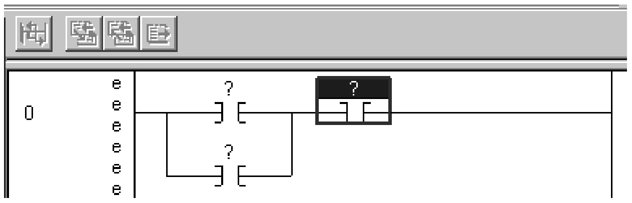
Right-click the branch and choose Add Branch Level.



Delete an Element

1. Click the element.

2. Click Delete.

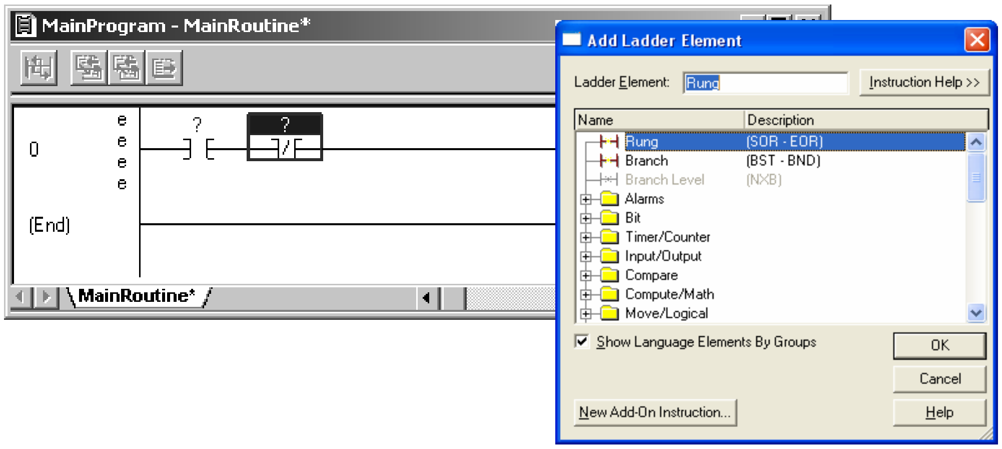


Use the Keyboard to Add an Element

You can add elements by using the computer keyboard.

1. Press Insert.

2. Type the mnemonic for the instruction or type Rung, Branch, or Branch Level.

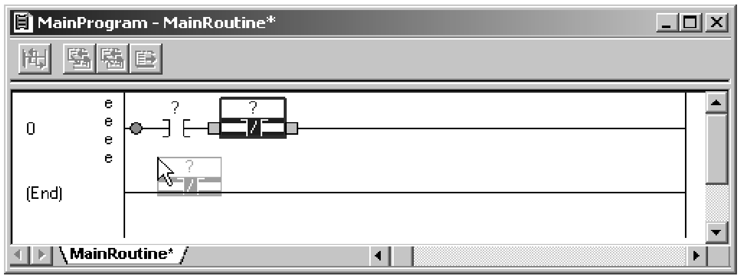


Name	Description
Rung	(SOR - EOR)
Branch	(BST - BND)
Branch Level	(NXB)
Alarms	
Bit	
Timer/Counter	
Input/Output	
Compare	
Compute/Math	
Move/Logical	

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).

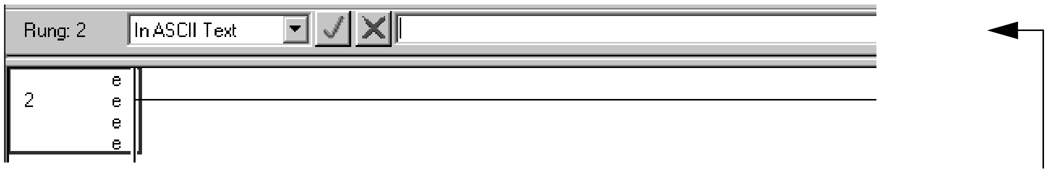


Enter Logic Using ASCII Text

You can add elements by using the typing their ASCII equivalents.

1. Double-click the rung.

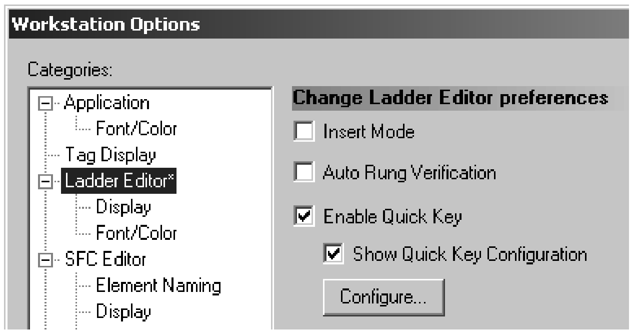
2. Enter the ASCII text for the rung.



Enable Quick Keys

Enable quick keys to use shortcuts when editing.

1. From the Tools menu, choose Options.
2. Click Ladder Editor.
3. Check these checkboxes.


Workstation Options

Categories:

- Application
  - Font/Color
- Tag Display
- Ladder Editor*
  - Display
  - Font/Color
- SFC Editor
  - Element Naming
  - Display

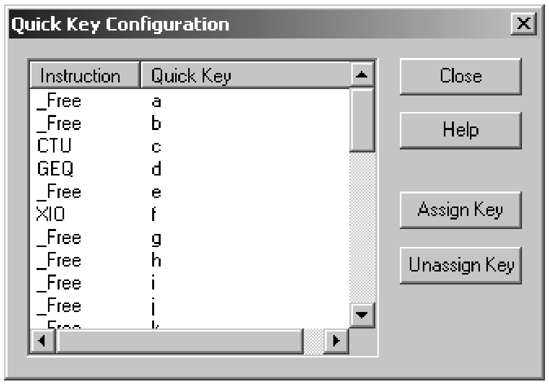
Change Ladder Editor preferences

- [ ] Insert Mode
- [ ] Auto Rung Verification
- [x] Enable Quick Key
  - [x] Show Quick Key Configuration

Configure...

To assign a key to an element:

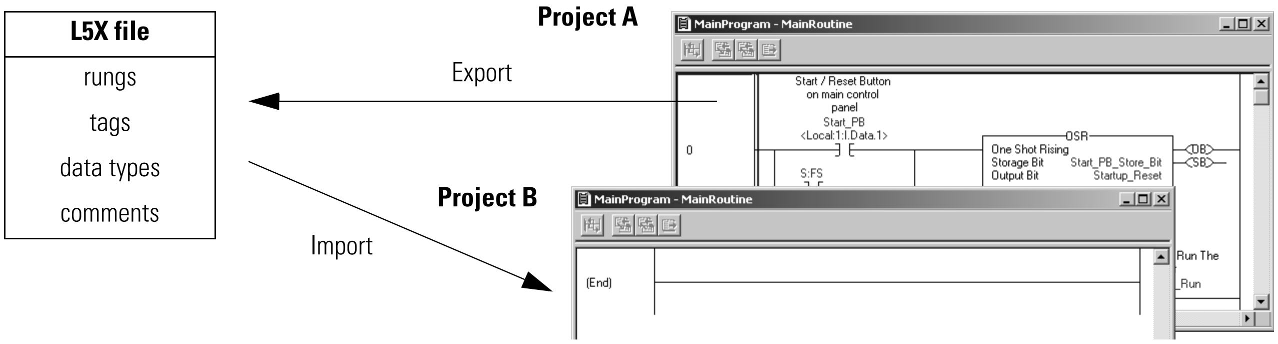
1. Click Configure... .
2. For the desired key, select the element.
3. When you have assigned the desired keys, click Close .


Quick Key Configuration

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

Export/Import Ladder Logic

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.

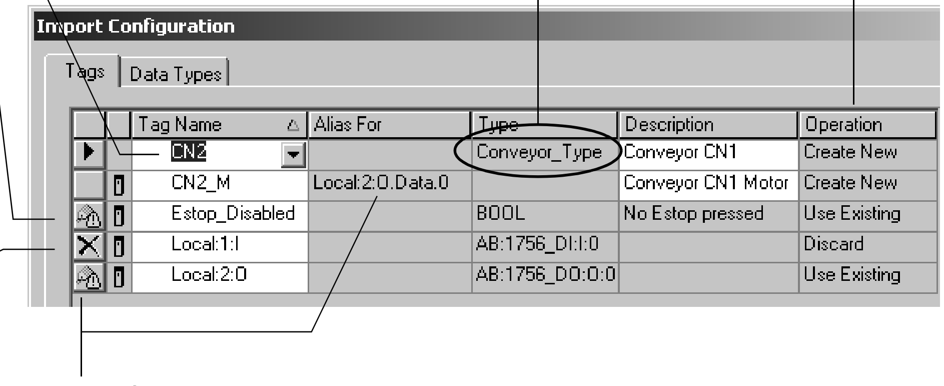
The Operation column shows what will happen to each tag and data type during the import. The software either creates it, uses an existing one in the project, or discards it (does not import it).

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

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

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.



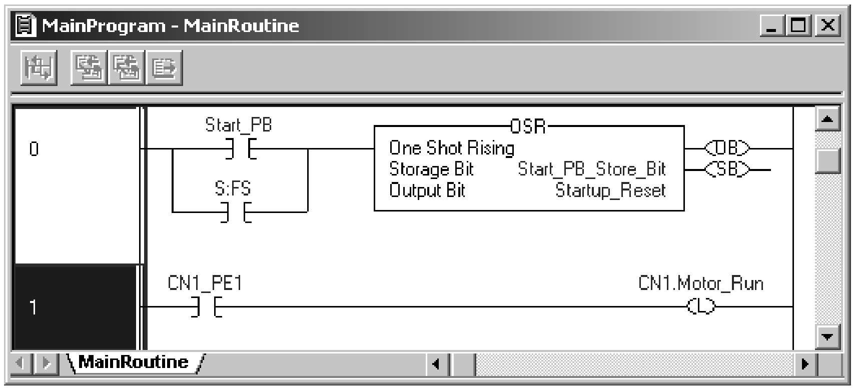
Tag Name	Alias For	Type	Description	Operation
CN2		Conveyor_Type	Conveyor CN1	Create New
CN2_M	Local:2:O.Data.0		Conveyor CN1 Motor	Create New
Estop_Disabled		BOOL	No Estop pressed	Use Existing
Local:1:I		AB:1756_DI:I:0		Discard
Local:2:O		AB:1756_DO:O:0		Use Existing

No new I/O tags are created.

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.

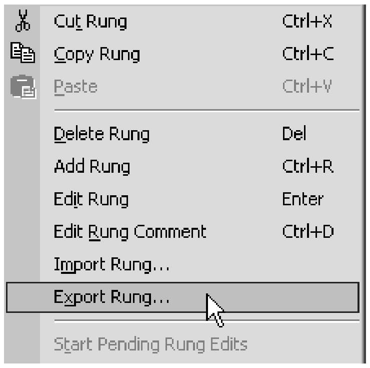
Export Rungs

1. Select the rungs to export.



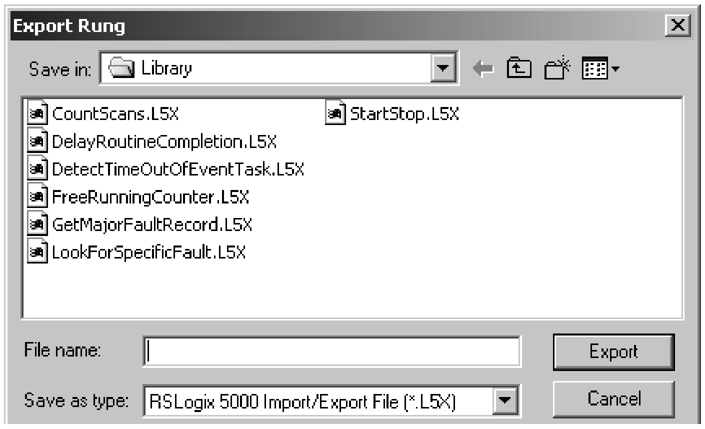
If rungs are	Do this
In sequence	Click the first rung and then Shift + click the last rung.
Out Of sequence	Click the first rung and then Ctrl + click each additional rung.

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



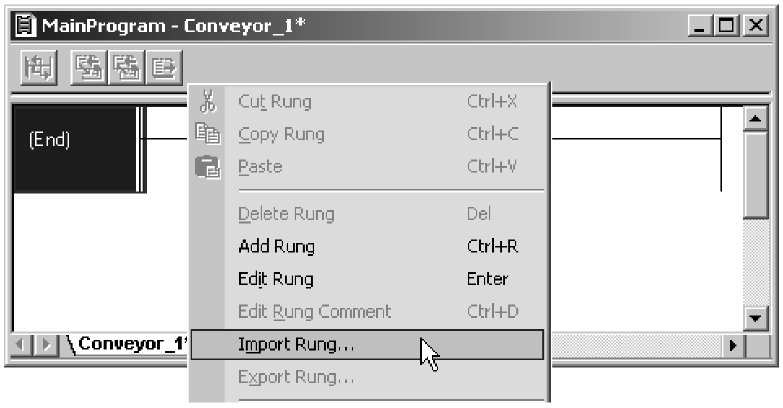
3. Choose a location and name for the file.

4. Create the file.

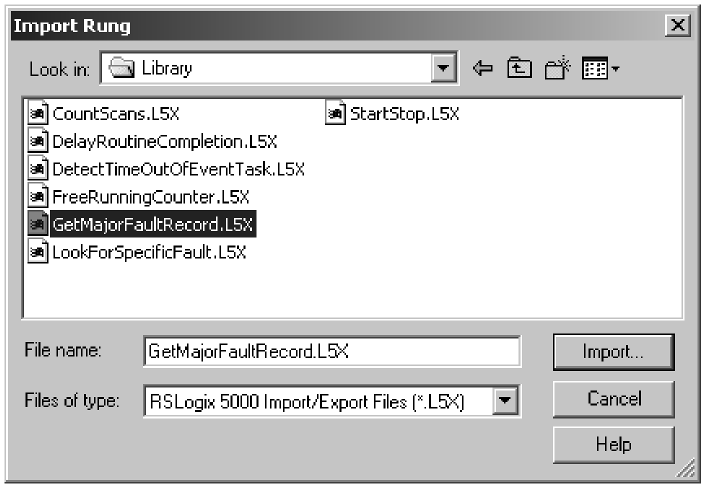


Import Rungs

1. Right-click the location for the rungs and choose Import Rung.



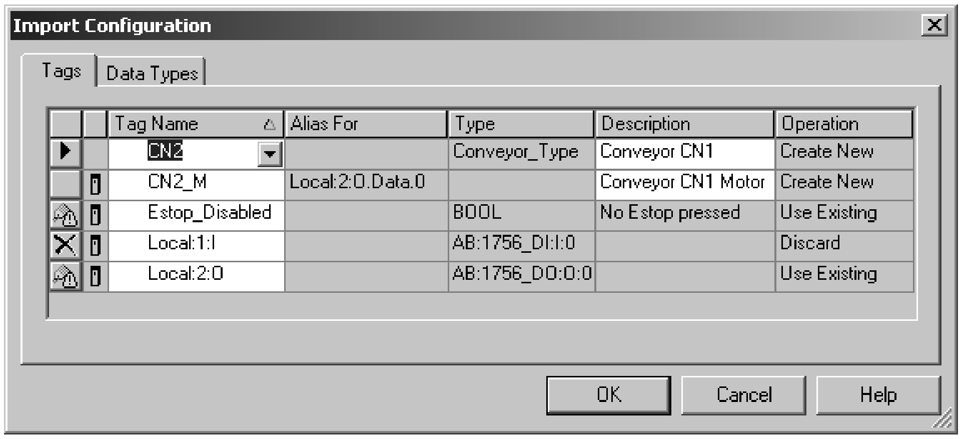
2. Select the file to import and click Import.



File name: GetMajorFaultRecord.L5X

Files of type: RSLogix 5000 Import/Export Files (*.L5X)

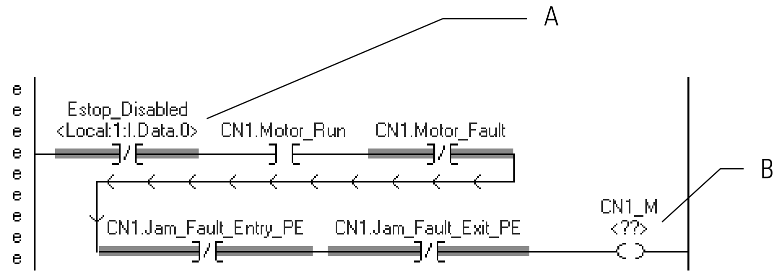
3. Check for conflicts in names.



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

4. Click OK.

Check Alias Tags

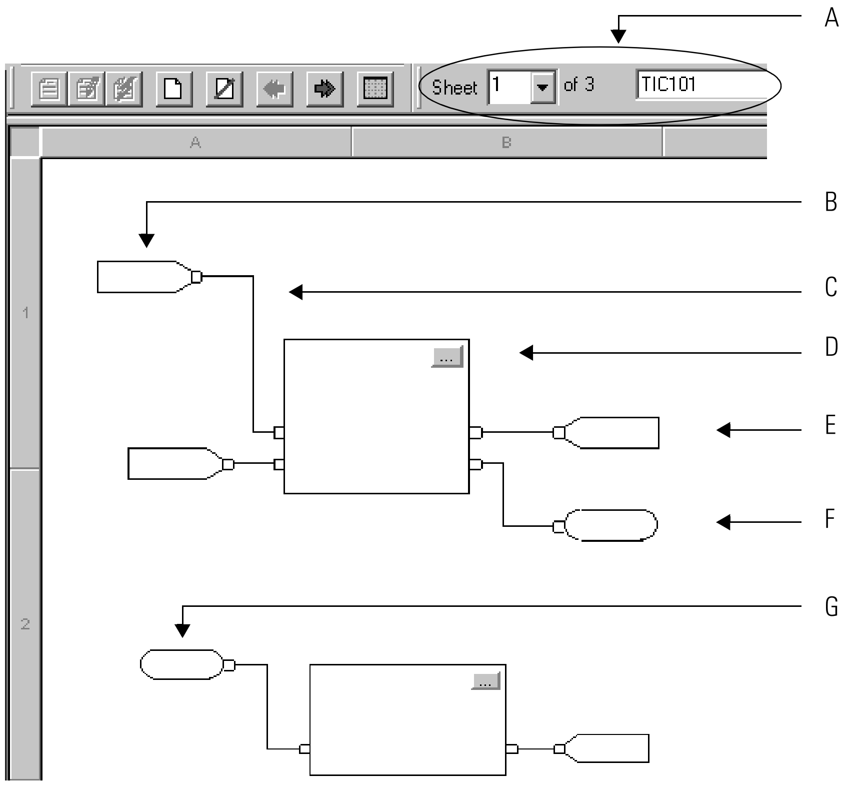


Check the alias tags in the rungs you import.

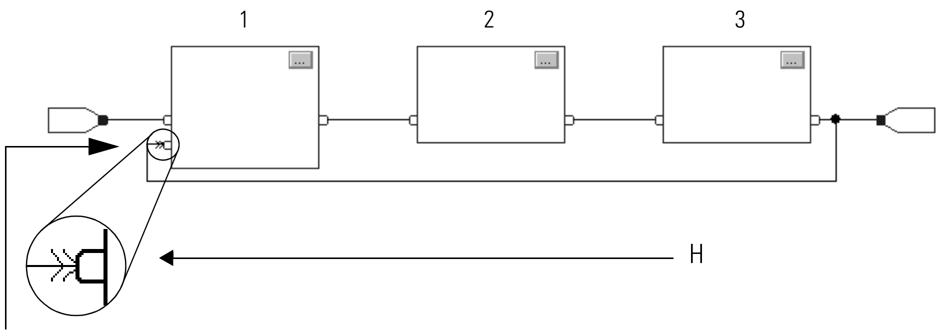
Item	Description
A	If you import an alias tag, make sure it points to the correct base tag. When a tag is an alias for a tag that already exists in the project, the software sets up the relationship between the alias and base tags.
B	If the project does not have the base tag, you have to either create 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.



Item	Description
A	Sheet divides the diagram into sections without affecting execution. When the routine executes, all sheets execute.
B	Input reference (IREF) reads a value from a tag or set a constant value
C	Wire transfers a value between elements
D	Instruction produces output values based on input values
E	Output reference (OREF) writes a value to a tag
F	Output wire connector (OCON) writes a value to one or more ICONs
G	Input wire connector (ICON) reads a value from an OCON on the same sheet or a different sheet in the routine. To read a value from another routine, use an OREF/IREF combination.



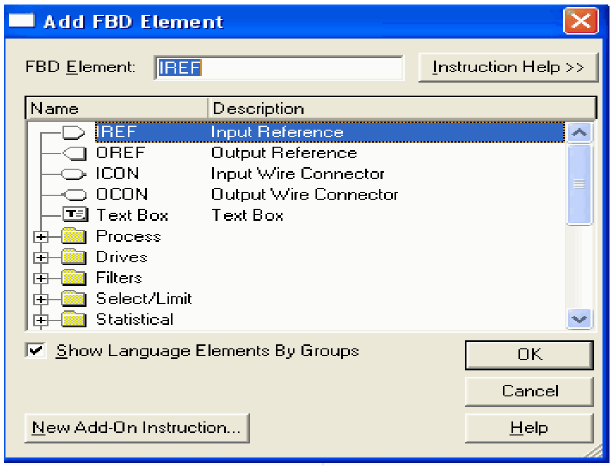
Item	Description
H	Assume data available indicator defines the data flow within the loop. The arrow indicates that the data serves as input to the first block in the loop. If a group of blocks are in a loop, you have to identify which block to execute first. Use the Assume Data Available indicator to mark the input wire that creates the loop (the feedback wire). This input pin uses the output that block 3 produced on the previous scan.

Use the Keyboard to Add an Element

You can add function block elements by using the computer keyboard.

1. Press Insert.

2. Type the mnemonic for the element and press Enter.

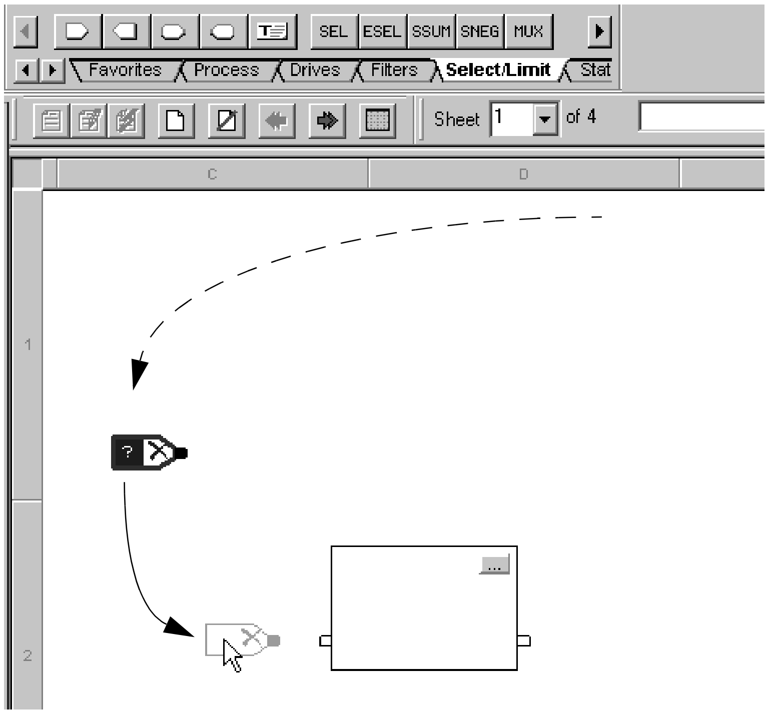

Add FBD Element

FBD Element: IREF

Name	Description
IREF	Input Reference
OREF	Output Reference
ICON	Input Wire Connector
OCON	Output Wire Connector
Text Box	Text Box
Process	
Drives	
Filters	
Select/Limit	
Statistical	

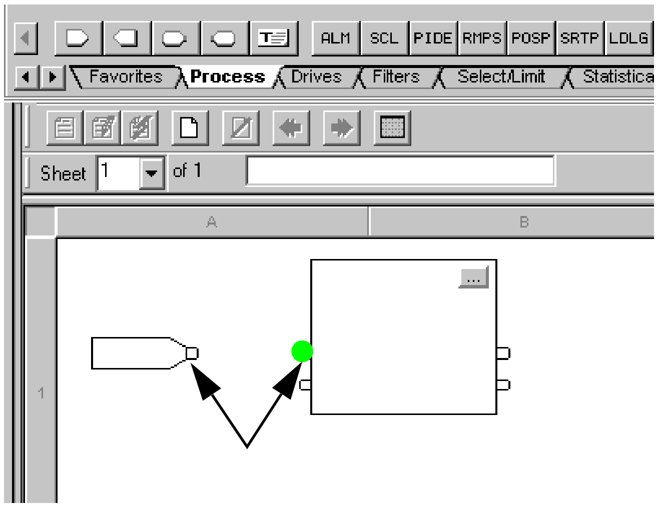
☑ Show Language Elements By Groups

3. Drag the element to the desired location.



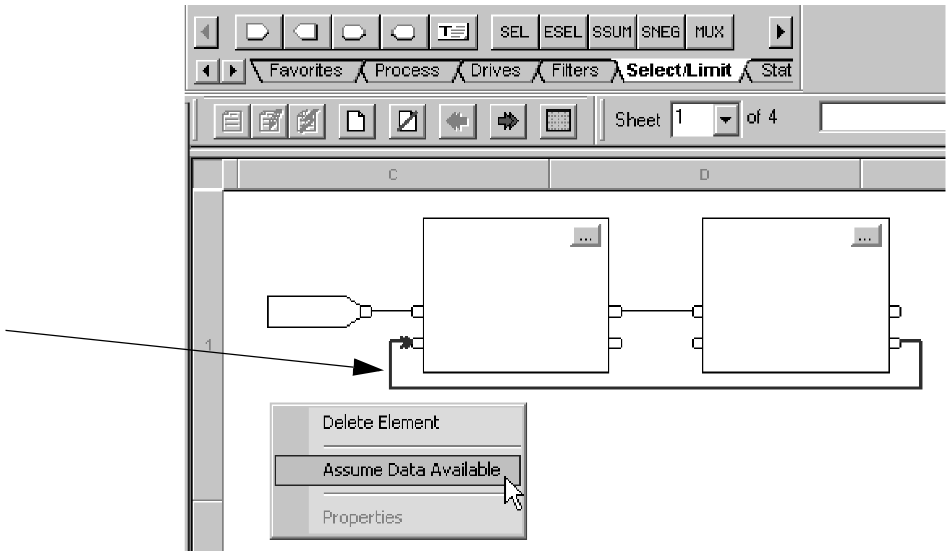
Connect Elements

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.

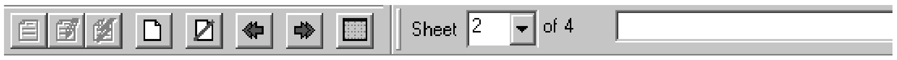


Add Sheet

Add sheets as need to a function block diagram.

1. Click New Sheet.

2. Type a name for the sheet.

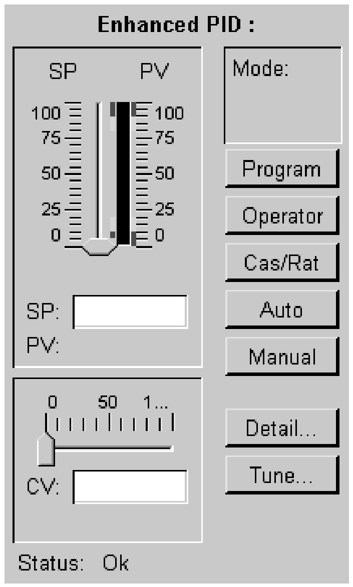


Use a Faceplate for a Function Block

RSLogix 5000 software includes faceplates (controls) for some of the function block instructions. A faceplate is an Active-X control that lets you interact with a function block instruction.

- Your RSLogix 5000 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:
  - FactoryTalk View SE
  - Microsoft Excel
- RSLogix 5000 software is not a valid Active-X container.
- Faceplates communicate with the controller via DDE/OPC topics in RSLinx Classic software. To use RSLinx Classic software for DDE/OPC topics, purchase either:
  - RSLinx Classic software as a separate package
  - RSLogix 5000 professional edition software, which includes RSLinx Classic professional edition software

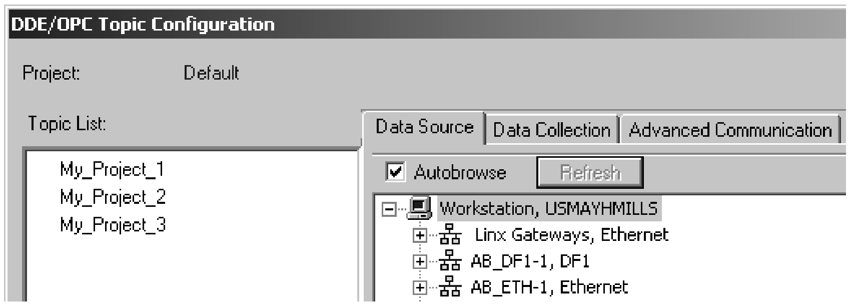
  RSLinx Classic Lite software, which comes with the other RSLogix 5000 software packages, does not provide DDE/OPC communication.



In RSLinx Classic 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.


DDE/OPC Topic Configuration

Project: Default

Topic List:

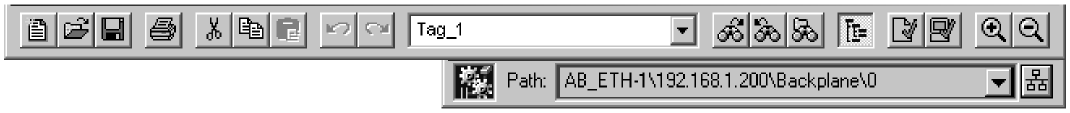
My_Project_1
My_Project_2
My_Project_3

Workstation, USMAYHMILLS
Linx Gateways, Ethernet
AB_DF1-1, DF1
AB_ETH-1, Ethernet

Set Up a Topic

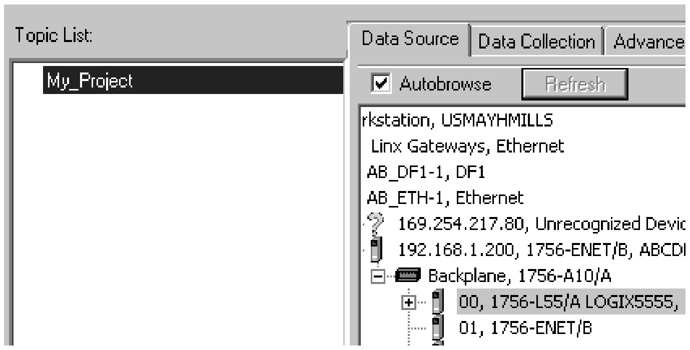
1. Use RSLogix 5000 software to create the topic.

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


Tag_1

Path: AB_ETH-1\192.168.1.200\Backplane\0

2. In RSLinx Classic software, check the topic.
   a. From the DDE/OPC menu, choose Topic Configuration.
   b. Select your project.
   c. Make sure the data source points to your controller.
   d. Click Done.

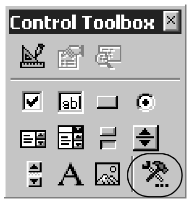

Topic List:

My_Project

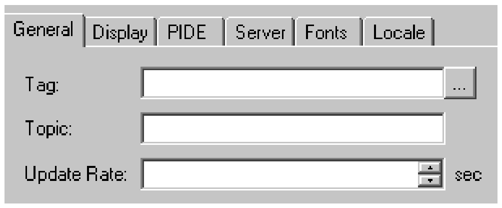
rkstation, USMAYHMILLS
Linx Gateways, Ethernet
AB_DF1-1, DF1
AB_ETH-1, Ethernet
169.254.217.80, Unrecognized Devic
192.168.1.200, 1756-ENET/B, ABCDI
Backplane, 1756-A10/A
00, 1756-L55/A LOGIX5555,
01, 1756-ENET/B

Add a Faceplate to Microsoft Excel Software

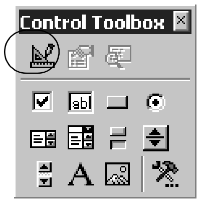
1. Start Microsoft Excel software.
2. Choose View > Toolbars > Control Toolbox.
3. Click and select the Logix 5000…Faceplate Control that you want.
4. In the location for the faceplate, drag the pointer to the desired size of the faceplate.
5. Right-click the faceplate and from the Logix 5000…Faceplate Control Object menu, choose Properties.



6. Click and browse to the tag that the faceplate controls.

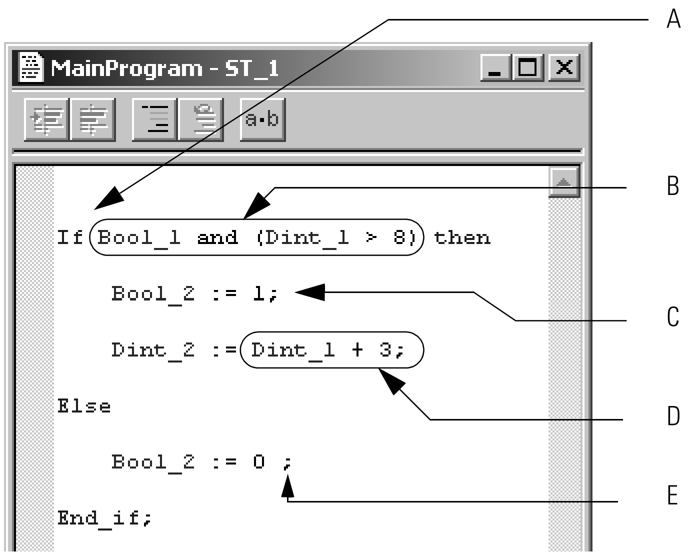


7. Select the update period for the control and click OK.
8. To exit design mode and use the control, click here.



Enter Structured Text

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



```
If Bool_1 and (Dint_1 > 8) then
    Bool_2 := 1;
    Dint_2 := Dint_1 + 3;
Else
    Bool_2 := 0 ;
End_if;
```

Item	Description
A	Construct defines logical conditions for the execution of other structured text code (other statements). In this example, the construct is If…Then…Else…End_if.
B	BOOL expression checks 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).
C	Assignment writes a value to a tag. The value moves from the right side of the := to the left side.
D	Numeric expression calculates a value.
E	Semicolon ';' terminates an assignment, instruction, or end of a construct.

As you enter structured text, follow these guidelines:

Guideline	Description	
Structured text is not 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.	
Use tabs, spaces, and carriage returns (separate lines) to make your structured text easier to read.	Tabs, spaces, and carriage returns have no effect on the execution of the structured text.	
	This	**Executes the same as this**

	`If Bool1 then`
	`    Bool2 := 1;`
	`End_if;`
		`If Bool1 then Bool2 := 1; End_if;`

	`Bool2 := 1;`	`Bool2:=1;`

Guideline	Description
Write BOOL expressions as either true or false	Use a BOOL expression to determine if specific conditions are true (1) or false (0). • A BOOL tag is already true (1) or false (0). *Do not* use an "=" sign to check its state.

This is OK	This is NOT OK
`If Bool1 …`	`If Bool1 = 1 …`
`If Not(Bool2) …`	`If Bool2 = 0 …`

• To check an integer, REAL, or string, make a comparison (=, <, <=, >, >=, <>).

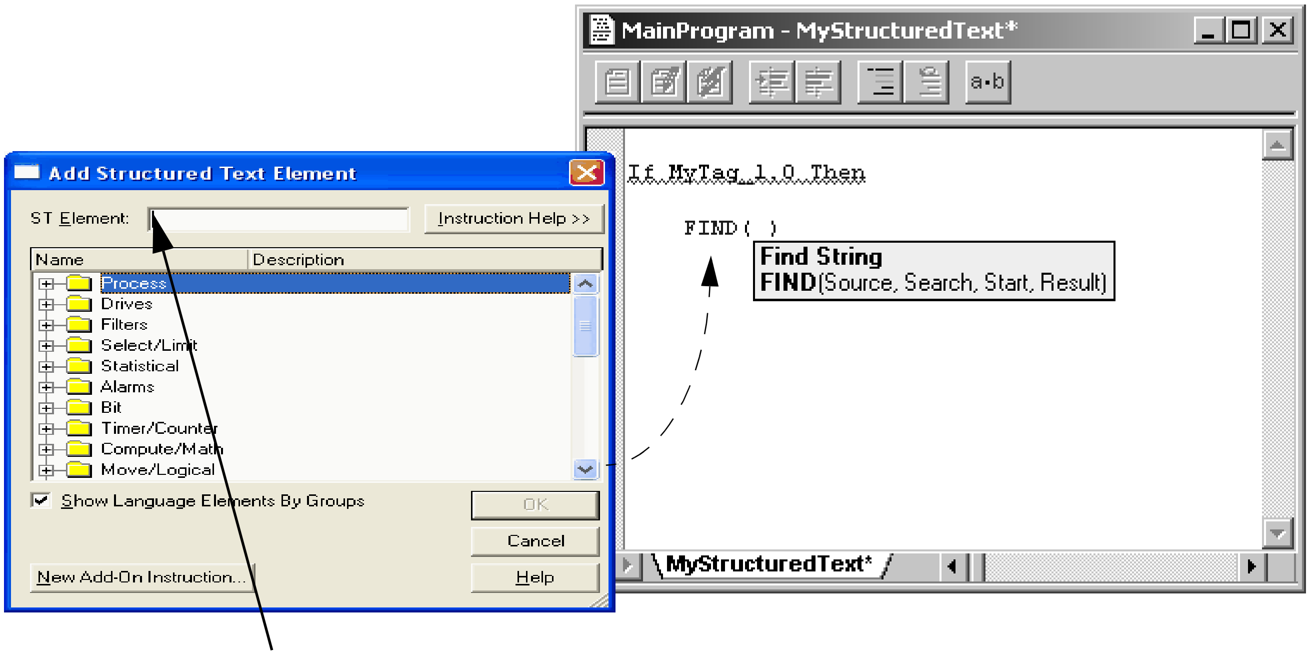
This is OK	This is NOT OK
`If Dint1 > 5 …`	`If Dint1 …`

Guideline	Description
For an assignment, start with the destination.	Write an assignment as follows: `Destination := Source;` $\xleftarrow{\text{data}}$

Browse For an Instruction

1. Press Alt + Insert.

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

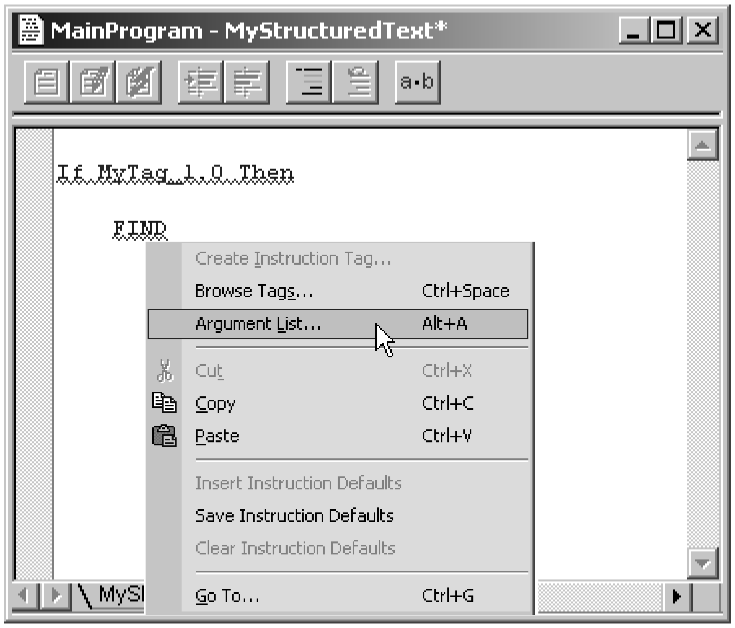


```
If MyTag_1.0 Then

   FIND( )
```

Find String
FIND(Source, Search, Start, Result)

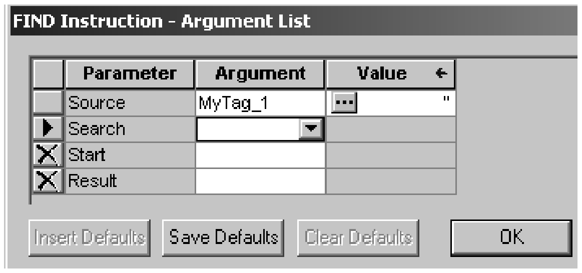
Assign Operands to an Instruction

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


MainProgram - MyStructuredText*
Create Instruction Tag...
Browse Tags... Ctrl+Space
Argument List... Alt+A
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Insert Instruction Defaults
Save Instruction Defaults
Clear Instruction Defaults
Go To... Ctrl+G
MySt

```
If MyTag_1.0 Then

    FIND
```

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

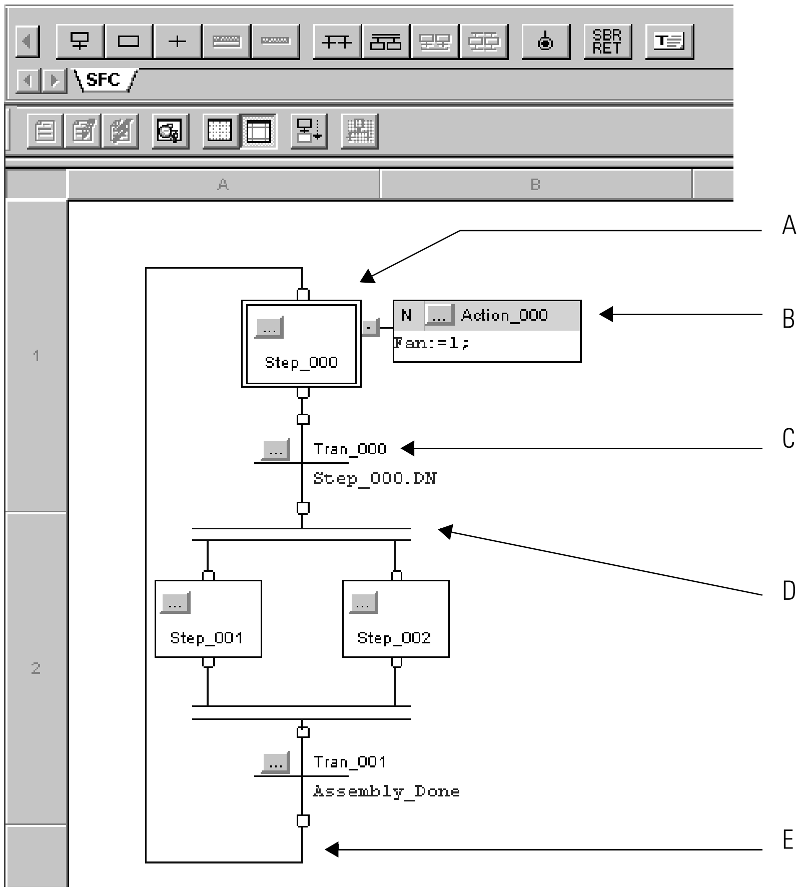


Parameter	Argument	Value
Source	MyTag_1	''
Search		
Start		
Result		

3. Click OK.

Enter a Sequential Function Chart

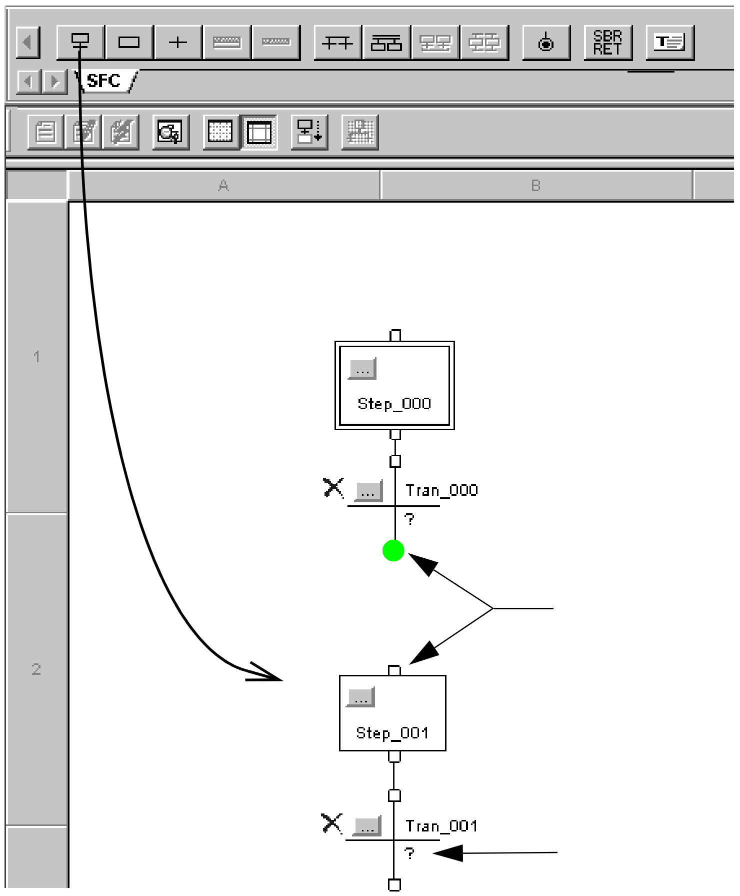
A sequential function chart (SFC) lets you define a sequence of states (steps) through which your machine or process progresses. The steps can execute structured text, call subroutines, or simply serve as signals for other logic.



Item	Description
A	Step is major function of your process. It contains the actions that occur at a particular time, phase, or station.
B	Action is one of the functions that a step performs. To program the action, either enter structured text or call a subroutine.
C	Transition is a 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.
D	Branch executes more than 1 step at the same time (simultaneous) or choose between different steps (selective).
E	Wire connects 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.

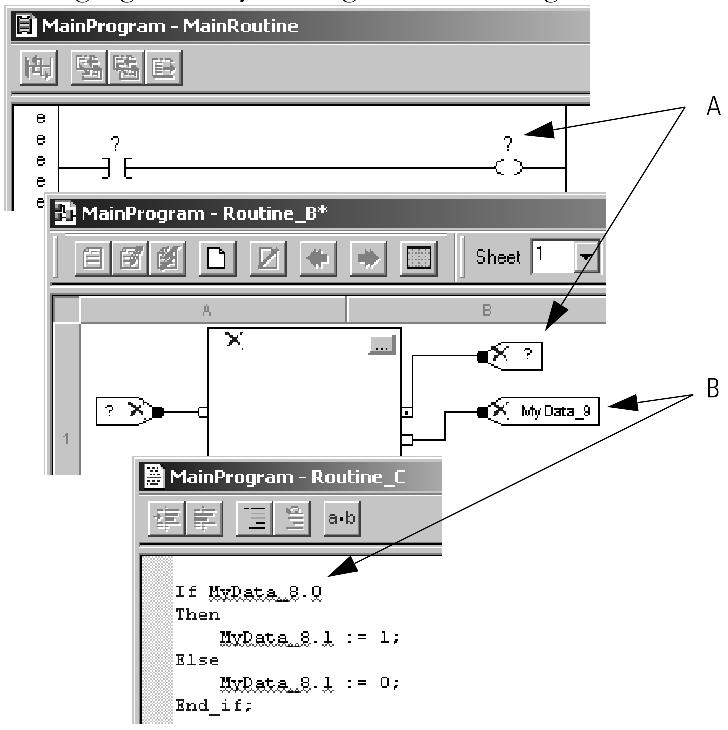


2. To manually connect elements, click corresponding pins. A green dot shows a valid connection point

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.



```
If MyData_8.0
Then
    MyData_8.1 := 1;
Else
    MyData_8.1 := 0;
End_if;
```

Item	Description
A	This instruction is missing an operand. You can enter and save logic without assigning operands. This lets you develop your logic in iterations and save libraries of code for re-use.
B	This is an undefined tag. You can enter and save logic without defining all the tags. This lets you develop your logic in iterations.

A tag name follows this format:

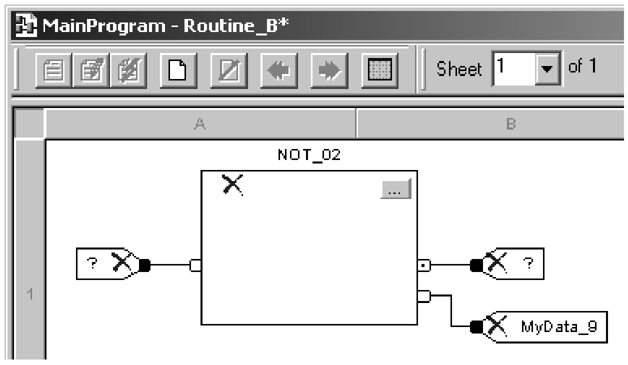
Name ▨ *[Element]* ▨ *.Member* ▨ *[Element]* ▨ *.Bit* or ▨ *.[Index]*

▨ = Optional

Where	Is
`Name`	Name that identifies this specific tag.
`Element`	Subscript or subscripts that point to a specific element within an array. • 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. • 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

1. Double-click the tag area.



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

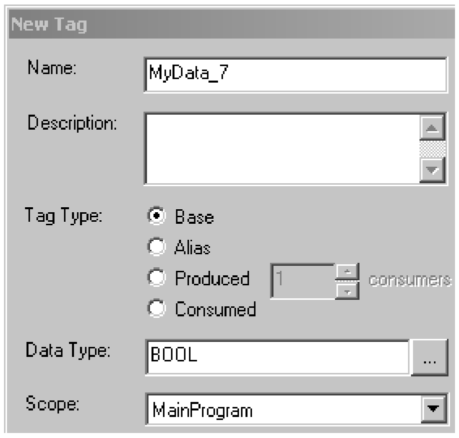
   Use underscores '_' in place of spaces.

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

4. Type the data type.

   To browse for a data type or assign array dimensions, click ⋯.

5. Choose the scope for the tag.


New Tag

Name: MyData_7

Description:

Tag Type: (●) Base ( ) Alias ( ) Produced 1 consumers ( ) Consumed

Data Type: BOOL

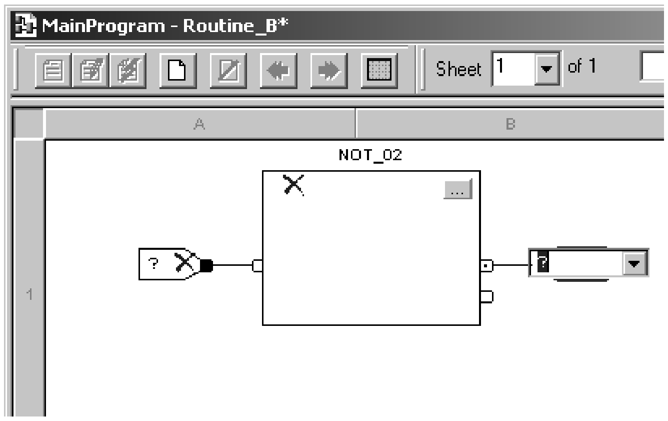
Scope: MainProgram

6. Click OK.

Select an Existing Tag

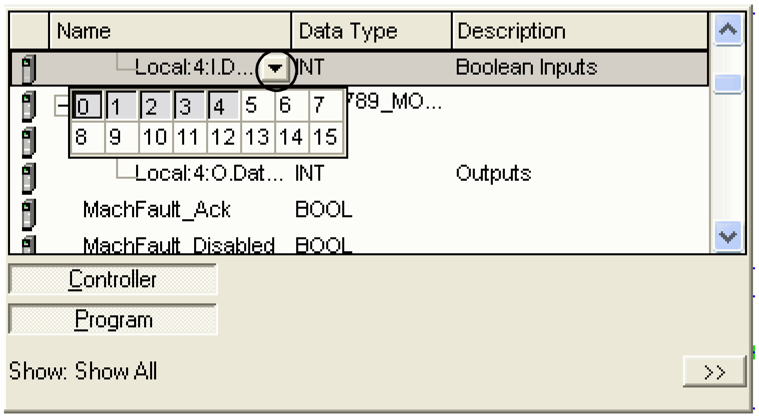
1. Double-click the tag area.

2. Click the ▼.



3. Select the desired tag.

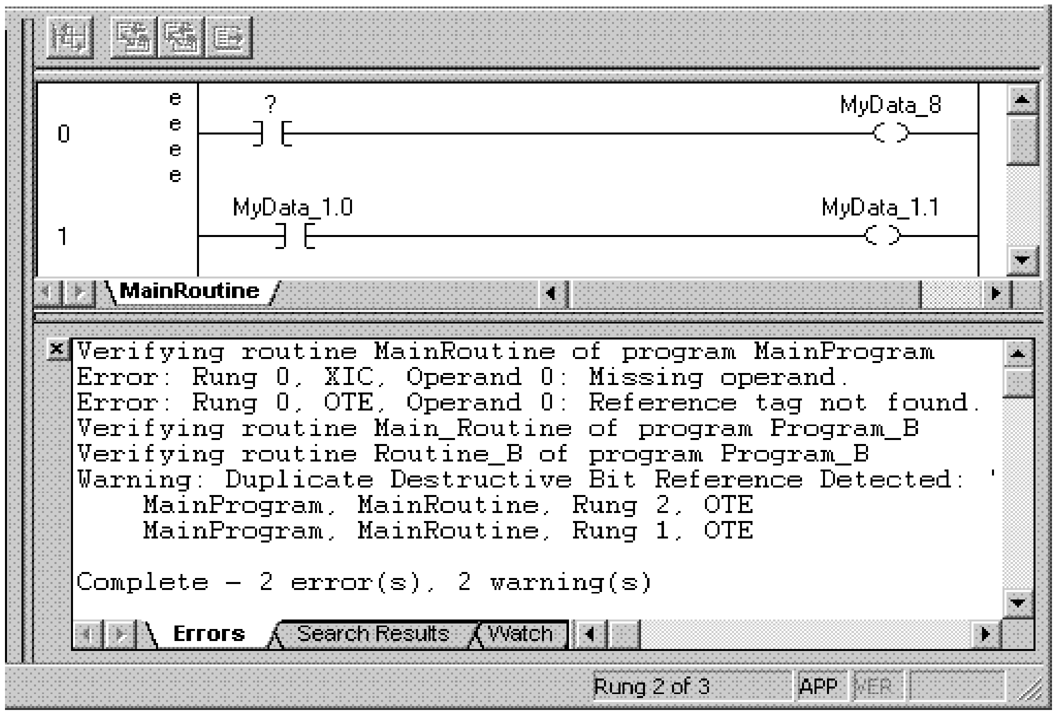
To select a bit, click the ▼.



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.

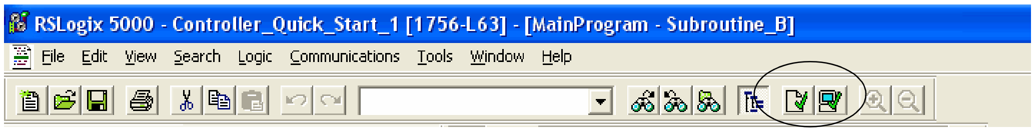


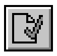
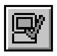
```
Verifying routine MainRoutine of program MainProgram
Error: Rung 0, XIC, Operand 0: Missing operand.
Error: Rung 0, OTE, Operand 0: Reference tag not found.
Verifying routine Main_Routine of program Program_B
Verifying routine Routine_B of program Program_B
Warning: Duplicate Destructive Bit Reference Detected: '
    MainProgram, MainRoutine, Rung 2, OTE
    MainProgram, MainRoutine, Rung 1, OTE

Complete - 2 error(s), 2 warning(s)
```

Item	Description
Verify	Check a routine or project for programming errors or incomplete configuration.
Warning	A 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	A 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: • You enable duplicate destructive bit detection. (It's off by default.) • You use the bit in a ladder logic OTE, ONS, OSF, or OSR instruction. • 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.

Follow these steps to verify a routine or project.

1. Choose a verify option.


RSLogix 5000 - Controller_Quick_Start_1 [1756-L63] - [MainProgram - Subroutine_B]
File Edit View Search Logic Communications Tools Window Help

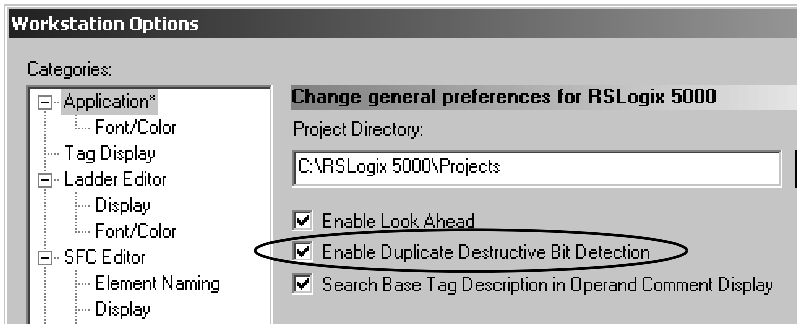
To go to	Click this
Verify routine in view	
Verify entire project	

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. Close the Errors tab.

4. To turn off duplicate destructive bit detection (it's on by default), from the Tools menu, choose Options.



Change general preferences for RSLogix 5000

Project Directory: C:\RSLogix 5000\Projects

- [✓] Enable Look Ahead
- [✓] Enable Duplicate Destructive Bit Detection
- [✓] Search Base Tag Description in Operand Comment Display

Guidelines for Tags

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

Guideline	Details
Create user-defined data types.	User-defined data types (structures) let you organize your data to match your machine or process. A user-defined data type provides these advantages: • 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. 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.
Use arrays to quickly create a group of similar tags.	An array creates multiple instances of a data type under a common tag name. • 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. 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. **Important:** Minimize the use of BOOL arrays. Many array instructions do not operate on BOOL arrays. This makes it more difficult to initialize and clear an array of BOOL data. • 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.

Guideline	Details	
Take advantage of program-scoped tags.	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. 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. Certain tags must be controller scope (controller tag).	
	If you want to use the tag	**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)
For integers, use the DINT data type.	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. • 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.	

Guideline	Details
Limit a tag name to 40 characters.	Here are the rules for a tag name: • 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
Use mixed case.	Although tags are not case sensitive (upper case *A* is the same as lower case *a*), mixed case is easier to read.

These tags are easier to read	Than these tags
Tank_1	TANK_1
Tank1	TANK1
	tank_1
	tank1

Guideline	Details
Consider the alphabetical order of tags.	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.

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, and so on.

Document a Project

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

What You Need

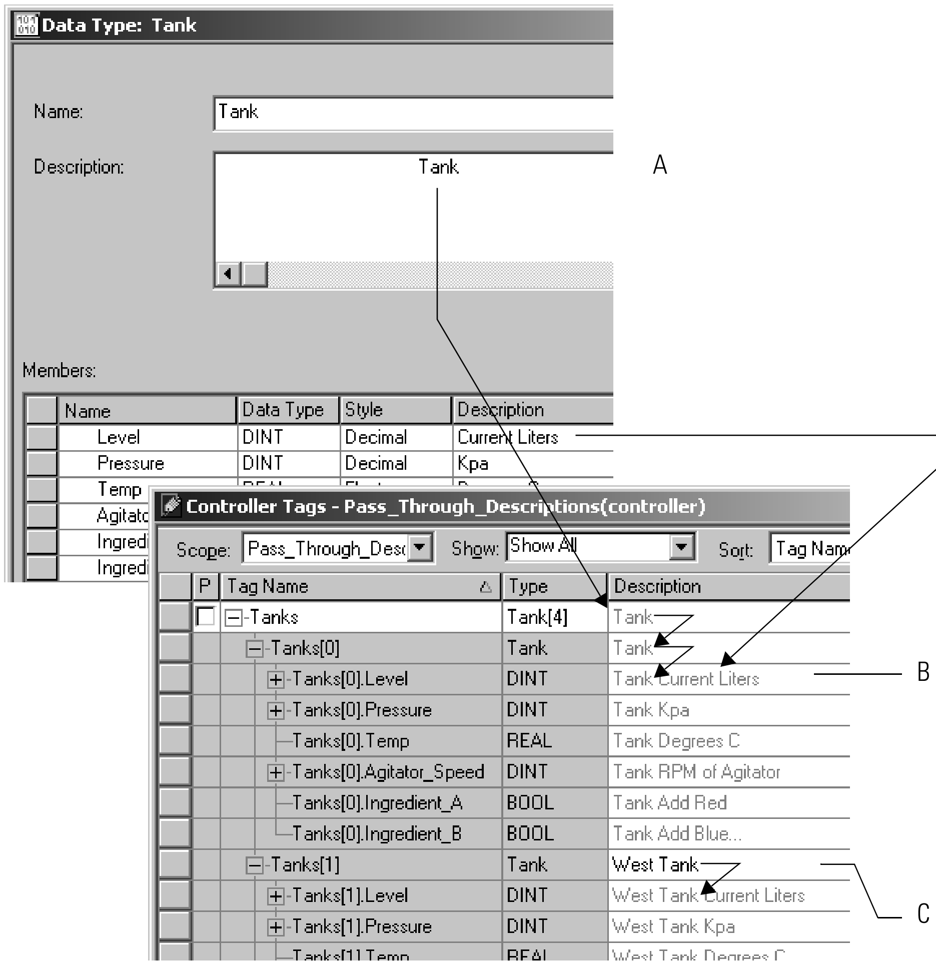
You need these items to complete the tasks in this manual:
- Personal computer running RSLogix 5000 software, version 16
- The project you are documenting

Follow These Steps

1. Describe a user-defined data type (page 98).
2. Add rung comments (page 101).
3. Enter and edit rung comments using Microsoft Excel (page 102).
4. Add comments to a function block diagram or SFC (page 105).
5. Add comments to structured text (page 107).

User-defined Data Type

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:



Name: Tank

Description: Tank

Members:

Name	Data Type	Style	Description
Level	DINT	Decimal	Current Liters
Pressure	DINT	Decimal	Kpa
Temp			
Agitato			
Ingredi			
Ingredi			

Scope: Pass_Through_Desc Show: Show All Sort: Tag Nam

P	Tag Name	Type	Description
☐	⊟-Tanks	Tank[4]	Tank
	⊟-Tanks[0]	Tank	Tank
	⊞-Tanks[0].Level	DINT	Tank Current Liters
	⊞-Tanks[0].Pressure	DINT	Tank Kpa
	Tanks[0].Temp	REAL	Tank Degrees C
	⊞-Tanks[0].Agitator_Speed	DINT	Tank RPM of Agitator
	Tanks[0].Ingredient_A	BOOL	Tank Add Red
	Tanks[0].Ingredient_B	BOOL	Tank Add Blue...
	⊟-Tanks[1]	Tank	West Tank
	⊞-Tanks[1].Level	DINT	West Tank Current Liters
	⊞-Tanks[1].Pressure	DINT	West Tank Kpa
	Tanks[1].Temp	REAL	West Tank Degrees C

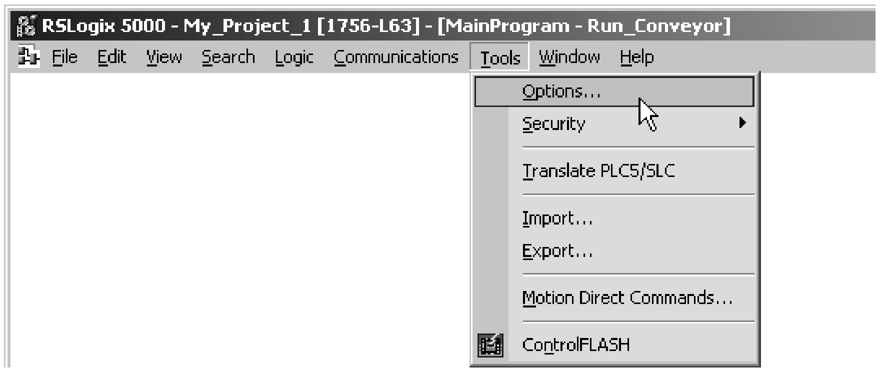
Item	Description
A	Pass through of descriptions are automatically created when possible, RSLogix 5000 software looks for an available description for a tag, element, or member. • Descriptions in user-defined data types ripple through to the tags that use that data type. • Description of an array tag ripples through to the elements and members of the array.
B	Append descriptions to base tags. 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.
C	Edit pass-through descriptions so that you can 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

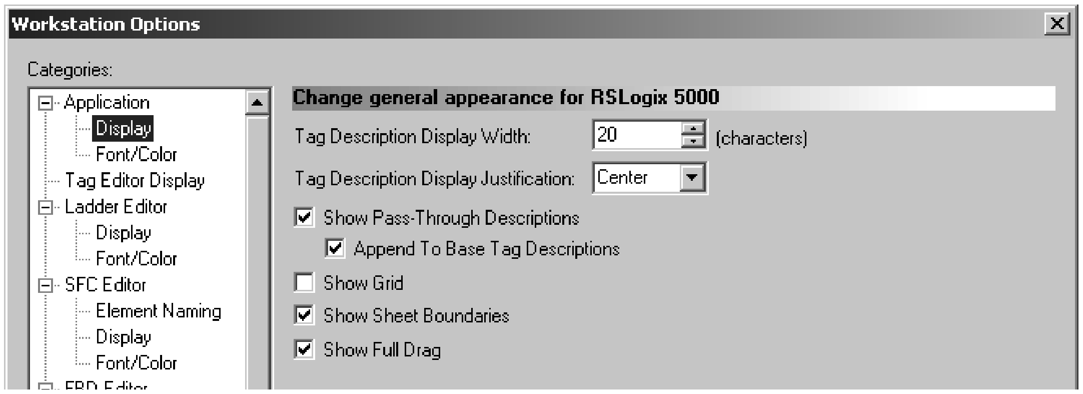
Turn Pass-Through and Append Descriptions On or Off

1. In RSLogix 5000 software, from the Tools menu, choose Options.


RSLogix 5000 - My_Project_1 [1756-L63] - [MainProgram - Run_Conveyor]
File Edit View Search Logic Communications Tools Window Help
Options...
Security
Translate PLC5/SLC
Import...
Export...
Motion Direct Commands...
ControlFLASH

2. Select the Display.

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



Change general appearance for RSLogix 5000

Tag Description Display Width: 20 (characters)

Tag Description Display Justification: Center

[✓] Show Pass-Through Descriptions

[✓] Append To Base Tag Descriptions

[ ] Show Grid

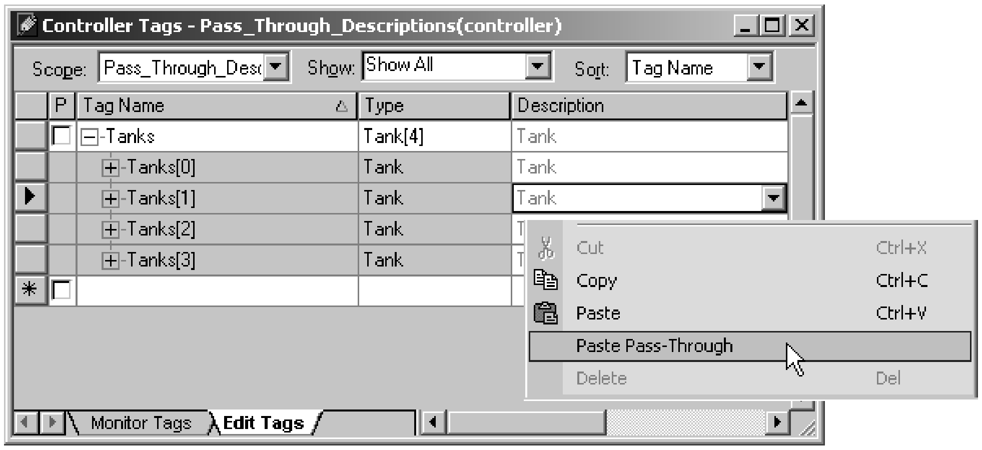
[✓] Show Sheet Boundaries

[✓] Show Full Drag

Paste a Pass-Through Description

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

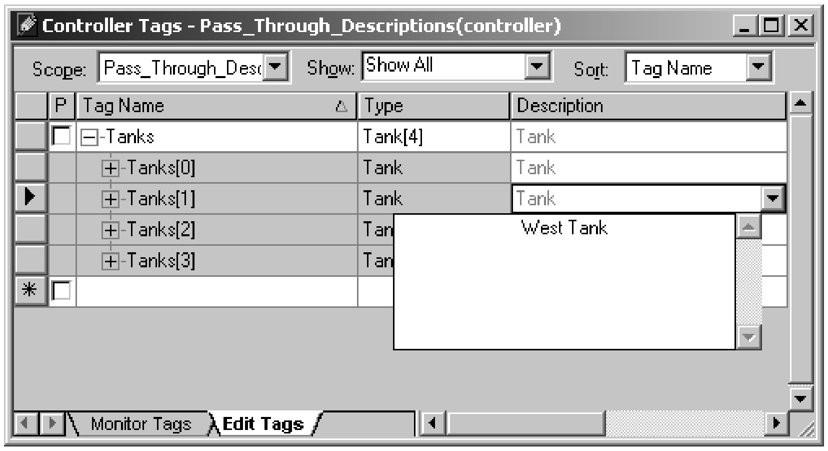
1. Right-click the pass-through description and choose Paste Pass-Through.



Scope: Pass_Through_Desc Show: Show All Sort: Tag Name

P	Tag Name	Type	Description
☐	-Tanks	Tank[4]	Tank
	+-Tanks[0]	Tank	Tank
	+-Tanks[1]	Tank	Tank
	+-Tanks[2]	Tank	T
	+-Tanks[3]	Tank	T
☐			

2. Edit the description and press Ctrl + Enter.



Scope: Pass_Through_Desc Show: Show All Sort: Tag Name

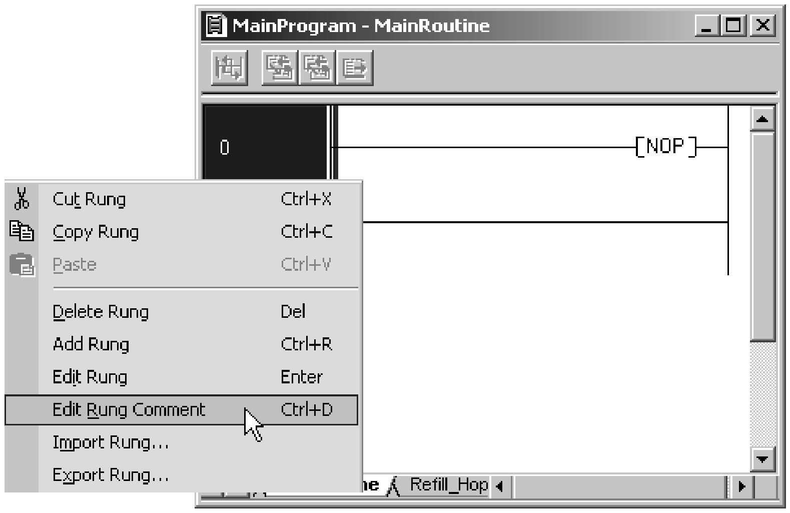
P	Tag Name	Type	Description
☐	-Tanks	Tank[4]	Tank
	+-Tanks[0]	Tank	Tank
	+-Tanks[1]	Tank	Tank
	+-Tanks[2]	Tan	
	+-Tanks[3]	Tan	
☐			

West Tank

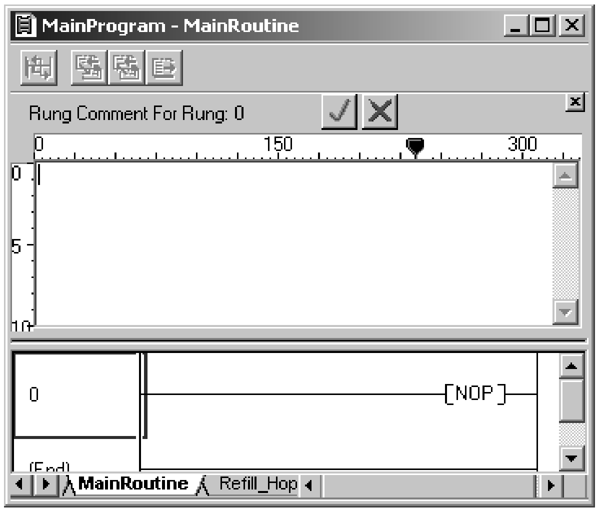
Add Rung Comments

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. Click the check to save the comment and close the entry window.

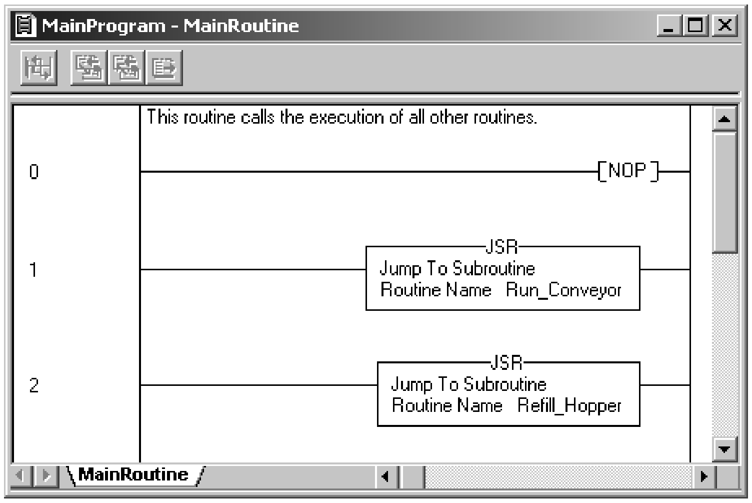
Rung Comments Using Microsoft Excel

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.

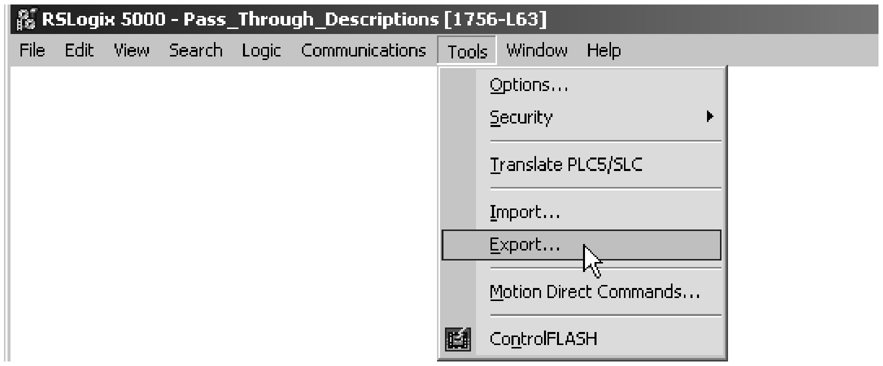
IMPORTANT Rung comments export in the CSV (comma delimited) format. Make sure you keep that format when you save and close the export file.

Export the Existing Comments

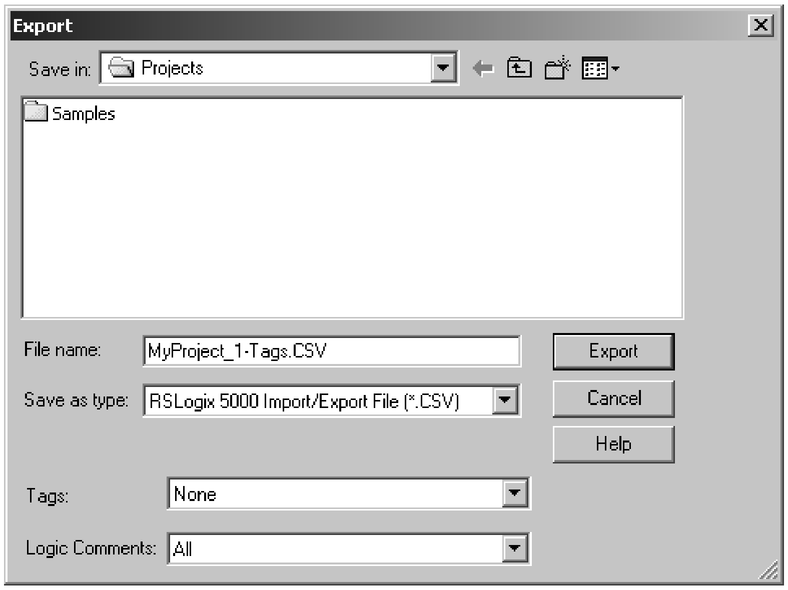
1. In RSLogix 5000 software, add at least one rung comment. This helps to format the export file.


MainProgram - MainRoutine
This routine calls the execution of all other routines.
0
NOP
1
JSR
Jump To Subroutine
Routine Name Run_Conveyor
2
JSR
Jump To Subroutine
Routine Name Refill_Hopper
MainRoutine

2. From the Tools menu, choose Export.


RSLogix 5000 - Pass_Through_Descriptions [1756-L63]
File Edit View Search Logic Communications Tools Window Help
Options...
Security
Translate PLC5/SLC
Import...
Export...
Motion Direct Commands...
ControlFLASH

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



File name: MyProject_1-Tags.CSV

Save as type: RSLogix 5000 Import/Export File (*.CSV)

Tags: None

Logic Comments: All

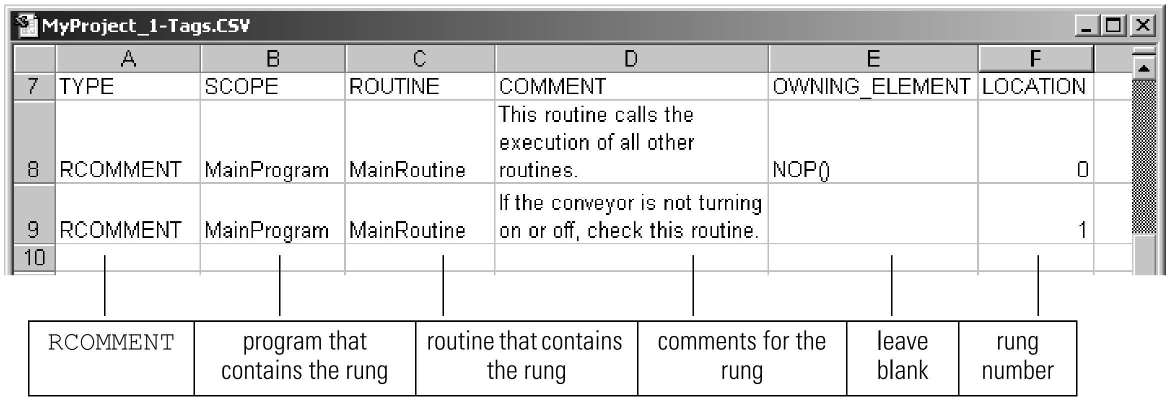
4. Choose what to export.

5. Click Export.

Edit the Export File

1. In Microsoft Excel software, open the export file.

2. Enter rung comments.



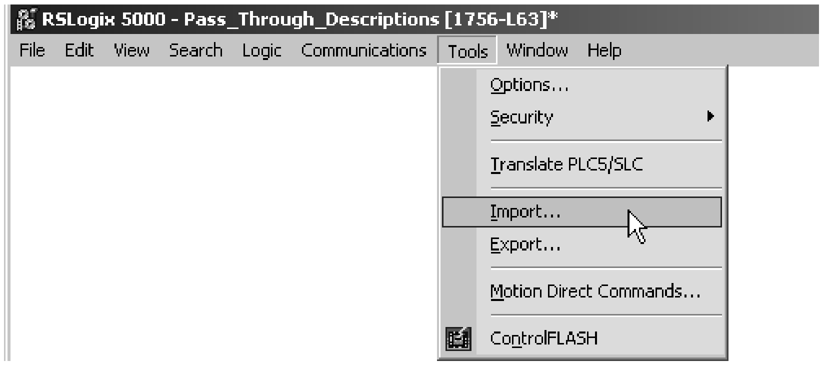
	A	B	C	D	E	F
7	TYPE	SCOPE	ROUTINE	COMMENT	OWNING_ELEMENT	LOCATION
8	RCOMMENT	MainProgram	MainRoutine	This routine calls the execution of all other routines.	NOP()	0
9	RCOMMENT	MainProgram	MainRoutine	If the conveyor is not turning on or off, check this routine.		1
10						

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

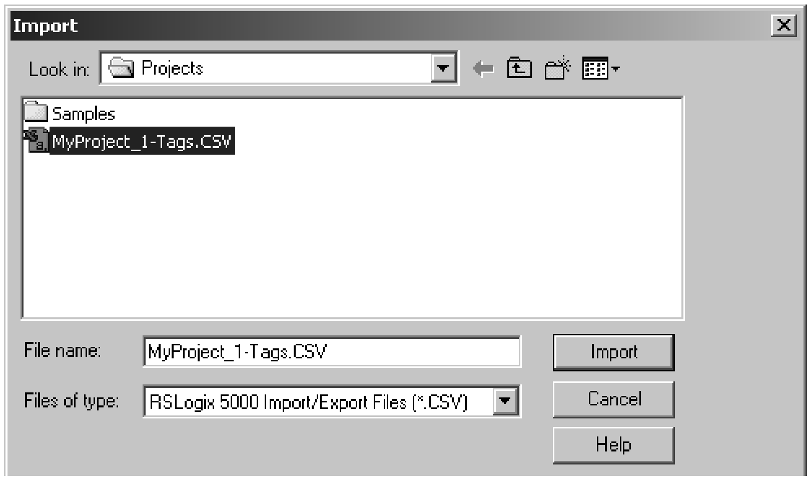
4. Save and close the file (keep it in the CSV format).

Import the New Comments

1. In RSLogix 5000 software, from the Tools menu, choose Import.

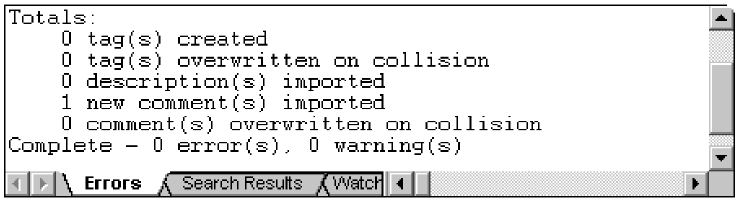

RSLogix 5000 - Pass_Through_Descriptions [1756-L63]*
File Edit View Search Logic Communications Tools Window Help
Options...
Security
Translate PLC5/SLC
Import...
Export...
Motion Direct Commands...
ControlFLASH

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


Import
Look in: Projects
Samples
MyProject_1-Tags.CSV
File name: MyProject_1-Tags.CSV
Files of type: RSLogix 5000 Import/Export Files (*.CSV)
Import
Cancel
Help

3. Click Import.

4. 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.



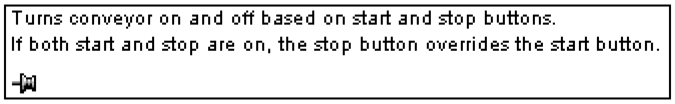
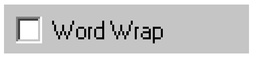
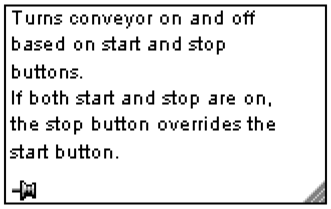
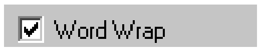
```
Totals:
    0 tag(s) created
    0 tag(s) overwritten on collision
    0 description(s) imported
    1 new comment(s) imported
    0 comment(s) overwritten on collision
Complete - 0 error(s), 0 warning(s)
```

Comments in a Function Block Diagram or SFC

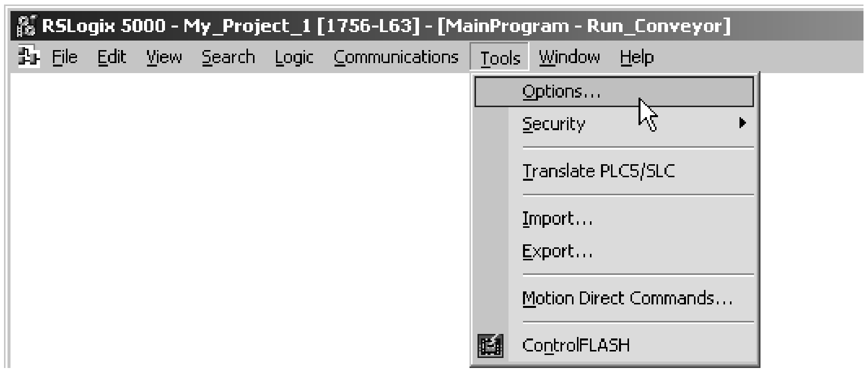
Use Text boxes to add notes about the diagram or chart in general or a specific element. Or use a text box to capture information that you will use later on as you develop the project.

Set the Word Wrap Option

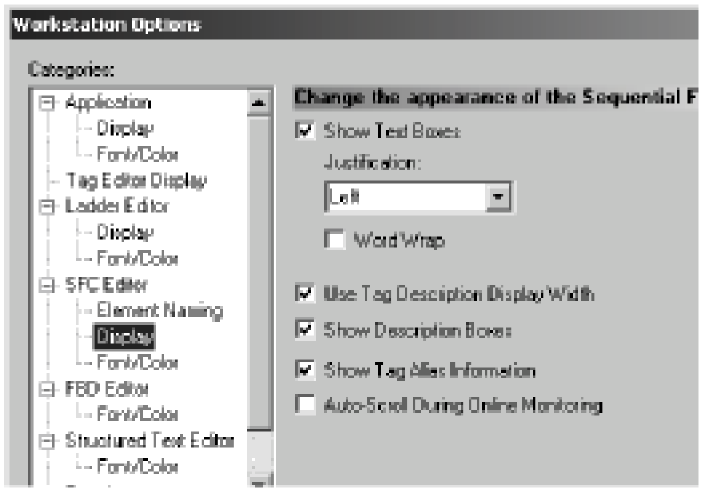
Use the word wrap option to control the width of the text box as you type. You set the option for function block diagrams and SFC independent of each other.

If you want text boxes to	Choose this option
Automatically grow to the width of the longest line of text in the box.  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.	 ☐ Word Wrap
Retain a fixed width and wrap the text. You can always manually resize the box.  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.	 ☑ Word Wrap

1. In RSLogix 5000 software, from the Tools menu, choose Options.


RSLogix 5000 - My_Project_1 [1756-L63] - [MainProgram - Run_Conveyor]
File Edit View Search Logic Communications Tools Window Help
Options...
Security
Translate PLC5/SLC
Import...
Export...
Motion Direct Commands...
ControlFLASH

2. Select the editor.


Workstation Options

Categories:
- Application
  - Display
  - Font/Color
- Tag Editor Display
- Ladder Editor
  - Display
  - Font/Color
- SFC Editor
  - Element Naming
  - Display
  - Font/Color
- FBD Editor
  - Font/Color
- Structured Text Editor
  - Font/Color

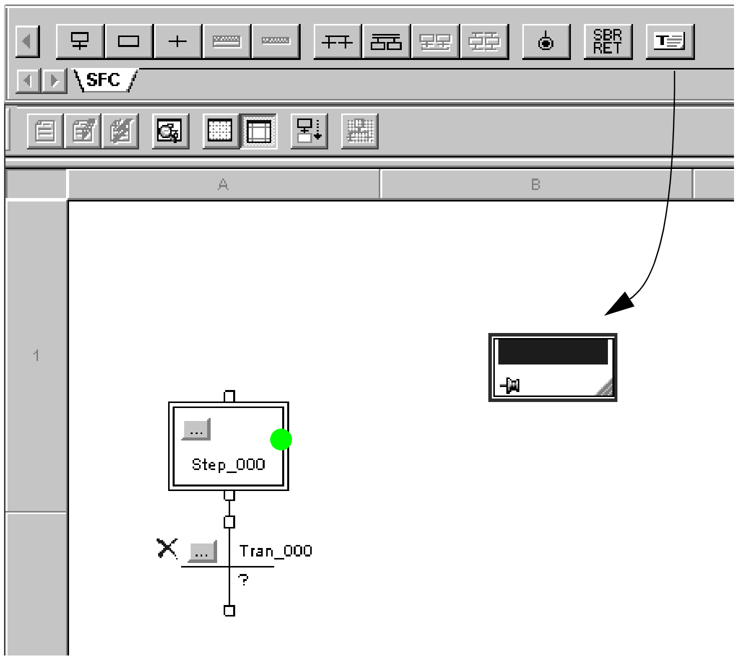
Change the appearance of the Sequential F

- [✓] Show Text Boxes
- Justification: Left
- [ ] Word Wrap
- [✓] Use Tag Description Display Width
- [✓] Show Description Boxes
- [✓] Show Tag Alias Information
- [ ] Auto-Scroll During Online Monitoring

3. Select or clear the word wrap option.

Add a Text Box

1. Drag the text box button from the toolbar to the chart.



2. Type the comment and press Ctrl + Enter.

3. To attach the text box to a specific element, click the pin symbol and then the corresponding element. A green dot shows a valid connection point.

Comments in 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.

Follow these steps to add comments to your structured text.

To add a comment	Use one of these formats
On a single line	`//`*comment*

At the end of a line of structured text	`(*`*comment*`*)`
	`/*`*comment*`*/`

Within a line of structured text	`(*`*comment*`*)`
	`/*`*comment*`*/`

That spans more than one line	`(*`*start of comment . . . end of comment*`*)`
	`/*`*start of comment . . . end of comment*`*/`

Here is an example.

Format	Example

`//`*comment*	**At the beginning of a line**
	`//Check conveyor belt direction`
	`IF conveyor_direction THEN...`
	
	At the end of a line
	`ELSE //If conveyor isn’t moving, set alarm light`
	`light := 1;`
	`END_IF;`

`(*`*comment*`*)`	`Sugar.Inlet[:=]1;(*open the inlet*)`
	
	`IF Sugar.Low (*low level LS*)& Sugar.High (*high level LS*)THEN...`
	
	`(*Controls the speed of the recirculation pump. The speed depends on the temperature in the tank.*)`
	`IF tank.temp > 200 THEN...`

`/*`*comment*`*/`	`Sugar.Inlet:=0;/*close the inlet*/`
	
	`IF bar_code=65 /*A*/ THEN...`
	
	`/*Gets the number of elements in the Inventory array and stores the value in the Inventory_Items tag*/`
	`SIZE(Inventory,0,Inventory_Items);`

Language Switching

With RSLogix 5000 software, version 17, you have the option to display project documentation, such as tag descriptions and rung comments for any supported localized language. You can store project documentation for multiple languages in a single project file rather than in language-specific project files. You define all the localized languages that the project will support and set the current, default, and optional custom localized language. The software uses the default language if the current language's content is blank for a particular component of the project. However, you can use a custom language to tailor documentation to a specific type of project file user.

Enter the localized descriptions in your RSLogix 5000 project, either when programming in that language or by using the import/export utility to translate the documentation off-line and then import it back into the project. Once you enable language switching in RSLogix 5000 software, you can dynamically switch between languages as you use the software.

Project documentation that supports multiple translations within a project includes:

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

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

Go Online to the Controller

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

What You Need

You need these items to complete the tasks in this manual:
- Personal computer running RSLogix 5000 software, version 16 or later, and RSLinx software
- The physical system to which you are connecting
- EtherNet/IP cabling
- EtherNet/IP communication card(s) for the for the module(s) in our sample project
- The project you want to access

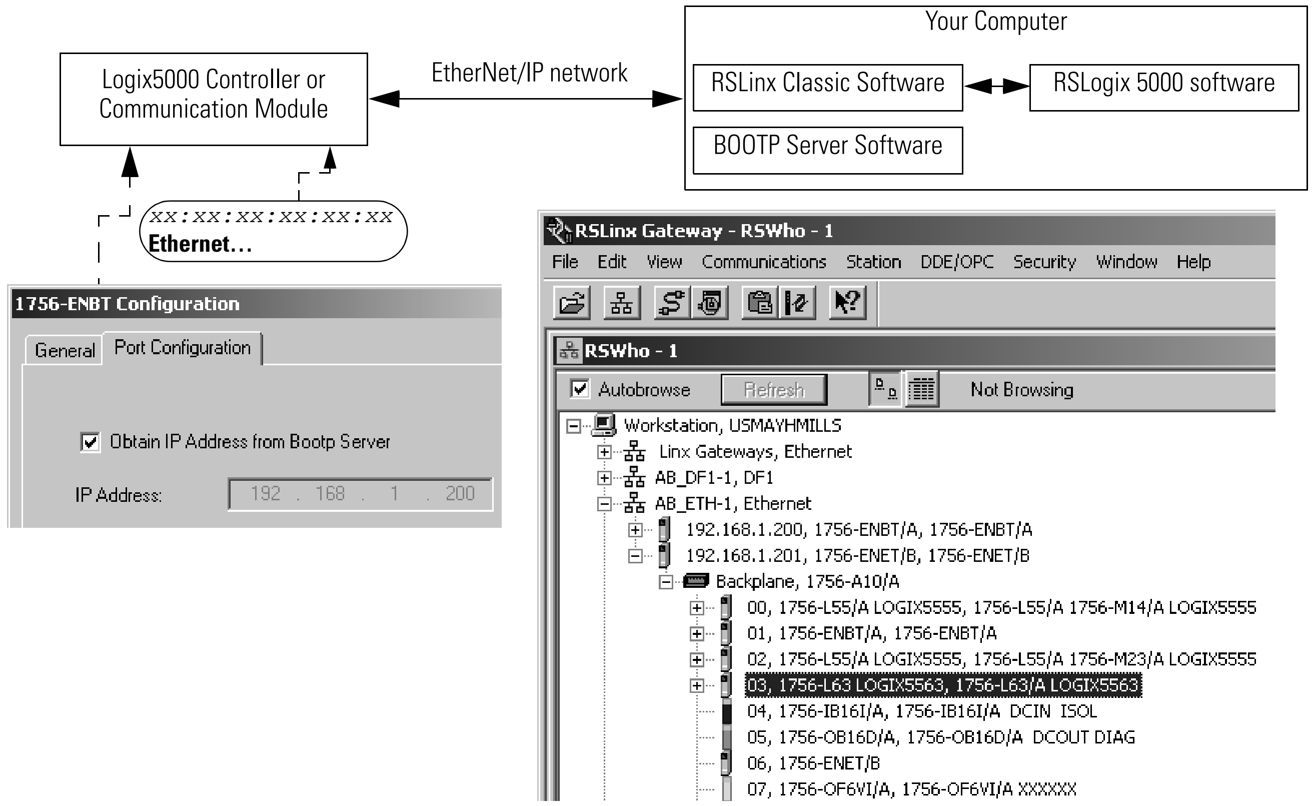
Follow These Steps

1. Establish EtherNet/IP communication with the controller (page 110)

2. Go online to a controller (page 116)

Establish EtherNet/IP Communication with the Controller

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



☑ Obtain IP Address from Bootp Server

IP Address: 192 . 168 . 1 . 200

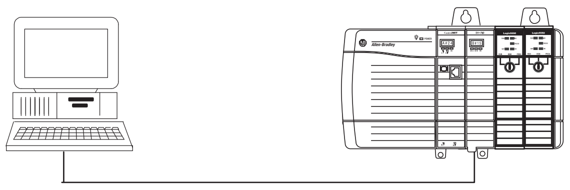
```
Workstation, USMAYHMILLS
  Linx Gateways, Ethernet
  AB_DF1-1, DF1
  AB_ETH-1, Ethernet
    192.168.1.200, 1756-ENBT/A, 1756-ENBT/A
    192.168.1.201, 1756-ENET/B, 1756-ENET/B
      Backplane, 1756-A10/A
        00, 1756-L55/A LOGIX5555, 1756-L55/A 1756-M14/A LOGIX5555
        01, 1756-ENBT/A, 1756-ENBT/A
        02, 1756-L55/A LOGIX5555, 1756-L55/A 1756-M23/A LOGIX5555
        03, 1756-L63 LOGIX5563, 1756-L63/A LOGIX5563
        04, 1756-IB16I/A, 1756-IB16I/A  DCIN  ISOL
        05, 1756-OB16D/A, 1756-OB16D/A  DCOUT DIAG
        06, 1756-ENET/B
        07, 1756-OF6VI/A, 1756-OF6VI/A XXXXXX
```

Item	Description
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. RSLinx 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.

2. Determine if your EtherNet/IP network is connected to the Internet or if it is a standalone network that does not connect to the Internet?



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

Obtain this	**If your network is connected to the Internet, from this source**	**If your network is a standalone network that does not connect to the Internet, from this source**
Ethernet address	Sticker on the device	Sticker on the device
IP address	Network administrator	192.168.1.x, where x = any value between 1 and 254[1]
Subnet mask		255.255.255.0[2]
Gateway address (may not be required)		Not needed

[1] In this case, your computer must use an IP address that is close to the EtherNet/IP device's IP address. For example, if the EtherNet/IP device uses the 192.168.1.*x* addressing, the computer must also use that addressing but with a different *x* value.

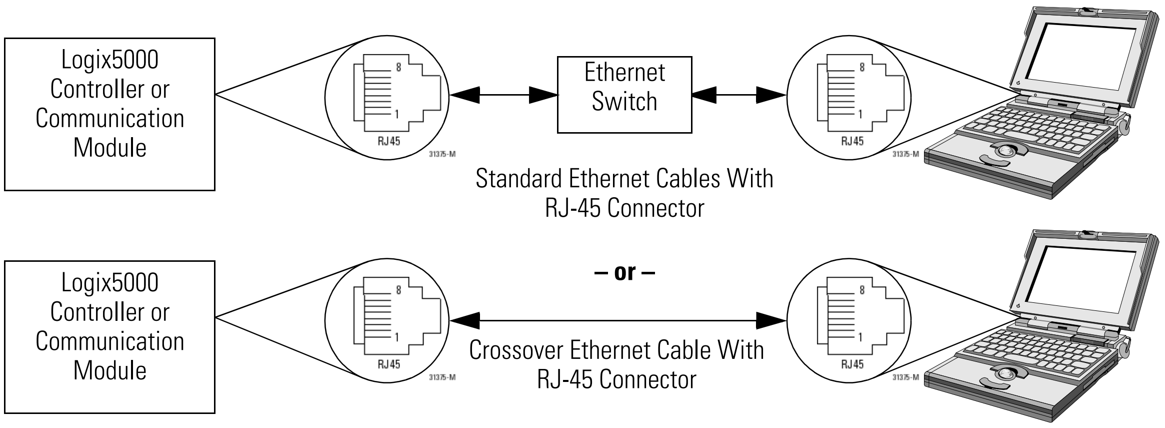
[2] In this case, your computer must use the same subnet mask value as the EtherNet/IP device.

Connect Your EtherNet/IP Device and Computer

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

ATTENTION

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.


Logix5000 Controller or Communication Module
RJ45
Ethernet Switch
RJ45
Standard Ethernet Cables With RJ-45 Connector
– or –
Logix5000 Controller or Communication Module
RJ45
Crossover Ethernet Cable With RJ-45 Connector
RJ45

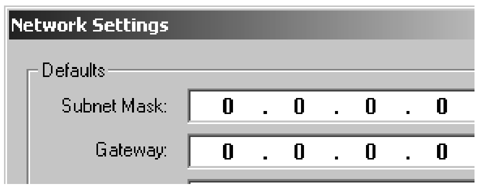
Assign an IP Address to the Controller or Communication Module

Follow these steps if you do not have a serial connection to the controller.

1. Start BOOTP server software by either of the following:

- Start > Programs > Rockwell Software > BOOTP-DHCP Server > BOOTP-DHCP Server
- 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 click OK.

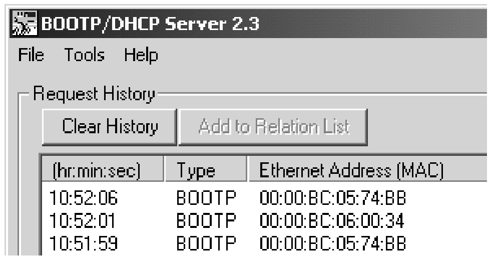

Network Settings

Defaults

Subnet Mask: 0 . 0 . 0 . 0

Gateway: 0 . 0 . 0 . 0

3. Double-click the Ethernet address of the controller/communication module.

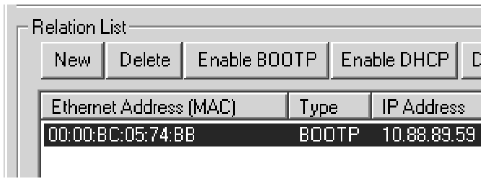


```
(hr:min:sec)  Type   Ethernet Address (MAC)
10:52:06      BOOTP  00:00:BC:05:74:BB
10:52:01      BOOTP  00:00:BC:06:00:34
10:51:59      BOOTP  00:00:BC:05:74:BB
```

4. Type the IP address and click OK.

5. In the Relation List (lower section), select the device and click Disable BOOTP.

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

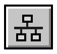


```
Ethernet Address (MAC)  Type   IP Address
00:00:BC:05:74:BB       BOOTP  10.88.89.59
```

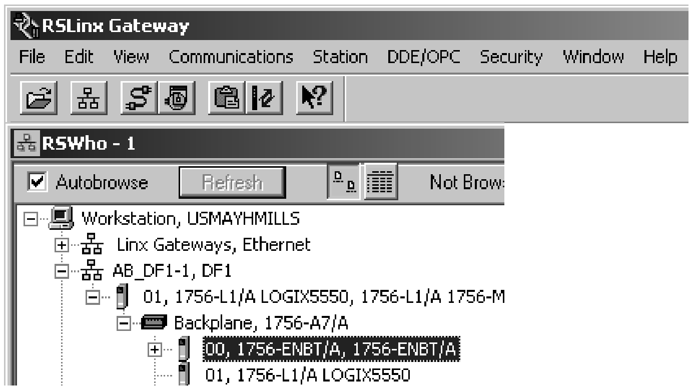
6. When you close the BOOTP server software, you are prompted to save your changes.
   - If you want a record of the IP address that you assigned to the device, save the changes.
   - Regardless of whether you save the changes, the device keeps the IP address.

Follow these steps if you have a serial connection to the controller.

1. Start RSLinx software.

2. Click .

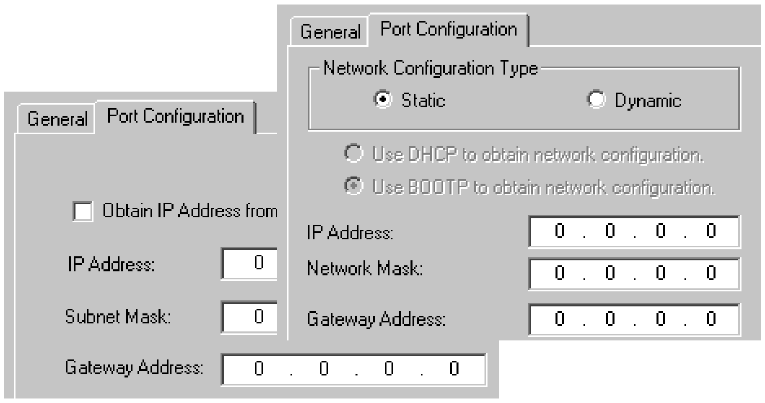
3. Browse to the EtherNet/IP device.



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.


General
Port Configuration
Obtain IP Address from
IP Address: 0
Subnet Mask: 0
Gateway Address: 0 . 0 . 0 . 0
General
Port Configuration
Network Configuration Type
Static
Dynamic
Use DHCP to obtain network configuration.
Use BOOTP to obtain network configuration.
IP Address: 0 . 0 . 0 . 0
Network Mask: 0 . 0 . 0 . 0
Gateway Address: 0 . 0 . 0 . 0

7. Type the:
   - IP address.
   - subnet mask.
   - gateway address (if required).

8. Click OK and then click Yes.

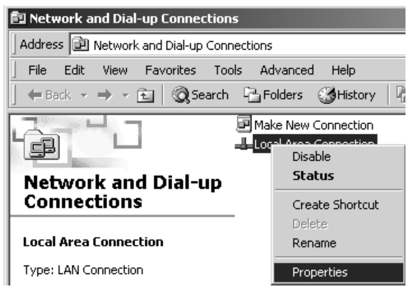
Assign an IP Address to Your Computer

If your EtherNet/IP network is a standalone network and your EtherNet/IP device uses IP address and subnet mask values, you may need to change the IP address and subnet mask values for your computer.

1. Choose Start > Settings > Network and Dial-up Connections.

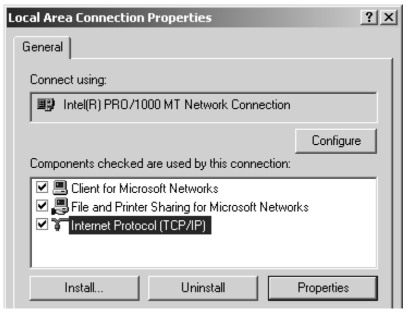
2. Right-click on Local Area Connection.

3. Choose Properties.



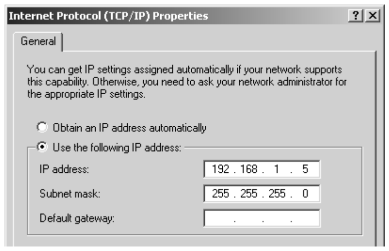
4. Select Internet Protocol (TCP/IP).

5. Choose Properties.


Local Area Connection Properties
General
Connect using:
Intel(R) PRO/1000 MT Network Connection
Configure
Components checked are used by this connection:
Client for Microsoft Networks
File and Printer Sharing for Microsoft Networks
Internet Protocol (TCP/IP)
Install...
Uninstall
Properties

6. Select Use the following IP address.

7. Change the IP address and subnet mask.


Internet Protocol (TCP/IP) Properties
General
Obtain an IP address automatically
Use the following IP address:
Default gateway:

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

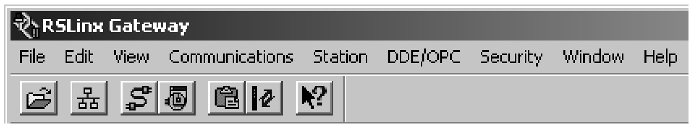
IP address: 192 . 168 . 1 . 5

Subnet mask: 255 . 255 . 255 . 0

8. Click OK.

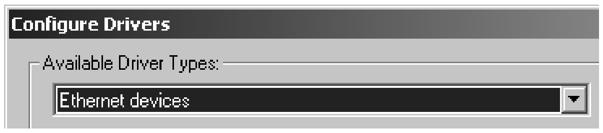
Configure an Ethernet Driver

1. Start RSLinx software.

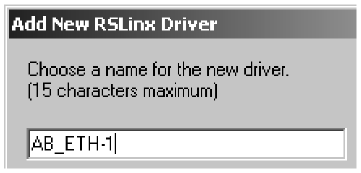


2. Click .

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



4. Accept the default name.

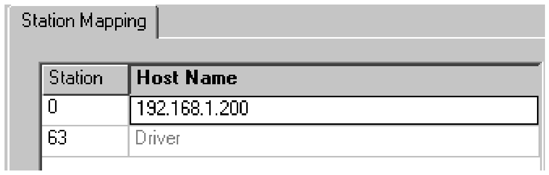


Add New RSLinx Driver

Choose a name for the new driver.
(15 characters maximum)

AB_ETH-1

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

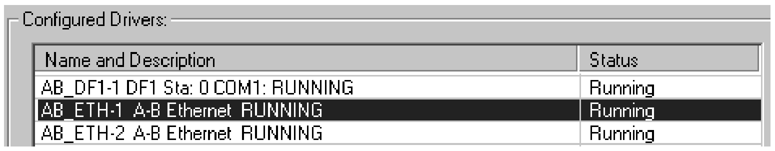


Station Mapping

Station	Host Name
0	192.168.1.200
63	Driver

6. Click OK.

   The driver is successfully configured and running.



Configured Drivers:

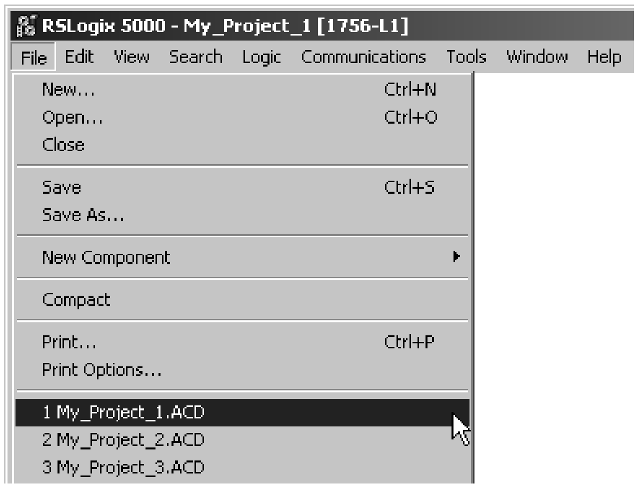
Name and Description	Status
AB_DF1-1 DF1 Sta: 0 COM1: RUNNING	Running
AB_ETH-1 A-B Ethernet RUNNING	Running
AB_ETH-2 A-B Ethernet RUNNING	Running

Online with a Controller

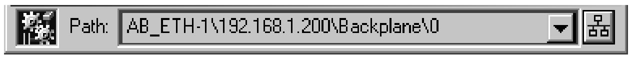
To monitor a project that is executing in a controller, go online with the controller. The procedure that you use depends on whether you have a copy of the project on your computer.

If Your Computer Has the Project For the Controller

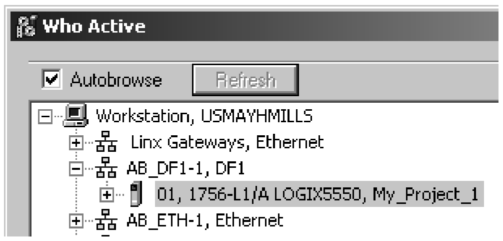
1. Open the RSLogix 5000 project for the controller.



2. Click to define a path to the controller.


Path: AB_ETH-1\192.168.1.200\Backplane\0

3. Select the controller.

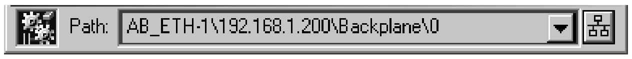


4. Click Go Online.

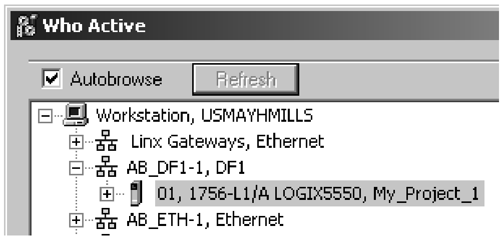
If Your Computer Does Not Have the Project For the Controller

1. Open the RSLogix 5000 project for the controller.

2. Click to define a path to the controller.


Path: AB_ETH-1\192.168.1.200\Backplane\0

3. Select the controller.



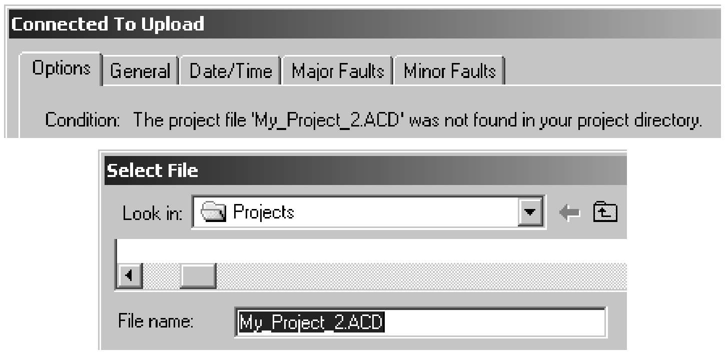
Who Active

☑ Autobrowse Refresh

Workstation, USMAYHMILLS
Linx Gateways, Ethernet
AB_DF1-1, DF1
01, 1756-L1/A LOGIX5550, My_Project_1
AB_ETH-1, Ethernet

4. Click .

5. Click to create the project file on your computer.



Connected To Upload

Condition: The project file 'My_Project_2.ACD' was not found in your project directory.

Select File

Look in: Projects

File name: My_Project_2.ACD

6. Click and then .

Program a Project Online

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

What You Need

You need these items to complete the tasks in this manual:
- Personal computer running RSLogix 5000 software, version 16 or later, and RSLinx software
- The physical system to which you are connecting
- The project you want to access

Follow These Steps

1. Edit Logic While Online (page 119)
2. Finalize All Edits in a Program (page 124)

Edit Logic While Online

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

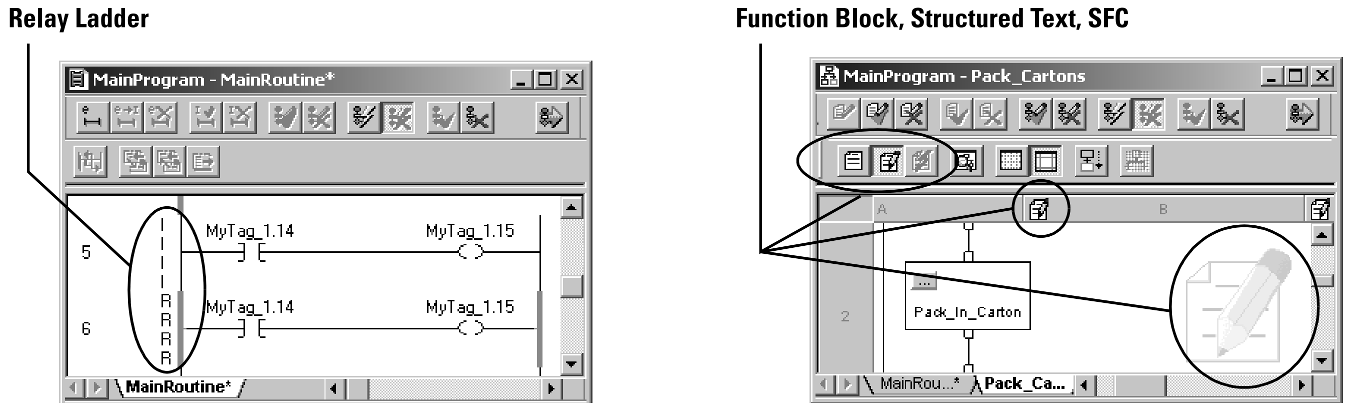
ATTENTION

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.

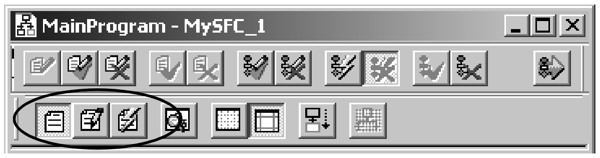
IMPORTANT

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
MainProgram - MainRoutine*
MyTag_1.14
MyTag_1.15
MainRoutine*
MainProgram - Pack_Cartons
Pack_In_Carton
MainRou...*
Pack_Ca...

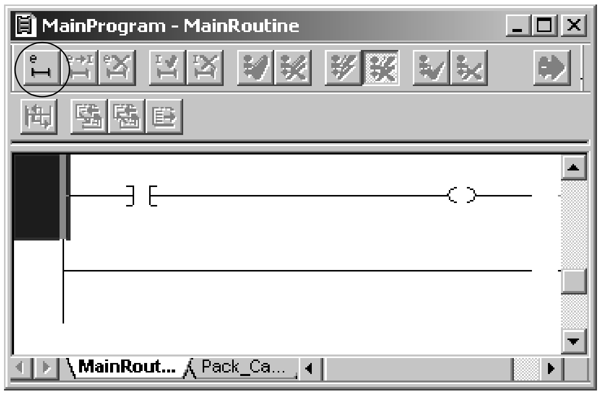
This marker		Means	Description
Relay ladder Function block Structured text SFC	r r - or - R R ≣	Original logic	When online, RSLogix 5000 software continues to show you the original logic while you edit a copy of the logic (pending edit). A green border or side rail shows which logic the controller is currently running. In function block, structured text, or SFC, use the buttons above the routine to switch between different views.  MainProgram - MySFC_1
Relay ladder Function block Structured text SFC	i i - or - e e ✎	Pending edits	This is a copy of the original logic for you to edit. Any changes remain on your computer until you accept the edits. • In relay ladder, you edit individual rungs within a routine. • In function block, structured text, or SFC, you edit an entire routine.

This marker		Means	Description
Relay ladder Function block Structured text SFC	I I - or - D D	Test edits	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). **If you** / **Then** Test the edits: • 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: • 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. In relay ladder, if you delete a rung the software immediately marks it as a test edit (upper-case 'D' character).

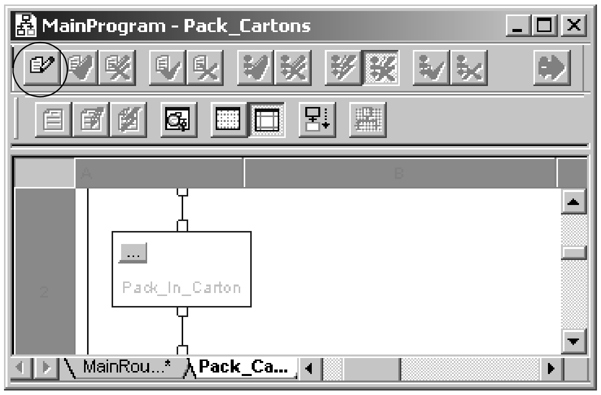
Start a Pending Edit

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

2. Start a pending edit.



Relay Ladder



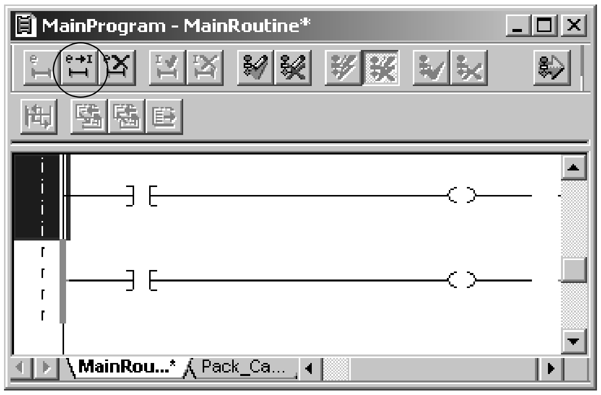
Function Block, Structured Text, SFC

Make and Accept Your Edits

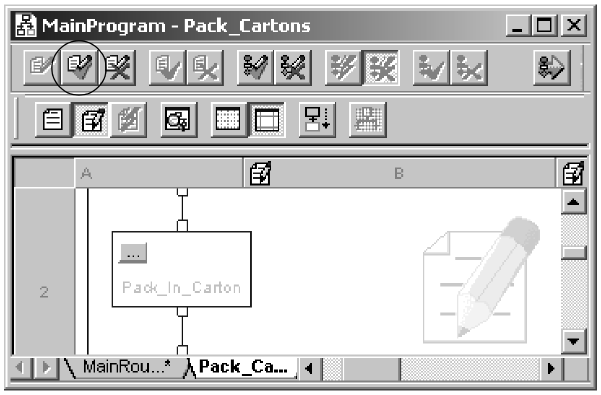
1. Make your changes.

2. Accept your changes.

   The changes download to the controller and become test edits.



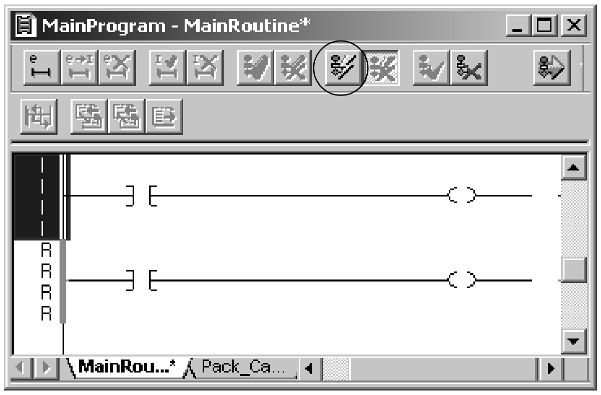
Relay Ladder



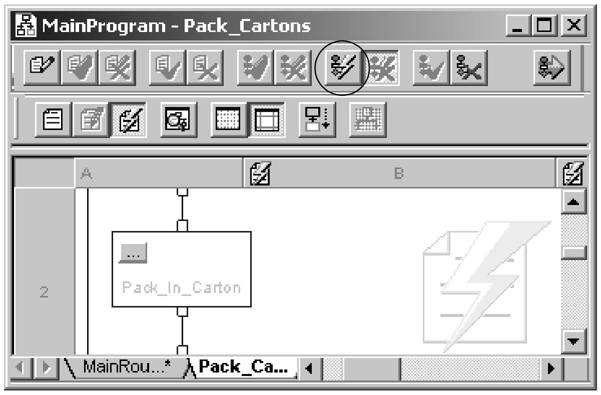
Function Block, Structured Text, SFC

Test the Edits

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

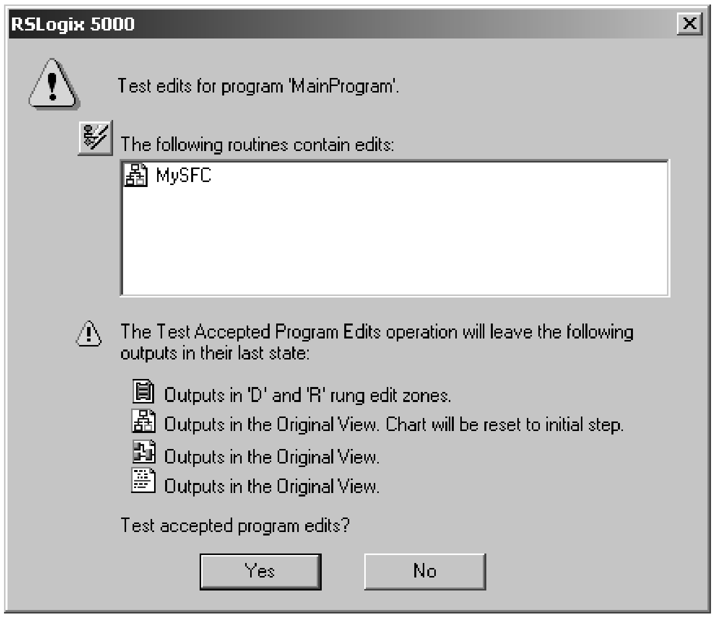


Relay Ladder



Function Block, Structured Text, SFC

2. Click Yes to test the edits.



RSLogix 5000

Test edits for program 'MainProgram'.

The following routines contain edits:

MySFC

The Test Accepted Program Edits operation will leave the following outputs in their last state:

Outputs in 'D' and 'R' rung edit zones.

Outputs in the Original View. Chart will be reset to initial step.

Outputs in the Original View.

Outputs in the Original View.

Test accepted program edits?

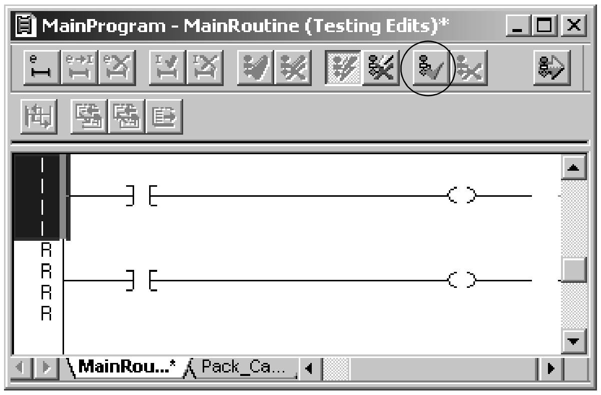
3. If the edits are not correct, click to switch execution back to your original logic (untest the edits).

   To make changes, start another pending edit.

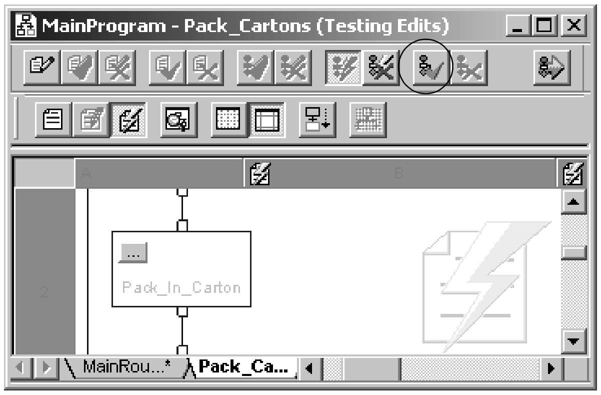
Assemble and Save the Edits

1. Assemble the edits.

   The edits become permanent and the original logic is removed.



Relay Ladder

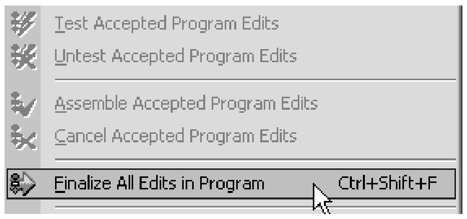


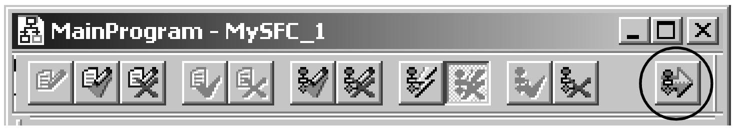
Function Block, Structured Text, SFC

2. Save the project.

Finalize All Edits in a Program

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


Test Accepted Program Edits
Untest Accepted Program Edits
Assemble Accepted Program Edits
Cancel Accepted Program Edits
Finalize All Edits in Program Ctrl+Shift+F



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.

Follow these steps 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.

IMPORTANT

When editing online, if the program scan time is large, or the number of modified rungs is large, you might see HMI and RSLogix 5000 communication timeouts when edits are finalized.

The timeout is caused by the (scan time) x (number of changed rungs). You could have a large program with a very fast scan, or a lot of rungs (but you only modified a few), and you will not see a timeout.

Troubleshoot the Controller

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

What You Need

You need these items to complete the tasks in this manual:
- Personal computer running RSLogix 5000 software, version 16 and RSLinx software
- The physical system you are troubleshooting
- The project you want to troubleshoot

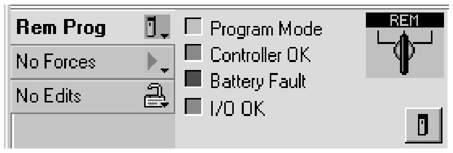
Follow These Steps

4. Troubleshoot I/O communication (page 126).
5. Clear a major fault (page 127).
6. Search a project (page 128).
7. Browse logic (page 130).
8. Force an I/O value (page 131).
9. Create and run a trend (histogram) (page 135).
10. View scan time (page 138).

Troubleshoot I/O Communication

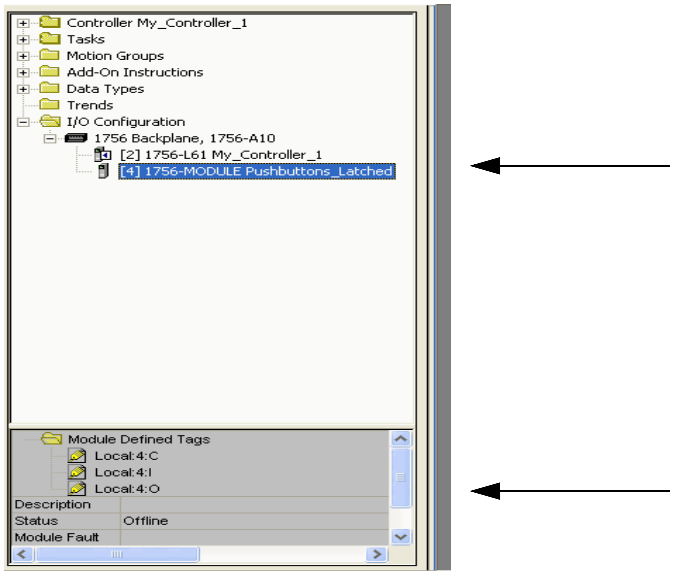
If there is a problem with several of the devices in your system, communication with an I/O module may have failed.

The I/O indicator on the front of the controller and in the programming software indicates status.



If the indicator is	Then
Off	Either: • There are no modules in the I/O configuration of the controller. • The controller does not 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 not responding.

The Controller Organizer also shows status.



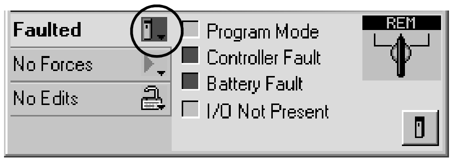
Description	
Status	Offline
Module Fault	

Indicator	**Description**
⚠	Shows that the controller is not communicating with the module.
Module fault	Communication with a module has failed.
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.

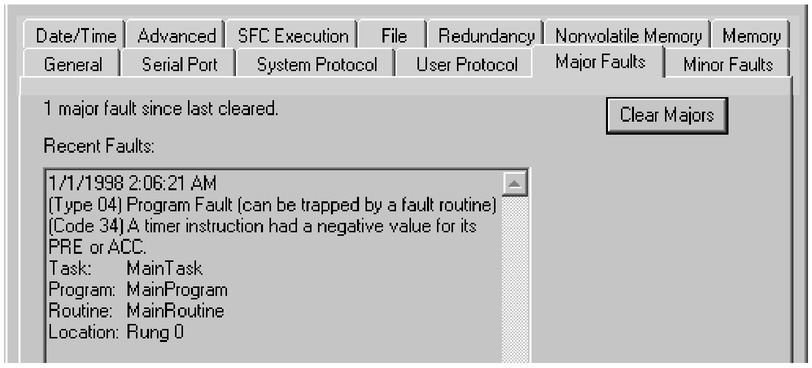
Clear a Major Fault

If your entire process unexpectedly shuts down, the controller may have experienced a major fault. A major fault is a condition severe enough for the controller to shut down.

1. Go online with the controller.
2. Choose Go To Faults.



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



1 major fault since last cleared.

Recent Faults:

```
1/1/1998 2:06:21 AM
(Type 04) Program Fault (can be trapped by a fault routine)
(Code 34) A timer instruction had a negative value for its
PRE or ACC.
Task:     MainTask
Program:  MainProgram
Routine:  MainRoutine
Location: Rung 0
```

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

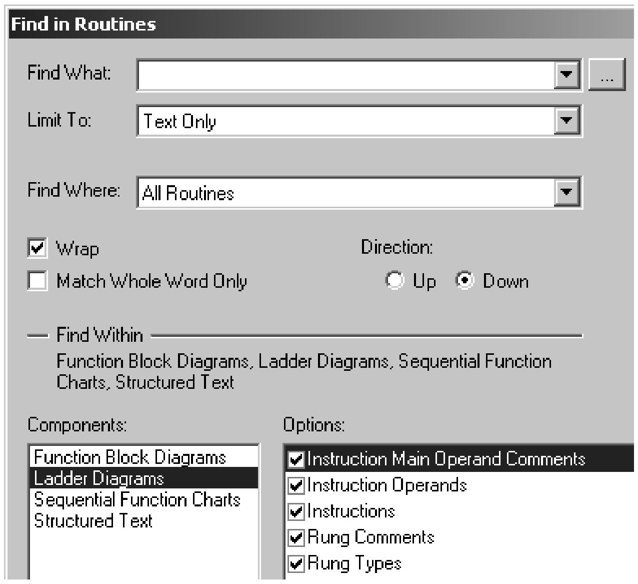
Search Functions in a Project

You can find an element of your logic (such as a tag, instruction, or comment) based on the characters that you search for.

To find a(n)	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 Element

1. Open the RSLogix 5000 project that you want to search.
2. From the Search menu, choose Find.


Find in Routines

Find What:

Limit To: Text Only

Find Where: All Routines

[✓] Wrap

[ ] Match Whole Word Only

Direction: Up [● Down]

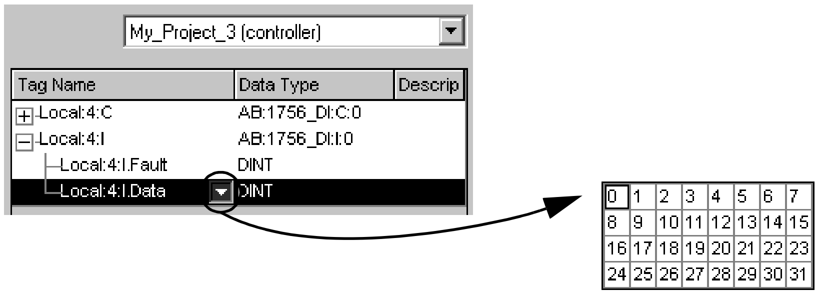
Find Within
Function Block Diagrams, Ladder Diagrams, Sequential Function Charts, Structured Text

Components:
Function Block Diagrams
Ladder Diagrams
Sequential Function Charts
Structured Text

Options:
[✓] Instruction Main Operand Comments
[✓] Instruction Operands
[✓] Instructions
[✓] Rung Comments
[✓] Rung Types

3. Specify the search criteria.
   a. Type the characters to find.

   To browse for a tag, click ..., select the tag, and click OK.
   To select a bit number, click the ▼.



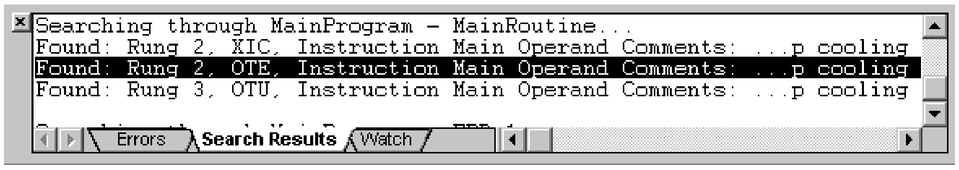
   b. Choose Text Only.
   c. Choose All Routines.
   d. Select each language and check the options in which to search.

   To display this section of the dialog box, click Find Within >>.

4. Click Find All.

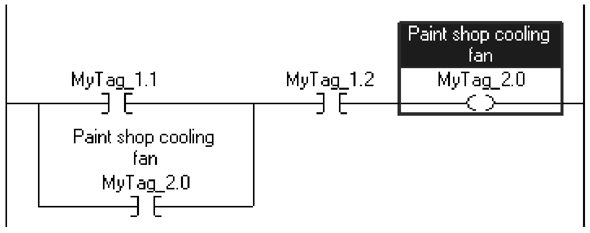
Go to an Instruction

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

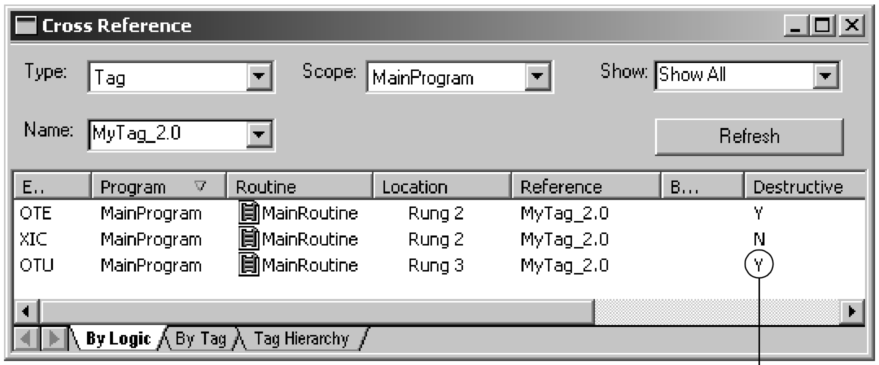


```
Searching through MainProgram - MainRoutine...
Found: Rung 2, XIC, Instruction Main Operand Comments: ...p cooling
Found: Rung 2, OTE, Instruction Main Operand Comments: ...p cooling
Found: Rung 3, OTU, Instruction Main Operand Comments: ...p cooling
```

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.



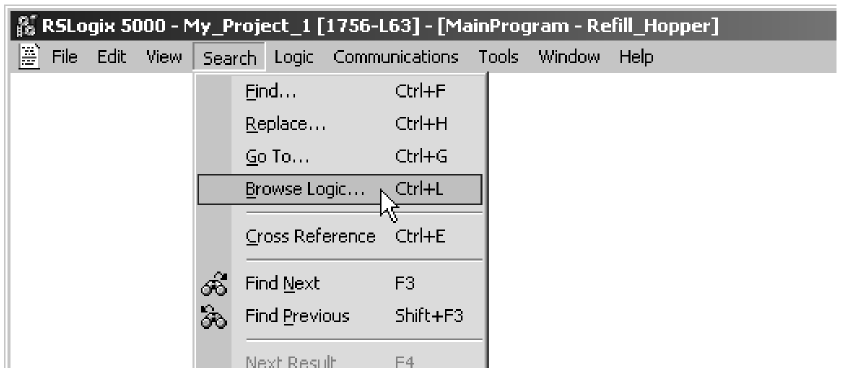
E..	Program	Routine	Location	Reference	B...	Destructive
OTE	MainProgram	MainRoutine	Rung 2	MyTag_2.0		Y
XIC	MainProgram	MainRoutine	Rung 2	MyTag_2.0		N
OTU	MainProgram	MainRoutine	Rung 3	MyTag_2.0		Y

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

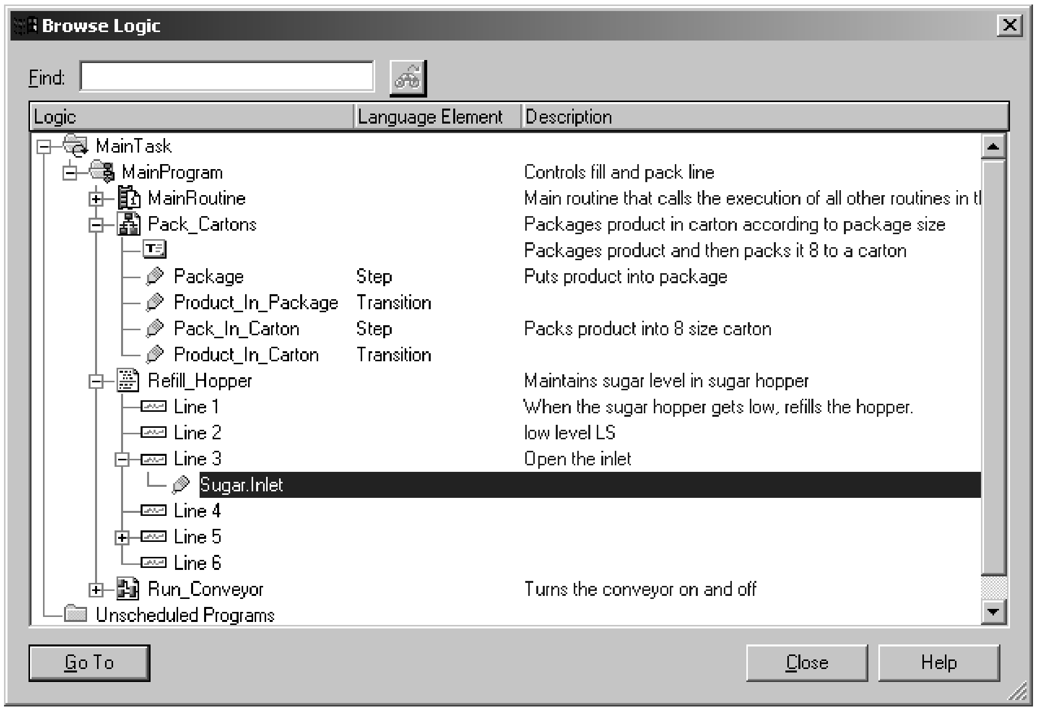
Browse Logic

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

1. In RSLogix 5000 software, from the Search menu, choose Browse Logic.



2. Expand and collapse entries to see or hide its contents.



Logic	Language Element	Description
MainTask		
MainProgram		Controls fill and pack line
MainRoutine		Main routine that calls the execution of all other routines in tl
Pack_Cartons		Packages product in carton according to package size
		Packages product and then packs it 8 to a carton
Package	Step	Puts product into package
Product_In_Package	Transition	
Pack_In_Carton	Step	Packs product into 8 size carton
Product_In_Carton	Transition	
Refill_Hopper		Maintains sugar level in sugar hopper
Line 1		When the sugar hopper gets low, refills the hopper.
Line 2		low level LS
Line 3		Open the inlet
Sugar.Inlet		
Line 4		
Line 5		
Line 6		
Run_Conveyor		Turns the conveyor on and off
Unscheduled Programs		

3. To go to the location of a element in logic, select the element and click Go To.

Forcing 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.

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.

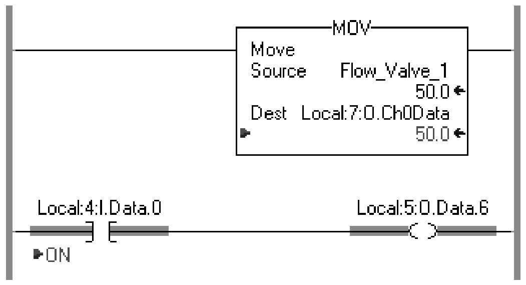
If you remove an individual force, forces remain in the enabled state.

If forces are enabled and you install a force, the new force immediately takes effect.

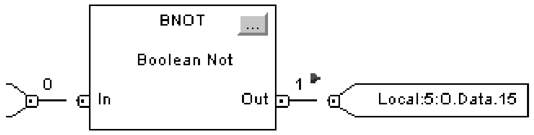
If you want to	Then
Override a value	Install an I/O force (force an I/O value)
Stop an individual force but leave other forces enabled and in effect	Remove an individual force
Stop all I/O forces but leave the I/O forces in the project	Disable all I/O forces

A force overrides a value from an input device or logic.

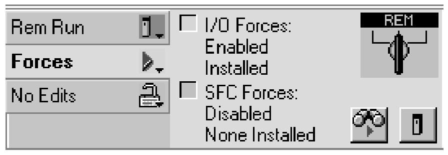
- Forcing an input tag overrides the value from the input device.
- Forcing an output tag overrides your logic and sends the force value to the output device.



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



The force indicator on the front of the controller and in the programming software indicates status.

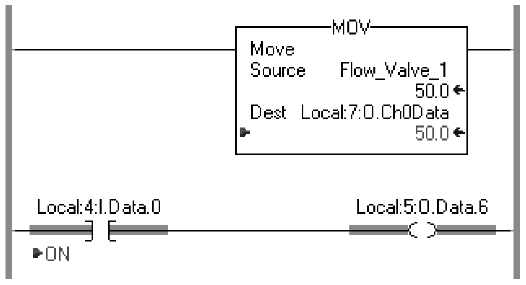


If the indicator is	Then
Off	• No tags contain I/O force values. • I/O forces are inactive (disabled).
Flashing amber	• One or more tags contain a force value. • I/O forces are inactive (disabled). • When you enable I/O forces, all existing I/O forces take effect.
Solid amber	• I/O forces are active (enabled). • Force values may or may not exist. • When you install (add) a force, it immediately takes effect.

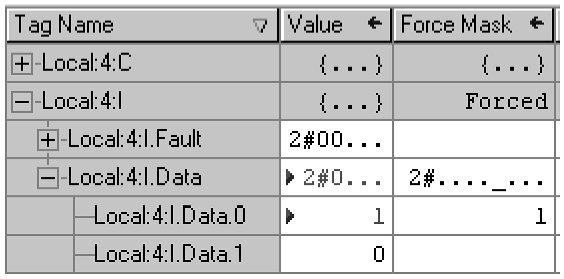
Install an I/O Force (Force an I/O Value

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 (for example, the BOOL value of a DINT tag).

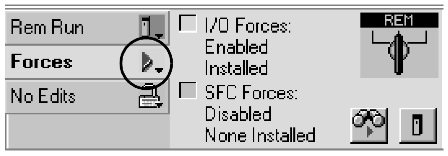


Tag Name	Value	Force Mask
+-Local:4:C	{...}	{...}
−-Local:4:I	{...}	Forced
+-Local:4:I.Fault	2#00...	
−-Local:4:I.Data	2#0...	2#...._...
Local:4:I.Data.0	1	1
Local:4:I.Data.1	0	

4. Install the force value:

To force a	Do this
BOOL value	Right-click the tag and choose Force ON or Force OFF.
Integer or REAL value	In the Force Mask column for the tag, type the value to which you want to force the tag and press [Enter}.

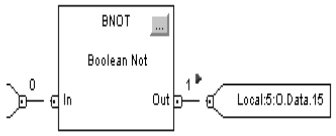
5. From the Forces menu, choose I/O Forcing > Enable All I/O Forces and click Yes.



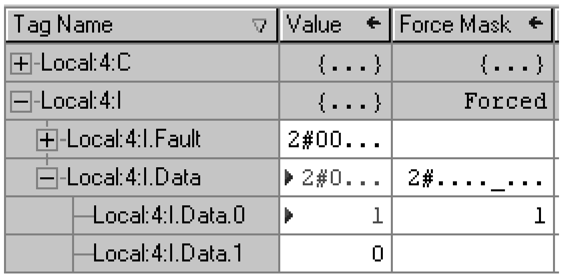
Remove an Individual Force

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 (for example, the BOOL value of a DINT tag).



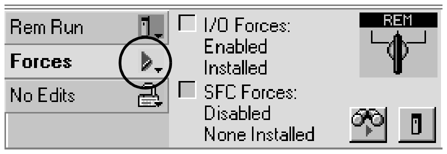
Tag Name	Value	Force Mask
+-Local:4:C	{...}	{...}
−-Local:4:I	{...}	Forced
+-Local:4:I.Fault	2#00...	
−-Local:4:I.Data	2#0...	2#...._...
Local:4:I.Data.0	1	1
Local:4:I.Data.1	0	

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

Disable All I/O Forces

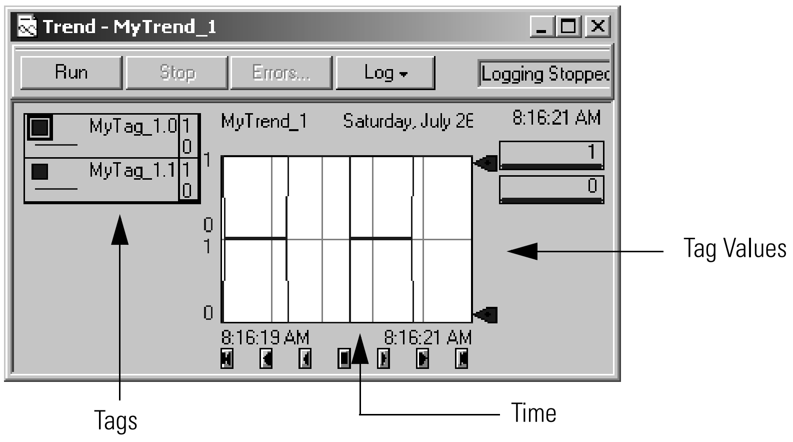
1. Go online with the controller.

2. From the Forces menu, choose I/O Forcing > Disable All I/O Forces and click Yes.



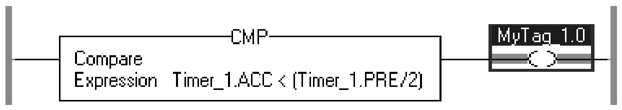
Data Trend (Histogram)

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.



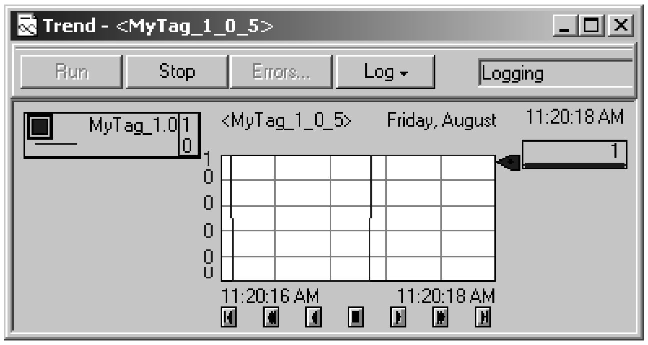
Run a Trend for a Tag

Right-click the first tag that you want to trend and choose Trend.

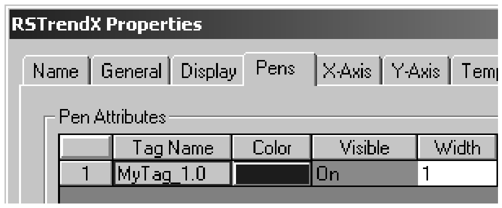


Add More Tags to the Trend

1. Right-click the chart and choose Chart Properties.



2. Click the Pens tab.



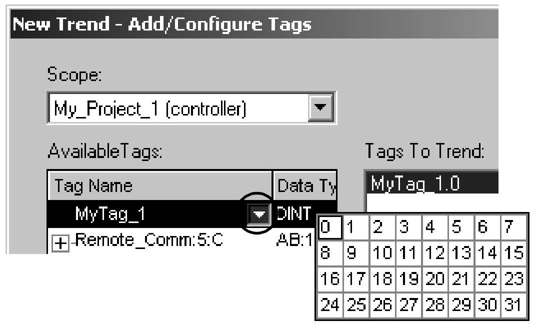
	Tag Name	Color	Visible	Width
1	MyTag_1.0	■	On	1

3. Click Add/Configure Tags.

4. Select a tag to add and click Add -->.

   To change the scope, select a scope.

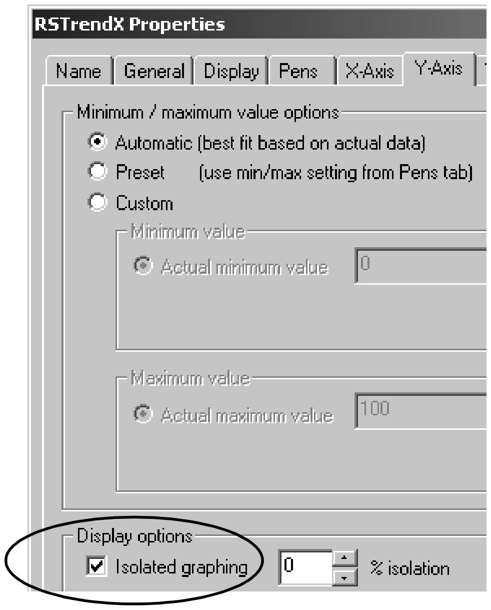
5. To select a bit number, click ▼.



6. When you have added the required tags, click OK.

7. Click the Y-Axis tab.

8. Choose the type of graphing and click OK.



RSTrendX Properties

Minimum / maximum value options

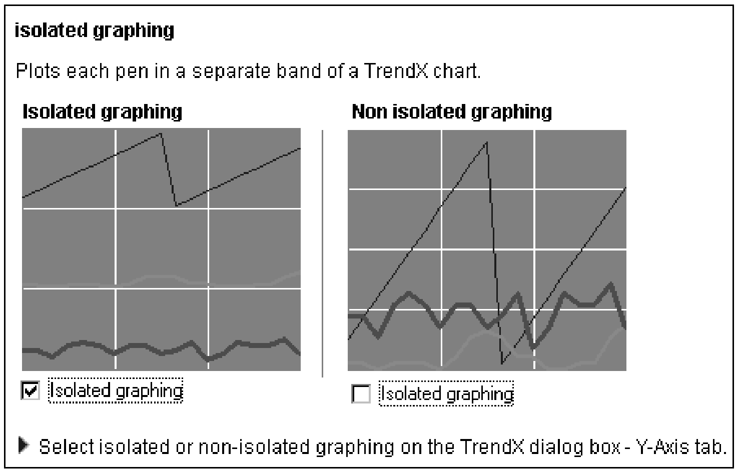
- (●) Automatic (best fit based on actual data)
- ( ) Preset (use min/max setting from Pens tab)
- ( ) Custom

Minimum value: Actual minimum value 0

Maximum value: Actual maximum value 100

Display options

☑ Isolated graphing 0 % isolation



isolated graphing

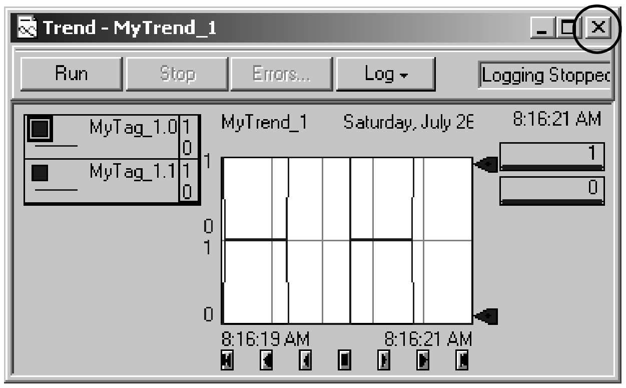
Plots each pen in a separate band of a TrendX chart.

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

9. To resume the trend, click Run .

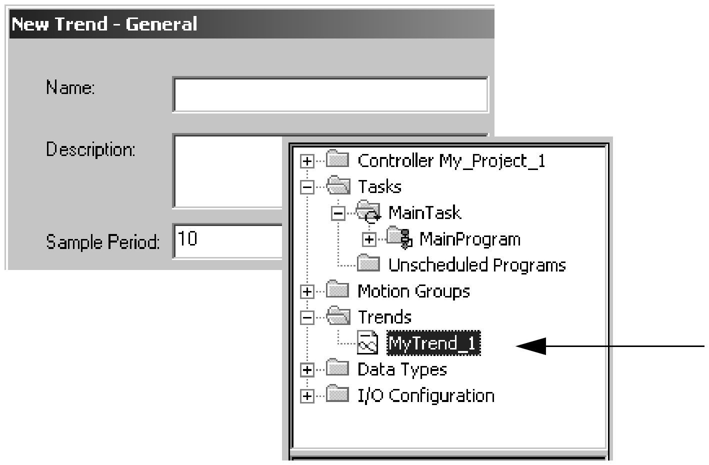
Save the Trend

1. Close the trend.



You get the choice to save the trend for future use.

2. Type a name for the trend and click Finish.



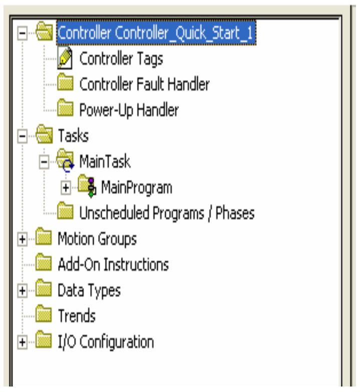
View Scan Time

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

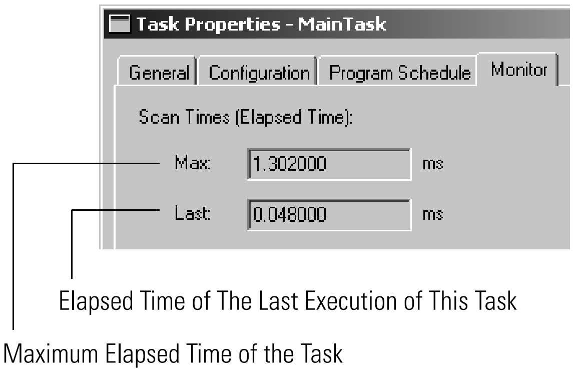
Scan Time	Description
Elapsed time (task scan time)	Time that has elapsed from the start of a task to the end of the task, in milliseconds. The elapsed time of a task includes the time that the task is interrupted to service communications or other tasks.
Execution time (program scan time)	Time to execute 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.



Scan Times (Elapsed Time):

Max: 1.302000 ms

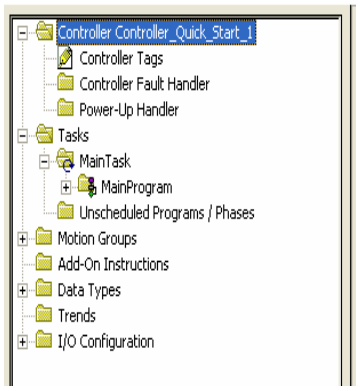
Last: 0.048000 ms

Elapsed Time of The Last Execution of This Task

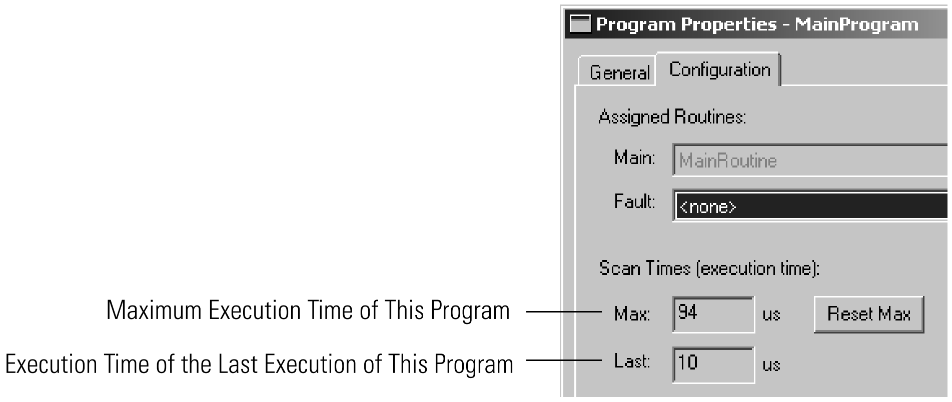
Maximum Elapsed Time of the Task

View Program Scan Time

1. Right-click and choose Properties.



2. Click the Configuration tab.



Assigned Routines:

Main: MainRoutine

Fault: <none>

Scan Times (execution time):

Max: 94 us

Last: 10 us

Maximum Execution Time of This Program

Execution Time of the Last Execution of This Program

Notes:

Notes:

Notes:

Rockwell Automation Support

Rockwell Automation provides technical information on the Web to assist you in using its products. At http://www.rockwellautomation.com/support/, you can find technical manuals, a knowledge base of FAQs, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools.

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

Installation Assistance

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

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

New Product Satisfaction Return

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

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

Documentation Feedback

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

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication 1756-QS001E-EN-P - October 2009
Supersedes Publication 1756-QS001D-EN-P - February 2007