Allen-Bradley

CompactLogix™ System

(Catalog Numbers 1769-L20, 1769-L30)

User Manual

Rockwell Automation

Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards.

The illustrations, charts, sample programs and layout examples shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Allen-Bradley does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the examples shown in this publication.

Allen-Bradley publication SGI-1.1, *Safety Guidelines for the Application, Installation and Maintenance of Solid-State Control* (available from your local Allen-Bradley office), describes some important differences between solid-state equipment and electromechanical devices that should be taken into consideration when applying products such as those described in this publication.

Throughout this manual we use notes to make you aware of safety considerations:

ATTENTION Or **WARNING**	Identifies information about practices or circumstances that can lead to personal injury or death, property damage or economic loss

Attention and warning statements help you to:

- identify a hazard
- avoid a hazard
- recognize the consequences

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

Introduction

This version of the CompactLogix System User Manual corresponds to version 11 of the controller. Revision bars (shown in the left margin of this page) indicate changed information. Changes made to this manual include:

For this information:	See:
You can now change the RPI for local I/O to be any rate from 2ms to 750ms	page 3-7
The 1769 Generic Profile information (previously in an appendix) was moved to the end of the Placing, Configuring, and Monitoring Local I/O chapter to make the information more accessible	chapter 3
Additional fault word information.	page 3-15
The messaging over DH-485 example (previously in an appendix) was moved to the Communicating with Devices on DH-485 chapter to make the information more accessible.	chapter 5
How to use a 1769-SDN scanner in a DeviceNet configuration.	chapter 6
The 1761-NET-DNI information (previously in an appendix) was moved to the Communicating with Devices on DeviceNet chapter to make the information more accessible.	chapter 6
You no longer need a 1761-NET-ENI module connected to your computer to be able to communicate with a CompactLogix controller over Ethernet. If you have series B ENI modules, the computer can use its own Ethernet card to communicate with the CompactLogix controller.	chapter 7
The 1761-NET-ENI information (previously in an appendix) was moved to the Communicating with Devices on EtherNet chapter to make the information more accessible.	chapter 7
Execution times and memory use are no longer documented in this manual. This information for all Logix controllers is now available in one publication.	*Logix5000 Controllers Execution Time and Memory Use Reference Manual*, publication 1756-RM087

Notes:

Who Should Use This Manual

Read this preface to familiarize yourself with the rest of the manual. This preface covers the following topics:

- who should use this manual
- how to use this manual
- related publications
- conventions used in this manual
- Rockwell Automation support

Use this manual if you are responsible for designing, installing, programming, or troubleshooting control systems that use Allen-Bradley CompactLogix™ controllers.

How to Use This Manual

As much as possible, we organized this manual to explain, in a task-by-task manner, how to install, configure, program, operate and troubleshoot a CompactLogix control system.

Related Documentation

The table below provides a listing of publications that contain important information about CompactLogix systems.

For	Read this document	Document number
Information on installing a CompactLogix controller	CompactLogix Modular Processors Installation Instructions	1769-IN047
Information on common procedures using RSLogix 5000 software.	Logix5000 Controllers Common Procedures Programming Manual	1756-PM001
Information on the CompactLogix Instruction Set	Logix5000 Controllers General Instruction Set Reference Manual	1756-RM003
Information on function block programming Logix controllers.	Logix5000 Controllers Process Control/Drives Instruction Set Reference Manual	1756-RM006
Exeuction times and memory use for instructions	Logix5000 Controllers Execution Time and Memory Use Reference Manual	1756-RM087
Information on installing, configuring, and using Compact Analog I/O modules	Compact I/O Analog Modules User Manual	1769-UM002
Information on using the 1769-ADN DeviceNet adapter	Compact I/O 1769-ADN DeviceNet Adapter User Manual	1769-UM001
Information on using the 1769-SDN DeviceNet scanner	Compact I/O 1769-SDN DeviceNet Scanner Module User Manual	1769-UM009
Information on grounding and wiring Allen-Bradley programmable controllers.	Allen-Bradley Programmable Controller Grounding and Wiring Guidelines	1770-4.1

If you would like a manual, you can:

- download a free electronic version from the internet at **www.theautomationbookstore.com**
- purchase a printed manual by:
  - contacting your local distributor or Rockwell Automation representative
  - visiting **www.theautomationbookstore.com** and placing your order
  - calling 1.800.963.9548 (USA/Canada) or 001.330.725.1574 (Outside USA/Canada)

Conventions Used in This Manual

The following conventions are used throughout this manual:

- Bulleted lists (like this one) provide information not procedural steps.
- Numbered lists provide sequential steps or hierarchical information.
- *Italic* type is used for emphasis.

Rockwell Automation Support

Rockwell Automation offers support services worldwide, with over 75 Sales/Support Offices, 512 authorized distributors and 260 authorized Systems Integrators located throughout the United States alone, plus Rockwell Automation representatives in every major country in the world.

Local Product Support

Contact your local Rockwell Automation representative for:

- sales and order support
- product technical training
- warranty support
- support service agreement

Technical Product Assistance

If you need to contact Rockwell Automation for technical assistance, please review the information in Appendix B, CompactLogix Troubleshooting, first. Then call your local Rockwell Automation representative.

Your Questions or Comments on the Manual

If you find a problem with this manual, please notify us. If you have any suggestions for how this manual could be made more useful to you, please contact us at the address below:

Rockwell Automation
Automation Control and Information Group
Technical Communication, Dept. A602V
P.O. Box 2086
Milwaukee, WI 53201-2086

Notes:

Chapter 1

Getting Started

Introduction 1-1
Creating and Downloading a Project 1-2
Creating a project. 1-3
Changing project properties 1-4
Configuring the CompactBus 1-5
Adding a local I/O module. 1-7
Changing module properties 1-9
Viewing I/O tags 1-10
Creating other tags. 1-11
Documenting I/O with alias tags 1-12
Entering logic 1-13
Downloading a project. 1-15
Viewing program scan time 1-16
Viewing controller memory usage 1-17
What To Do Next 1-17

Chapter 2

What Is CompactLogix?

Using This Chapter 2-1
Developing Programs 2-2
Defining Tasks. 2-3
Defining Programs 2-5
Defining Routines 2-5
Selecting a System Overhead Percentage 2-6

Chapter 3

Placing, Configuring, and Monitoring Local I/O

Using This Chapter 3-1
Placing Local I/O Modules. 3-1
System Power Budget Calculation and Considerations . . 3-3
Verifying I/O Layout by Adding Total Words of Backplane Memory Used 3-4
Determining When the Controller Updates I/O. 3-5
Configuring the CompactBus 3-5
Configuring Local I/O Modules 3-7
Communication Formats. 3-8
Inhibiting I/O Module Operation 3-9
Configuring Response to a Connection Failure 3-11
Accessing I/O Data 3-11
Using Aliases to Simplify Tag Names 3-13
Direct Connections for I/O Modules 3-13
Monitoring I/O Modules 3-14
Displaying Fault Data. 3-14
End-Cap Detection and Module Faults 3-16
Configuring I/O Modules Using the Generic Profile 3-17

Chapter 4

Communicating with Devices on a Serial Link

Using This Chapter . . . 4-1
Default Communication Configuration . . . 4-1
System Protocol Options . . . 4-2
Using the Ch 0 Default Communication Push Button . . . 4-2
Configuring Your System for a Serial Link . . . 4-3
Step 1: Configure the Hardware . . . 4-4
Step 2: Configure the Serial Port(s) of the Controller . . . 4-7
Step 3: Configure the Serial Communication Driver . . . 4-10
Example 1: Workstation Directly Connected to a Controller 4-11
Configuring a DF1 Point-to-Point Station. . . . 4-11
Example 2: Workstation Remotely Connected. . . . 4-12
Master/Slave Communication Methods . . . 4-12
Configuring a DF1 Slave Station . . . 4-14
Configuring a DF1 Master Station . . . 4-14
Example 3: Controller Connected to a Bar Code Reader . . . 4-16
Connect the ASCII Device to the Controller. . . . 4-16
Configuring User Mode . . . 4-18
Programming ASCII Instructions . . . 4-18

Chapter 5

Communicating with Devices on a DH-485 Link

Using This Chapter . . . 5-1
Configuring Your System for a DH-485 Link. . . . 5-2
Step 1: Configure the Hardware . . . 5-2
Step 2: Configure the DH-485 Port of the Controller . . . 5-4
Planning a DH-485 Network . . . 5-5
DH-485 Token Rotation . . . 5-5
Network Initialization. . . . 5-6
Number of Nodes and Node Addresses. . . . 5-6
Installing a DH-485 Network . . . 5-7
Grounding and Terminating a DH-485 Network . . . 5-9
Example: Messaging on DH-485 . . . 5-9
Configuring and Programming the 1769-L20 Controller . 5-10
Configuring and Programming the 1769-L30 Controller . 5-15

Chapter 6
Communicating with Devices on a DeviceNet link

Using This Chapter 6-1
Configuring Your System for a DeviceNet Link 6-1
Example 1: Controlling DeviceNet Devices 6-2
Step 1: Configuring the 1769-ADN Adapter 6-3
Step 2: Setting Up the 1769-SDN Scanlist 6-5
Step 3: Creating a Project 6-10
Step 4: Enter Program Logic 6-12
Example 2: Sending Messages Over DeviceNet Using a 1761-NET-DNI Interface Converter 6-13
Step 1: Configure the Hardware 6-13
Step 2: Commissioning the 1761-NET-DNI Module 6-13
Step 3: Connecting the Controllers to DeviceNet 6-15
Step 4: Sending Messages 6-16

Chapter 7
Communications on Ethernet

Using This Chapter 7-1
Configuring Your System for an Ethernet Link 7-1
Step 1: Assigning IP Addresses 7-2
Step 2: Configuring the 1761-NET-ENI 7-3
Confirming the configuration 7-6
Step 3: Mapping Data Table Files 7-7
Configuring an Ethernet Driver in RSLinx 7-8
Example 1: Creating MSG Programs 7-9
Example 2: Staggering Multiple Messages 7-12

Appendix A
CompactLogix System Specifications

Using This Appendix A-1
CompactLogix Controller A-1
1747-BA Battery A-2
Dimensions A-3
CompactLogix Modular Controller A-3
CompactLogix System A-3
Compact I/O Expansion Power Supply and End Caps A-4

Appendix B
CompactLogix Troubleshooting

Using This Appendix B-1
CompactLogix Controller LEDs B-1
Identifying Controller Faults B-2
Calling Rockwell Automation for Assistance B-3

Appendix C

Maintaining the Battery

Using this Appendix C-1
Storing Replacement Batteries C-1
Estimating Battery Life. C-1
Replacing a Battery C-3

Appendix D

Dynamic Memory Allocation in CompactLogix Controllers

Messages D-2
RSLinx Tag Optimization D-2
Trends D-3
DDE/OPC Topics D-3
    Maximum Messaging Connections per PLC D-3
    Checking “Use Connections for Writes to Controller”. . . D-4
    Number of Connections Needed. D-4
    Viewing the Number of Open Connections. D-4

Getting Started

Introduction

This chapter introduces the CompactLogix controller and provides a quick overview on creating and downloading a project. The steps in this chapter introduce the basic aspects of the CompactLogix controller.

The CompactLogix controller offers state-of-art control and I/O modules in a small cost-effective package. CompactLogix systems feature:

- Communications over DH-485, DeviceNet, or Ethernet via peer-to-peer messaging.
- RTU functionality using dial-up modems and DF1 Full-Duplex protocol.
- RTU functionality with radio frequency or leased-line modems and DF1 Half-Duplex protocol.
- Local RS-232 connection(s) for controller project upload/download, DF1 Full-Duplex communications, DH-485 networking, or for ASCII communications.
- Remote programming over DH-485, DeviceNet, and Ethernet.

The following table lists some of the differences between the two available CompactLogix controllers.

Specification	CompactLogix5320	CompactLogix5330
Communication ports	(1) RS-232	(2) RS-232
User memory	64K bytes	256K bytes
Maximum number of I/O modules supported[1]	8 I/O modules	16 I/O modules
Maximum number of I/O banks supported	2 banks	3 banks

(1) You must use the I/O memory worksheet on page 3-4 to verify whether the layout of the I/O modules you select.

Creating and Downloading a Project

The following diagram illustrates the steps you follow to create and download a project. The remainder of this chapter provides examples of each step.

1 **Create a project** see page 1-2

2 **Configure the CompactBus** see page 1-5

3 **Configure I/O modules** see page 1-7

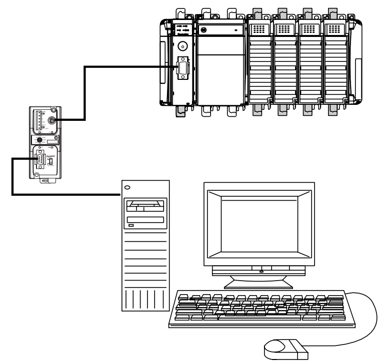
4 **Create tags** see page 1-11

5 **Enter logic** see page 1-13

6 **Download a project** see page 1-15

7 **View status** see page 1-16

System setup for this quick start:



You need:

- RSLogix5000 programming software
- RSLinx communication software
- DF1 point-to-point, serial connection from the workstation to the controller (using 1756-CP3 or 1747-CP3 cable)
- 1761-NET-AIC isolator recommended for channel 0 isolation

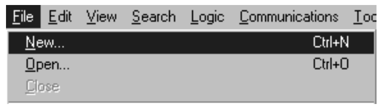
If you don't have this hardware, you can still follow these steps. Substitute the I/O modules you have for the ones listed above and make the appropriate changes.

If you use a 1769-L30 controller, you do not need to use an isolator. Channel 1 is an isolated RS-232 port.

Creating a project

1 **Create a project**

1. **Select File → New.**



2. **Define the project.** The software uses the project name you enter with an .ACD extension to store your project.

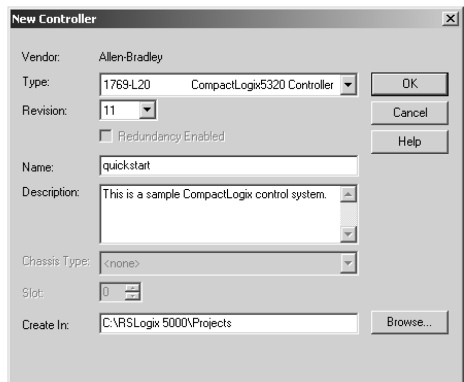
Select a controller type.

Select the controller revision.

Name the project.

Describe the project (optional).

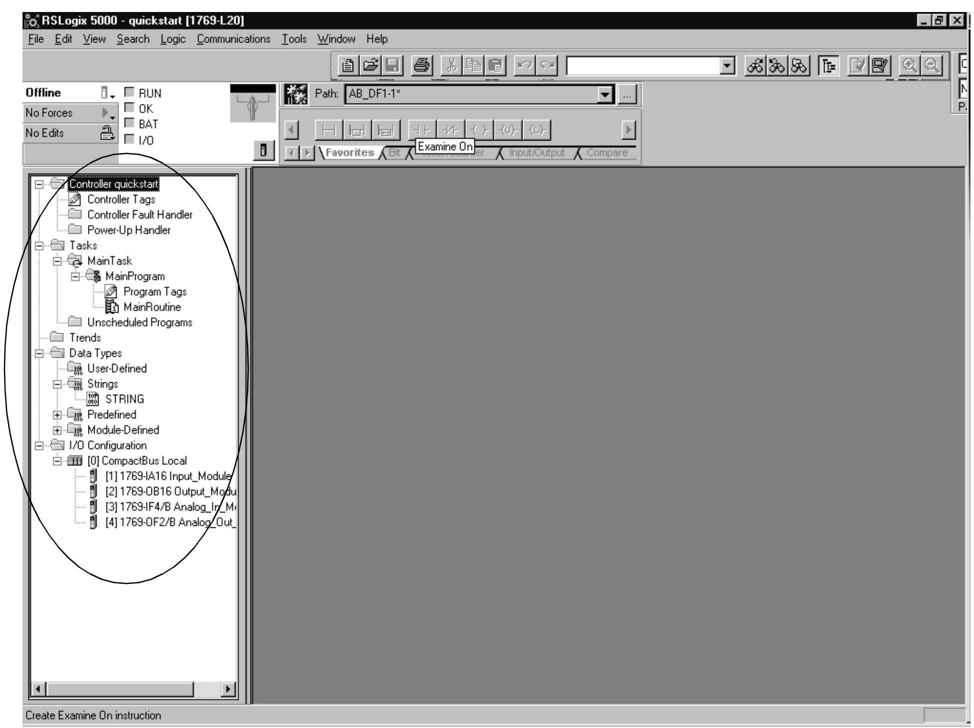
Select where to store the project (typically use the default directory).



Click OK.

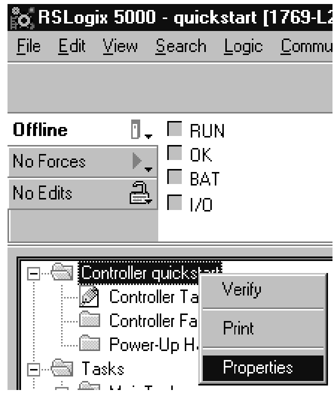
The software creates the new controller and displays:

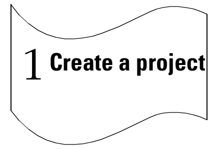
controller organizer



Changing project properties

1. View properties for Controller quick_start.



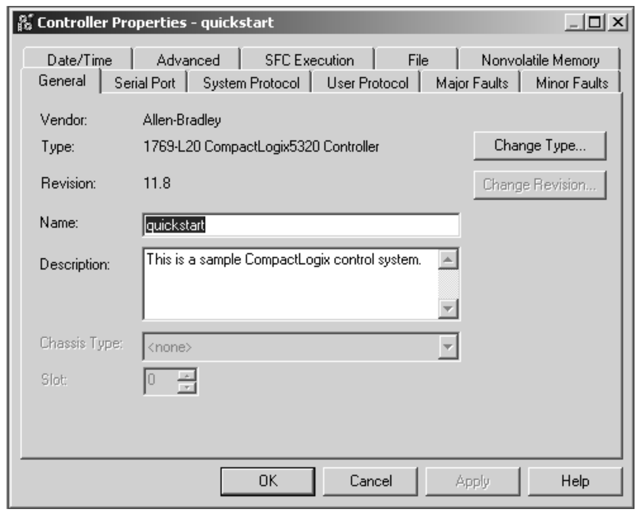


A. Place the cursor over the Controller quick_start folder.

B. Click the right mouse button and select Properties.

2. View the General tab.

The screen defaults to the General tab.



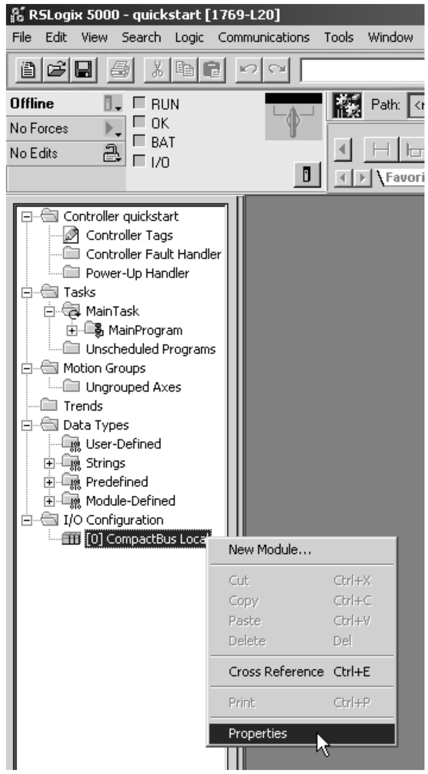
Verify that the controller settings are correct. Make changes if necessary.

Click OK.

Configuring the CompactBus

2 Configure the CompactBus

1. View properties for the CompactBus.

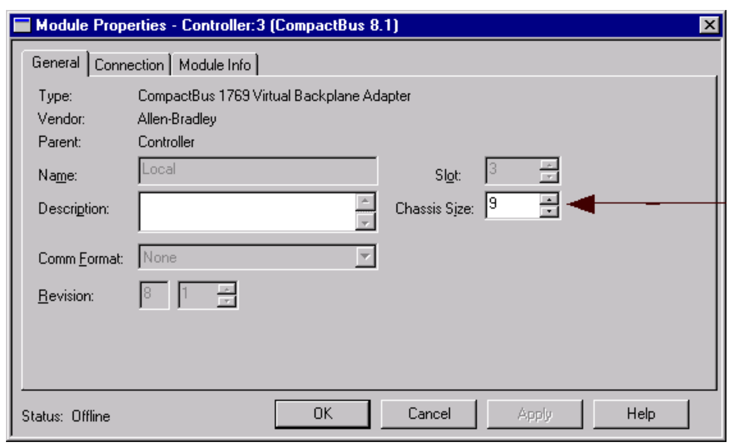


A. Place the cursor over the CompactBus.

B. Click the right mouse button and select Properties.

2. View the General tab.

The screen defaults to the General tab.



Verify that the module settings are correct. Make changes if necessary.

Click OK.

Specify the size of the chassis. Enter the number of modules you plan to install, including the controller. If the total number of modules is less than 9, there is no need to decrease this number. The system will operate the same. If you decrease this number and want to add a module later, you will need to remember to increase the chassis size at that time.

continued

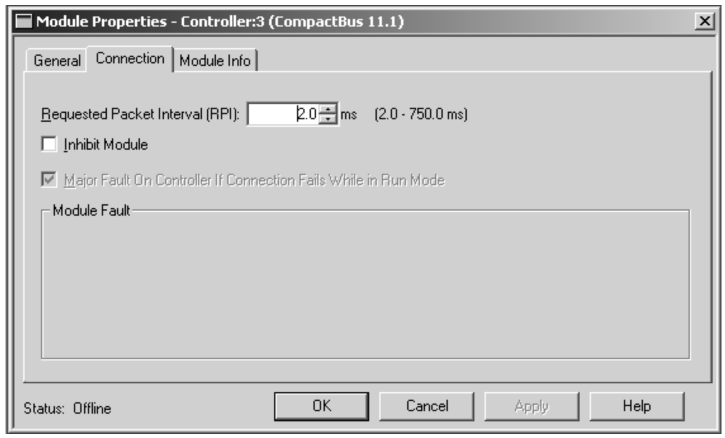
Configuring the CompactBus (*continued*)

2 **Configure the CompactBus**

3. View the Connections tab.

Specify the RPI for the CompactLogix system

Inhibit and uninhibit the CompactLogix system.



Click OK.

The RPI you specify here is the RPI for every 1769 module in this controller's system. Specify an RPI from 2-750ms for the system. You do not specify individual RPI values for each module.

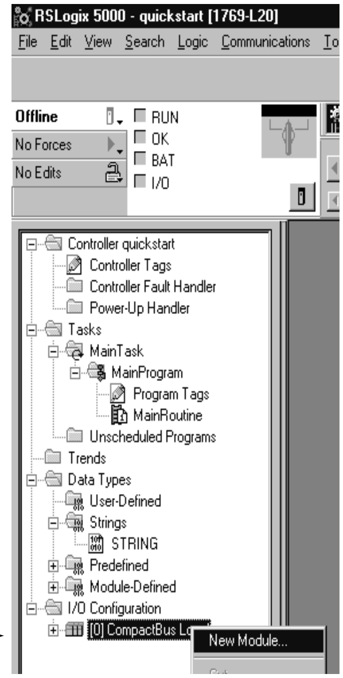
By inhibiting and uninhibiting the CompactBus, you can write new configuration data to the entire system at once.

The controller's response to a CompactBus connection failure is fixed to always fault the controller. It is not configurable.

Adding a local I/O module

3 **Configure I/O modules**

1. **Create a new module.**



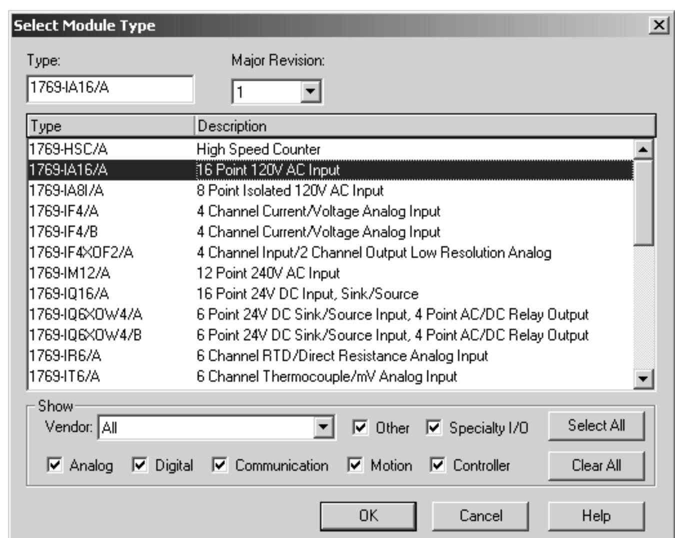
Important: If you add I/O modules other than the ones listed for this example, verify backplane memory use to make sure that the controller can support the proposed system (see page 3-4). The controller supports 256 words of backplane memory and some I/O modules consume more backplane words than others.

A. Place the cursor over [0] CompactBus Local.

B. Click the right mouse button and select New Module.

2. **Select an I/O module to add.**

Select a catalog number.
For this example, select 1769-IA16.



Click OK.

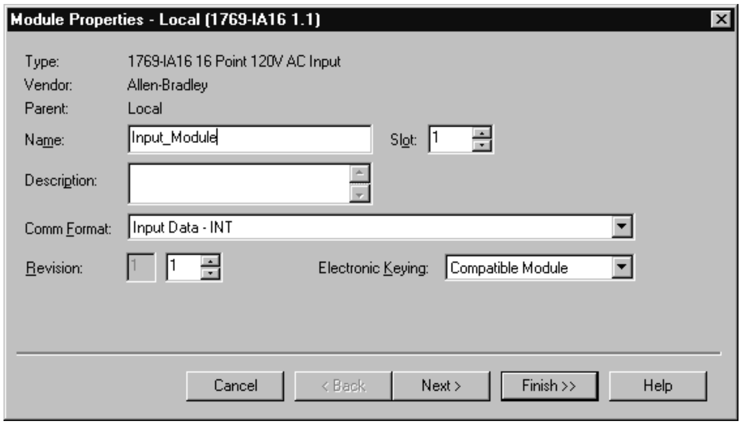
continued

Adding a local I/O module (*continued*)

3 **Configure I/O modules**

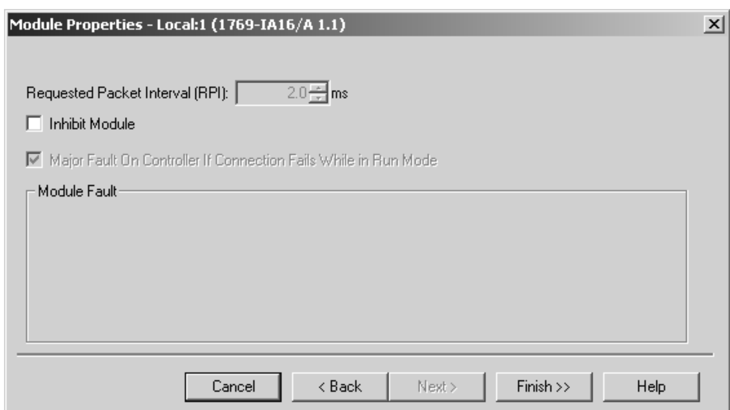
3. Identify the input module. These screens are specific to the 1769-IA16 input module.

You should enter a name.
Verify the slot number.
Describe the module (optional).
Select the communication format.
Specify electronic keying.



Click Next.

4. Use the Create wizard to configure the input module. Use default values for this example.
Different modules have different screens and parameters. If you do not want to go through each screen in the Create wizard, click Finish to create the module using default values.



Click Finish.

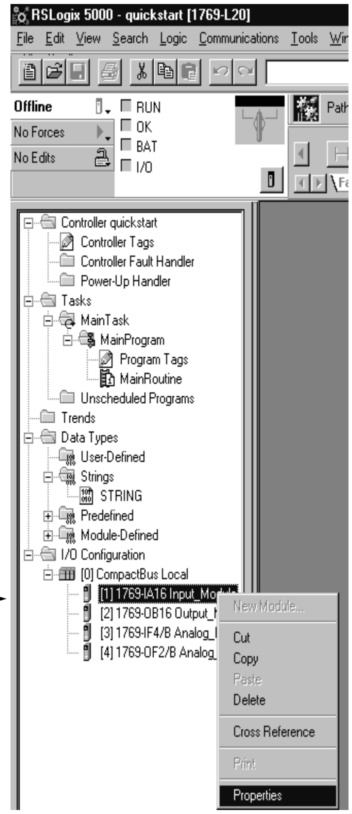
Rather than specifying an individual RPI value for each I/O module, you specify on RPI value for the whole CompactLogix system when you configure the CompactBus (see page 1-6).

Add the other I/O modules for this quick start (see page 1-2) to the Controller Organizer.

Changing module properties

3 **Configure I/O modules**

1. View properties for the module.



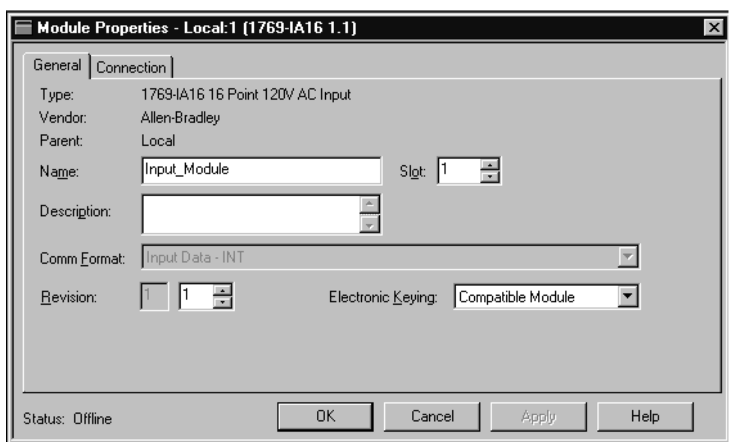
A. Place the cursor over the 1769-IA16 module.

B. Click the right mouse button and select Properties.

2. View the General tab.

The screen defaults to the General tab.

Verify that the module settings are correct. Make changes if necessary.

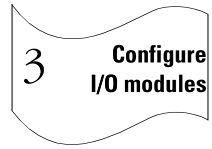


Click OK.

The tabs that appear depend on the type of module.

Important: If you want to change the communication format of a module, you must first delete the module and then re-add it using the communication format you want.

Viewing I/O tags



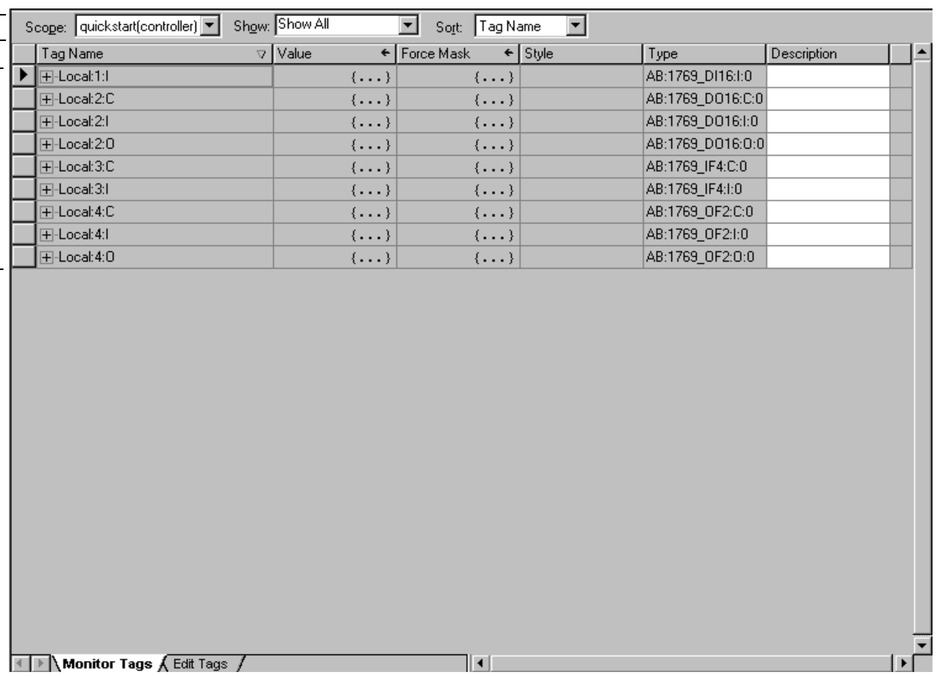
1. View the tags for the controller.

Place the cursor on the Controller Tags folder and double-click.

The software displays the module-defined tags for the I/O modules you created.

Tag Scope(1)

1769 module tags



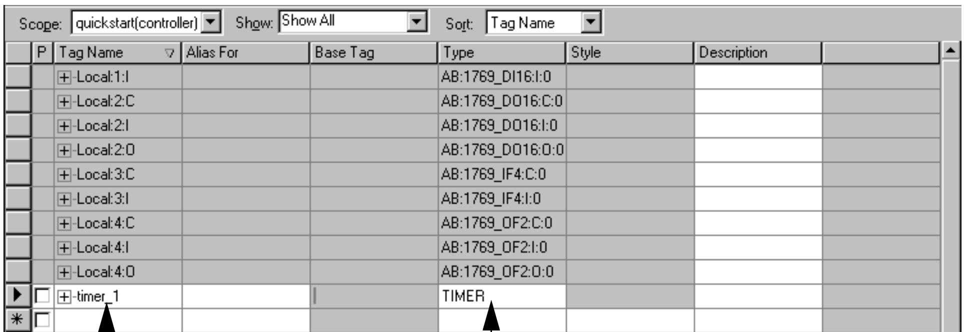
(1) Tag scope allows tag name uniqueness to be limited to a single scope (controller or program). Thus, the same tag name can be used in many programs without ambiguity.

Click the Edit Tags tab.

Creating other tags

4 **Create tags**

1. **Create a tag.**



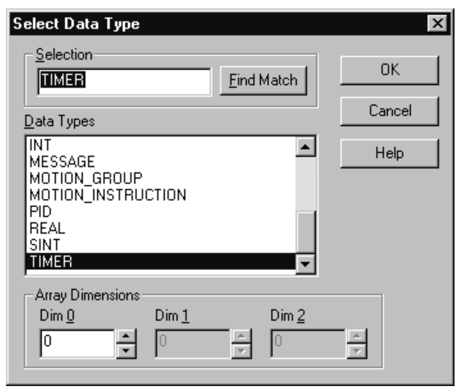
Enter the name of the new tag.

Tab to this column and select the data type.

2. **Select the data type.**

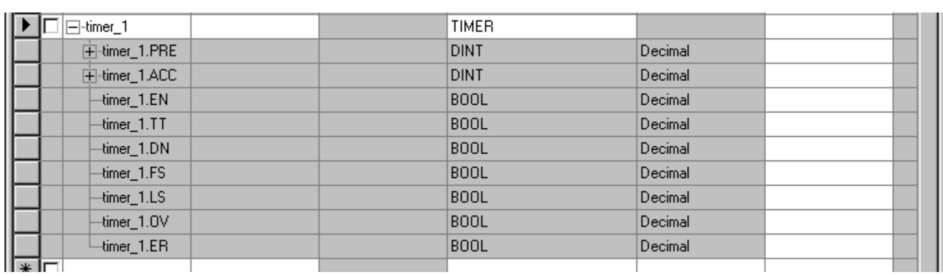
Select TIMER.

Click OK.



The software displays the tag.

Click + to display the members of the TIMER structure.



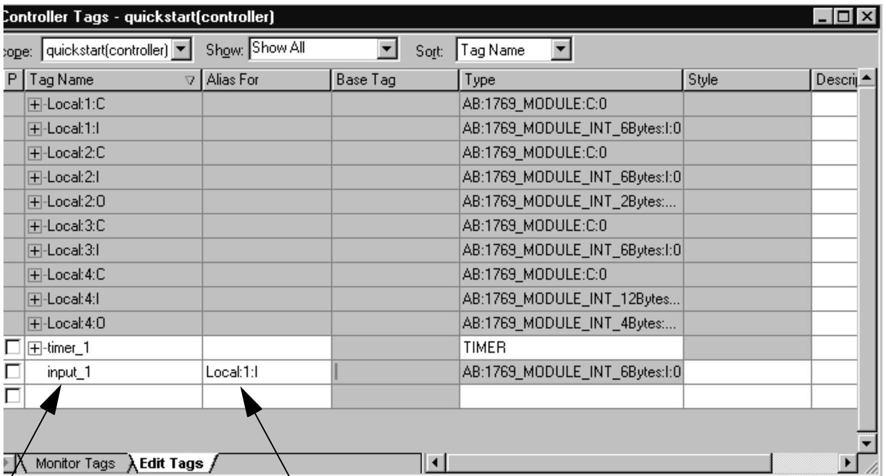
You might have to resize the column to see the tag extensions.

continued

Documenting I/O with alias tags

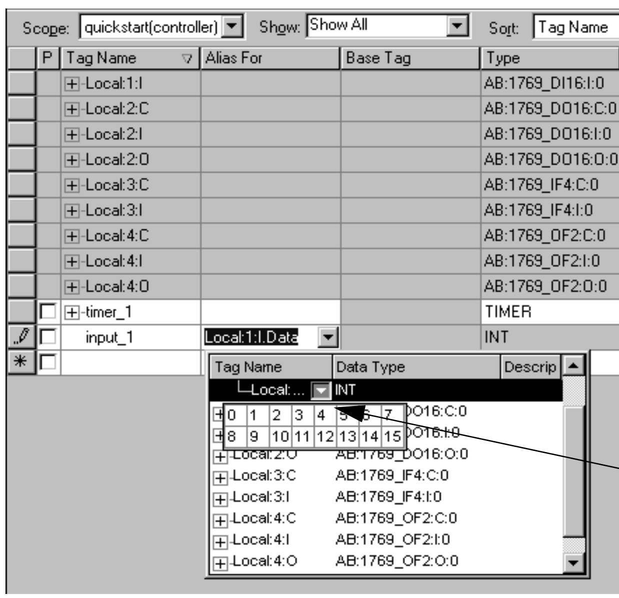
4 **Create tags**

1. **Create an alias tag *input_1* for Local:1:I.Data.1.**



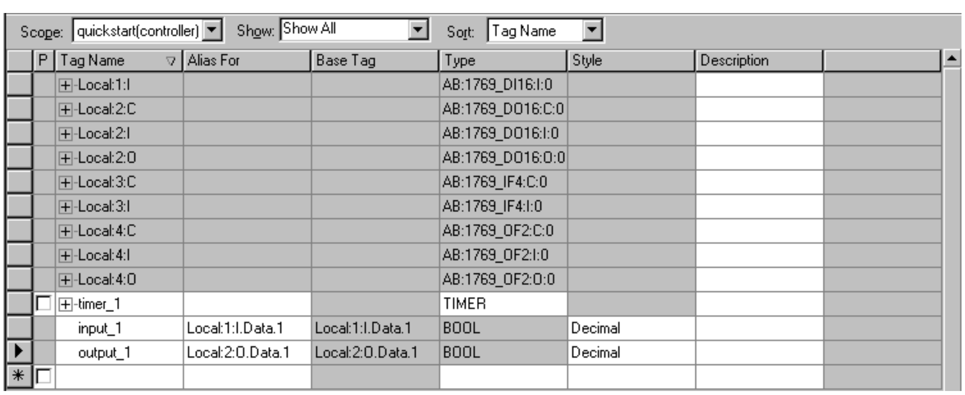
Enter the name of the tag. Tab here or click in the box. Click here to select a tag to reference.

2. **Select an input data word.**



Click here to display a grid of bits and select the input bit.

3. **Repeat steps 1 and 2 above to create an alias tag *output_1* for Local:2:O.Data.1**

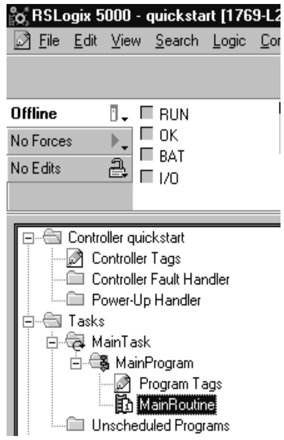


Entering logic

5 Enter logic

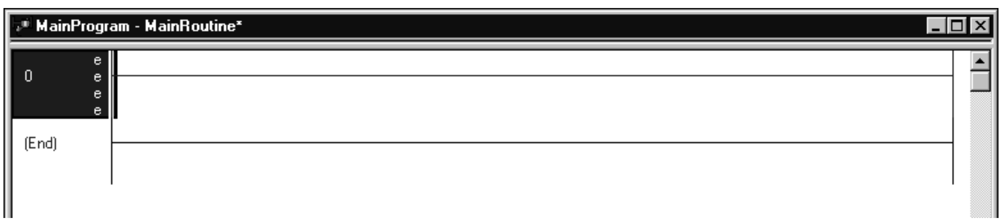
1. Use default task, program, and routine.

When you created the project, the software automatically created a MainTask, MainProgram, and MainRoutine. Use these defaults for this example.

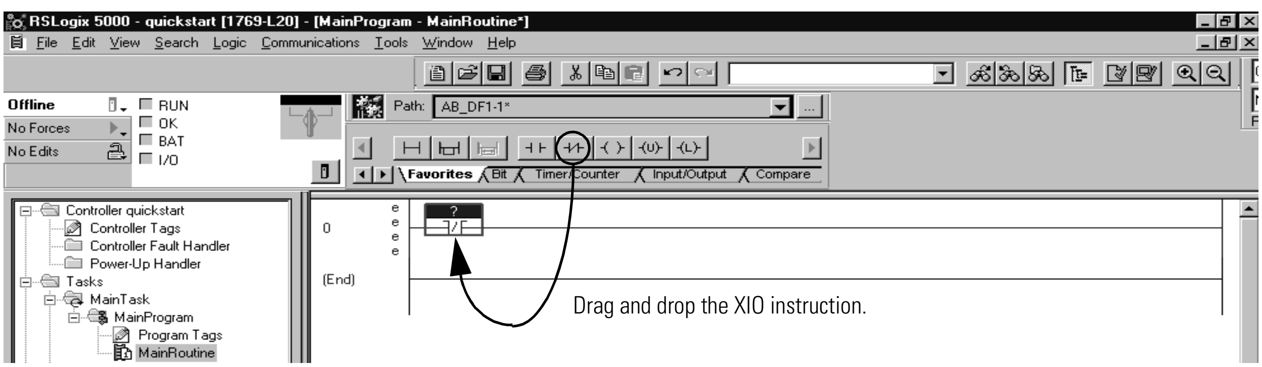


Double-click MainRoutine.

The software displays an empty routine.



2. Enter an XIO instruction.



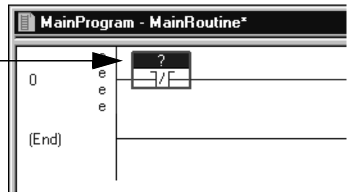
continued

Entering logic (*continued*)

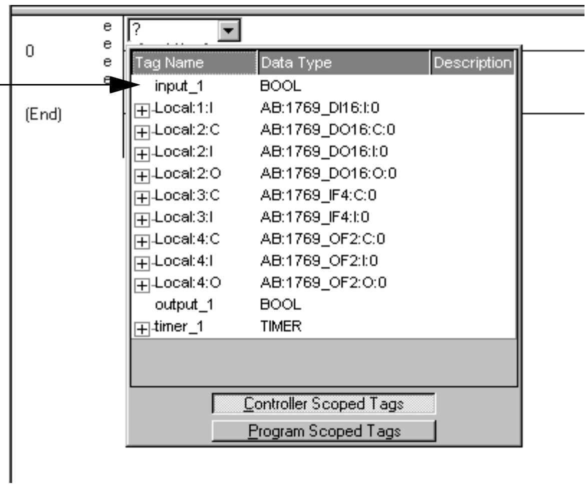
5 **Enter logic**

3. Assign a tag to the XIO instruction.

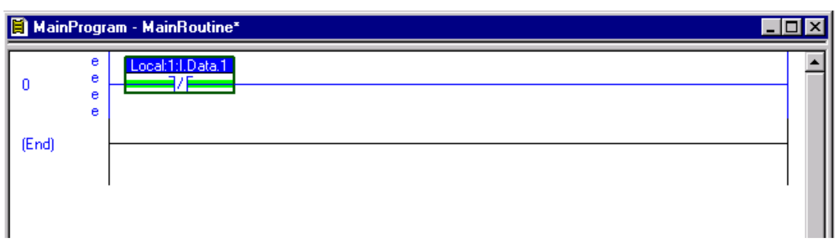
Double-click the tag area of the instruction.



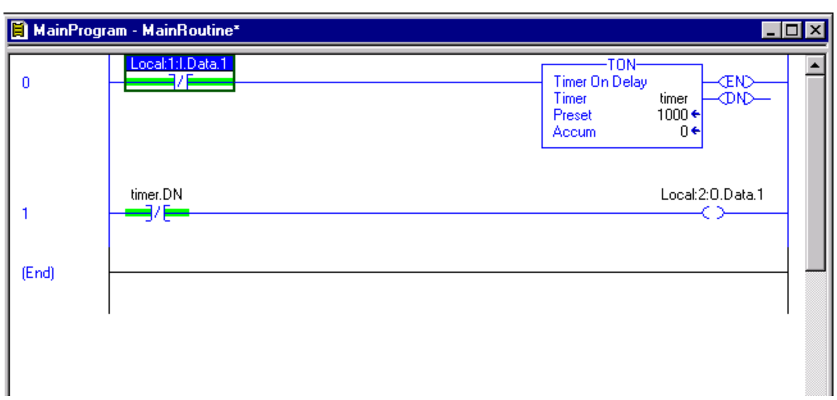
Use the drop-down menu to select *input_1*.



The software displays an incomplete rung.



4. Enter this logic.

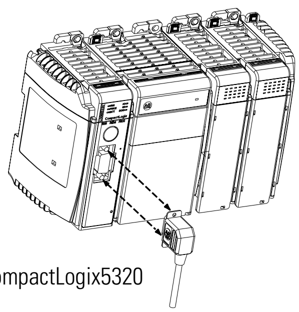


5. To save the project, from the File menu, select Save.

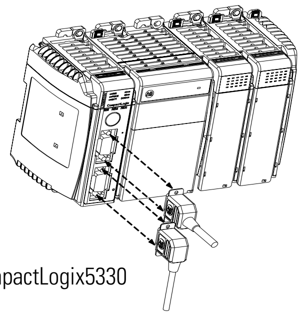
Downloading a project

6 **Download a project**

1. **Make a serial connection from the workstation to the controller.**



CompactLogix5320



CompactLogix5330

2. **Configure an RSLinx communication driver:**

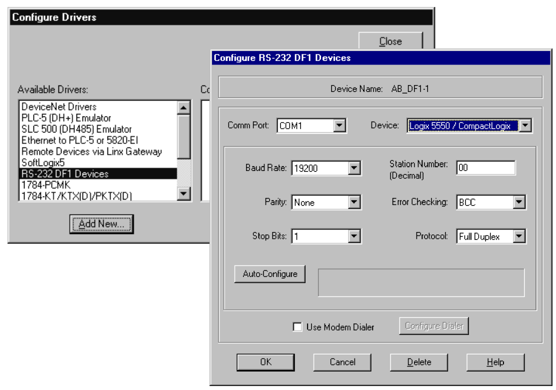
A. In RSLinx software, select Communication → Configure Driver.

B. From the Available Driver Types list, select “RS-232 DF1 Devices” and click Add New.

C. Choose a name for the driver and click OK.

D. Select “Logix5550/CompactLogix” and specify the COM port. Click Autoconfigure to have the software determine the remaining serial settings.

The default value for Error Checking is BCC. The PLC-5 and most peripherals usually use BCC error checking. Most SLC 500 based products use CRC.



Important: If the RSLinx autoconfiguration fails:

A. Press and hold the Channel 0 Default Communication push button on the controller until the Channel 0 Default Communication (DCH0) LED turns on (steady green).

B. Run RSLinx autoconfiguration again to establish a connection.

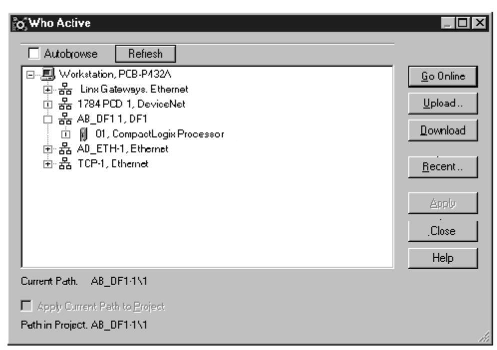
4. **Download the project from the Communications menu.**

A. Make sure the controller is in Program mode.

B. In RSLogix5000 software, select Communication → “Who Active”.

C. Expand the DF1 network and select your controller.

D. Click Download. Confirm the download when prompted.

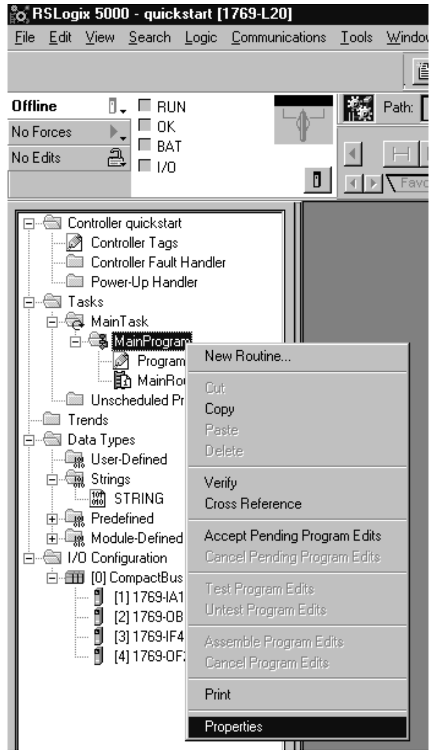


4. **Place the controller in Remote Run mode.**

Viewing program scan time

7 **View status**

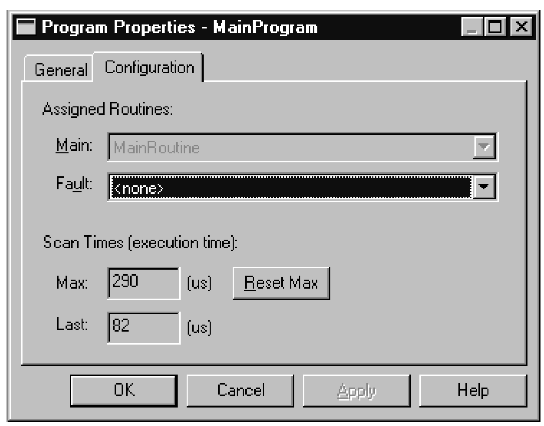
1. View properties for the MainProgram.



A. Place the cursor over the MainProgram folder.

B. Click the right mouse button and select Properties.

2. Select the Configuration tab.



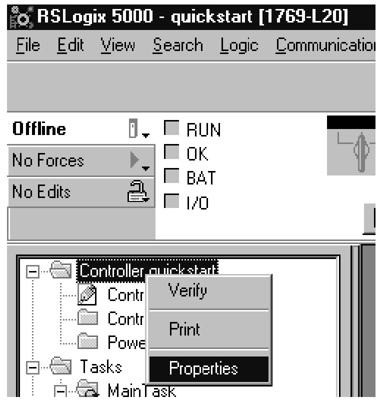
The Configuration tab displays:

- maximum scan time (the largest scan time since the last reset)
- last scan time (the time this task used during the previous scan)

Viewing controller memory usage

7 **View status**

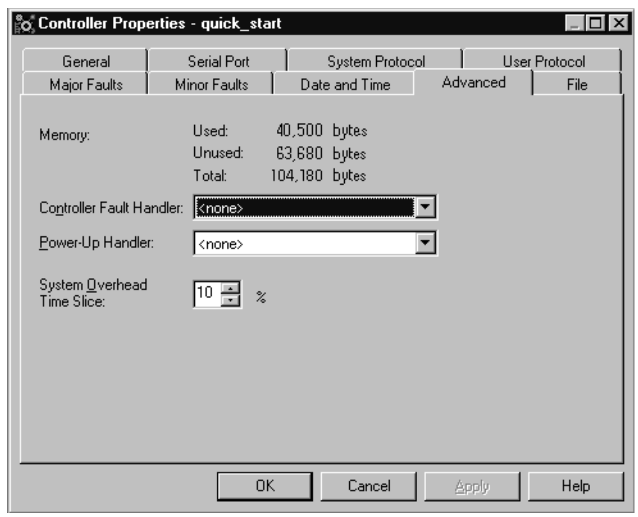
1. **View properties for Controller quick_start.**



A. Place the cursor over the Controller quick_start folder.

B. Click the right mouse button and select Properties.

2. **Select the Advanced tab.**



In addition to other information, the Advanced tab displays controller memory usage.

Important: The amount of memory that the software displays includes both the user available memory and the memory reserved for overhead. See the specifications for your controller to determine how much memory you have available for programming. This dialog box might display a higher number, but the additional memory is required by system overhead and may not be available for programming.

What To Do Next

Once your controller is installed and operating, you can use RSLogix5000 programming software to develop and test your control application.

Use the remaining chapters in this manual as reference material for how the CompactLogix controller operates in the Logix environment.

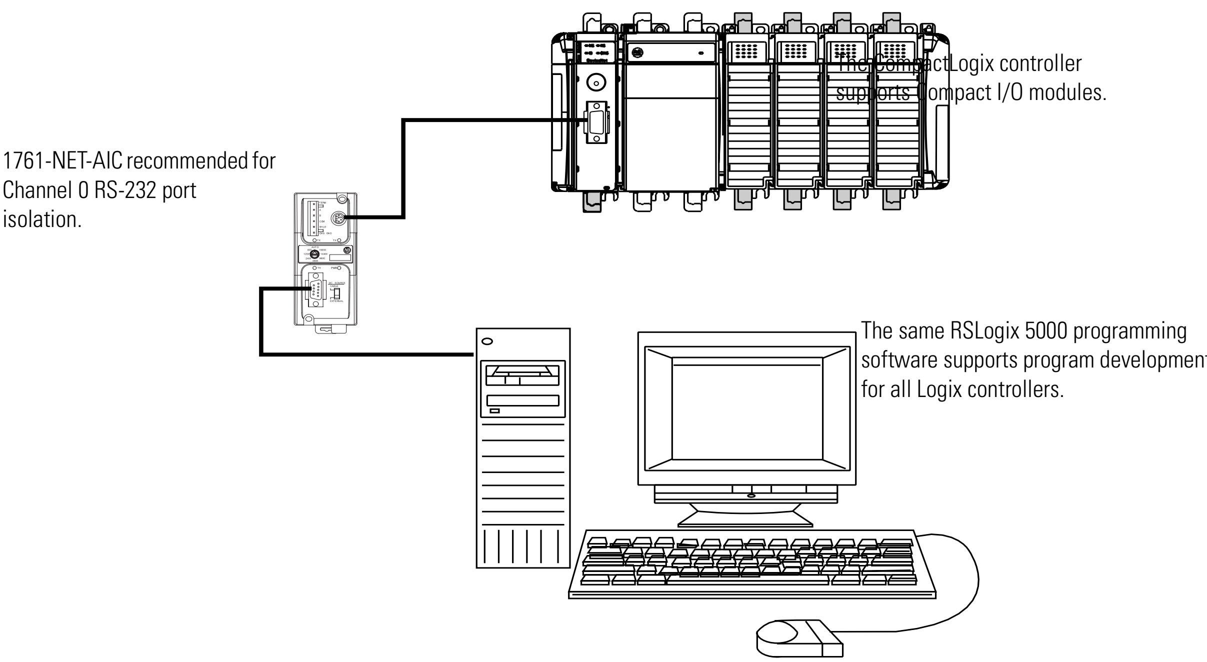
Notes:

What Is CompactLogix?

Using This Chapter

The CompactLogix controller, part of the Logix family of controllers, provides a small, cost-effective system built on these components:

- CompactLogix controller that supports the Logix instruction set.
- RSLogix 5000 programming software that supports every Logix controller.
- Compact I/O modules that provide a compact, DIN-rail or panel-mounted I/O system.
- Serial port that supports multiple communication protocols. The 1769-L20 has one serial port. The 1769-L30 has two. Channel 1 on the 1769-L30 is isolated.
- Communication interface modules provide peer-to-peer communication and program upload/download over DH-485, DeviceNet, or Ethernet.

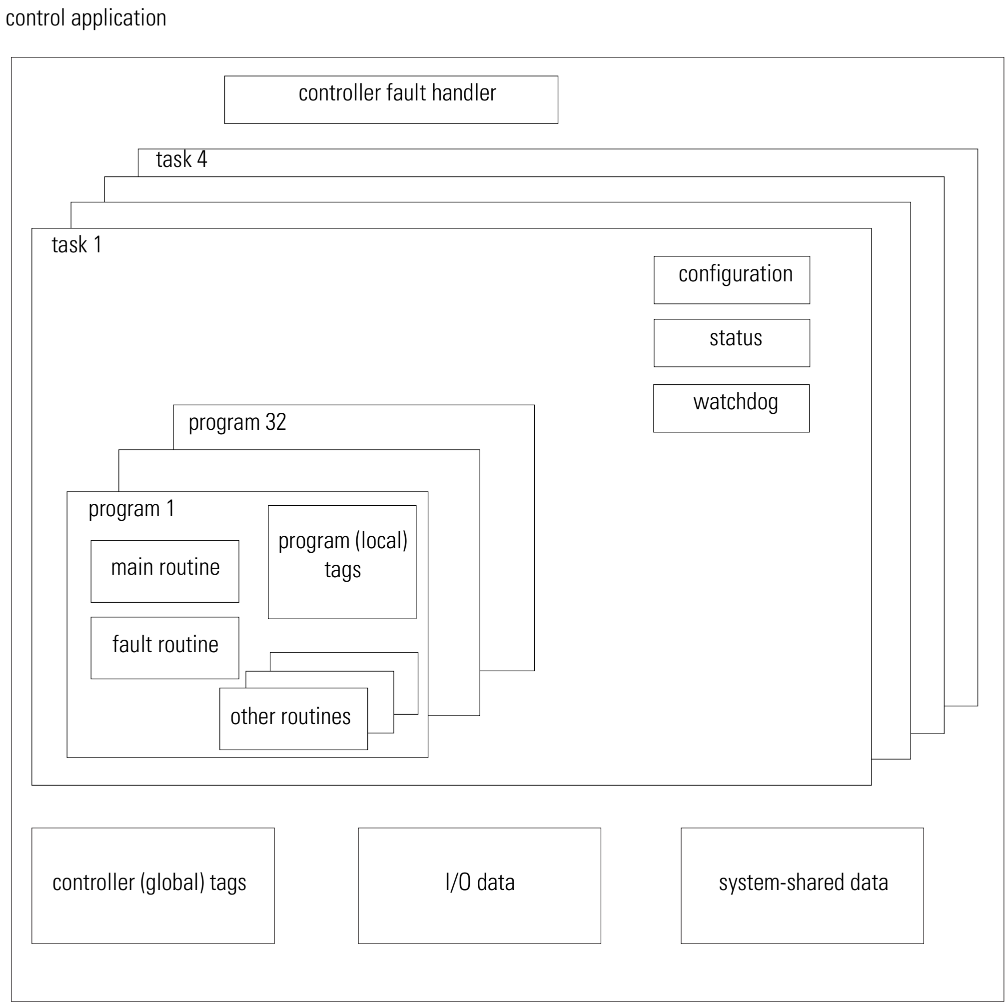


For information about:	See page
developing programs	2-2
selecting a system overhead percentage	2-6

Developing Programs

The controller operating system is a preemptive multitasking system that is IEC 1131-3 compliant. This environment provides:

- tasks to configure controller execution
- programs to group data and logic
- routines to encapsulate executable code written in a single programming language



Defining Tasks

A task provides scheduling and priority information for a set of one or more programs. You can configure tasks as either continuous or periodic. The CompactLogix controller supports as many as 4 tasks, only one of which can be continuous.

A task can have as many as 32 separate programs, each with its own executable routines and program-scoped tags. Once a task is triggered (activated), all the programs assigned to the task execute in the order in which they are grouped. Programs can only appear once in the Controller Organizer and cannot be shared by multiple tasks.

Specifying Task Priorities

Each task in the controller has a priority level. The operating system uses the priority level to determine which task to execute when multiple tasks are triggered. You can configure periodic tasks to execute from the lowest priority of 15 up to the highest priority of 1. A higher priority task will interrupt any lower priority task. The continuous task has the lowest priority and is always interrupted by a periodic task.

The CompactLogix controller uses a dedicated periodic task at priority 7 to process I/O data. This periodic task executes every 2 ms. Its total execution time is as long as it takes to scan the configured I/O modules.

How you configure your tasks affects how the controller receives I/O data. Tasks at priorities 1 to 6 take precedence over the dedicated I/O task. Tasks in this priority range can impact I/O processing time. A task of priority 1 to 6 that requires 1/2 ms to execute and is scheduled to run every millisecond, consumes 1 ms of CPU time. This leaves the dedicated I/O task 1 ms to complete its job of scanning the configured I/O.

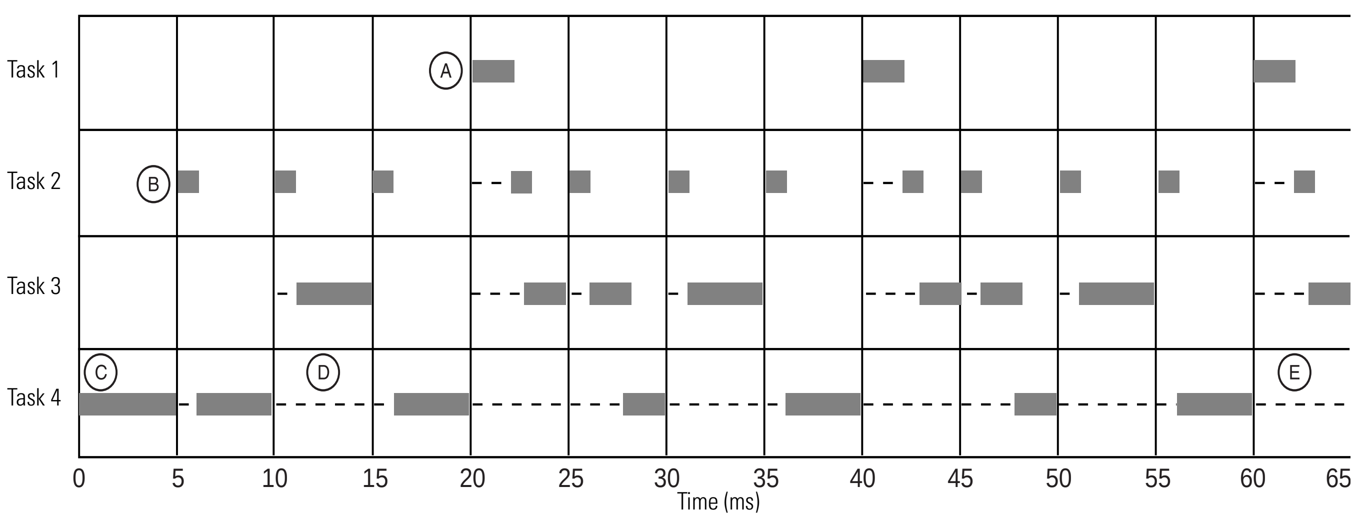
However, if you schedule two high priority tasks (1 to 6) to run every millisecond, and they both require 1/2 ms or more to execute, no CPU time would be left for the dedicated I/O task. Furthermore, if you have so much configured I/O that the execution time of the dedicated I/O task approaches 2 ms (or the combination of the high priority tasks and the dedicated I/O task approaches 2 ms) no CPU time is left for low priority tasks (8 to 15).

TIP

For example, if your program needs to react to inputs and control outputs at a deterministic rate, configure a periodic task with a priority higher than 7 (1 through 6). This keeps the dedicated I/O task from affecting the periodic rate of your program. However, if your program contains a lot of math and data manipulation, place this logic in a task with priority lower than 7 (8 through 15), such as the continuous task, so that the dedicated I/O task is not adversely affected by your program.

The following example shows the task execution order for an application with periodic tasks and a continuous task.

Task:	Priority Level:	Task Type:	Example Execution Time:	Worst Case Completion Time:
1	5	20 ms periodic task	2 ms	2 ms
2	7	dedicated I/O task 2 ms selected RPI	1 ms	3 ms
3	10	10 ms periodic task	4 ms	8 ms
4	none (lowest)	continuous task	25 ms	60 ms



Notes:

A. The highest priority task interrupts all lower priority tasks.

B. The dedicated I/O task can be interrupted by tasks with priority levels 1 to 6. The dedicated I/O task interrupts tasks with priority levels 8 to 15. This task runs at the selected RPI rate scheduled for the CompactLogix system (2ms in this example).

C. The continuous task runs at the lowest priority and is interrupted by all other tasks.

D. A lower priority task can be interrupted multiple times by a higher priority task.

E. When the continuous task completes a full scan it restarts immediately, unless a higher priority task is running.

Defining Programs

Each program contains program tags, a main executable routine, other routines, and an optional fault routine. Each task can schedule as many as 32 programs.

The scheduled programs within a task execute to completion from first to last. Programs that are not attached to any task show up as unscheduled programs. You must specify (schedule) a program within a task before the controller can scan the program.

Defining Routines

A routine is a set of logic instructions in a single programming language, such as ladder logic. Routines provide the executable code for the project in a controller. A routine is similar to a program file or subroutine in a PLC or SLC controller.

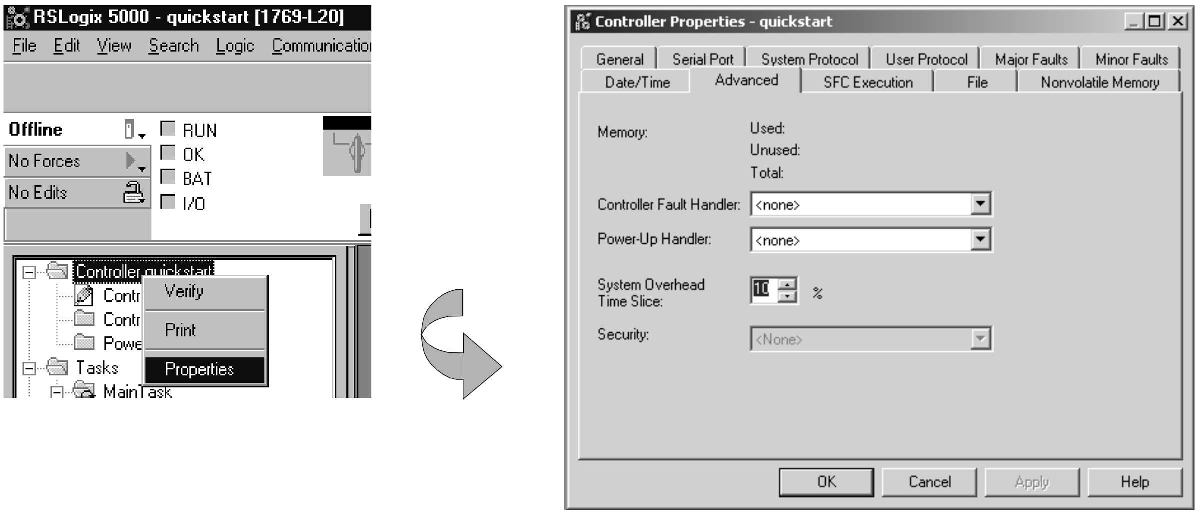
Each program has a main routine. This is the first routine to execute when the controller triggers the associated task and calls the associated program. Use logic, such as the Jump to Subroutine (JSR) instruction, to call other routines.

You can also specify an optional program fault routine. The controller executes this routine if it encounters an instruction-execution fault within any of the routines in the associated program.

Selecting a System Overhead Percentage

The Controller Properties dialog lets you specify a percentage for system overhead. This percentage specifies the percentage of controller time (excluding the time for periodic tasks) that is devoted to communication and background functions.

1. View properties for the controller and select the Advanced tab.



System overhead functions include:

- communicating with programming and HMI devices (such as RSLogix 5000 software)
- responding to messages
- sending messages, including block-transfers

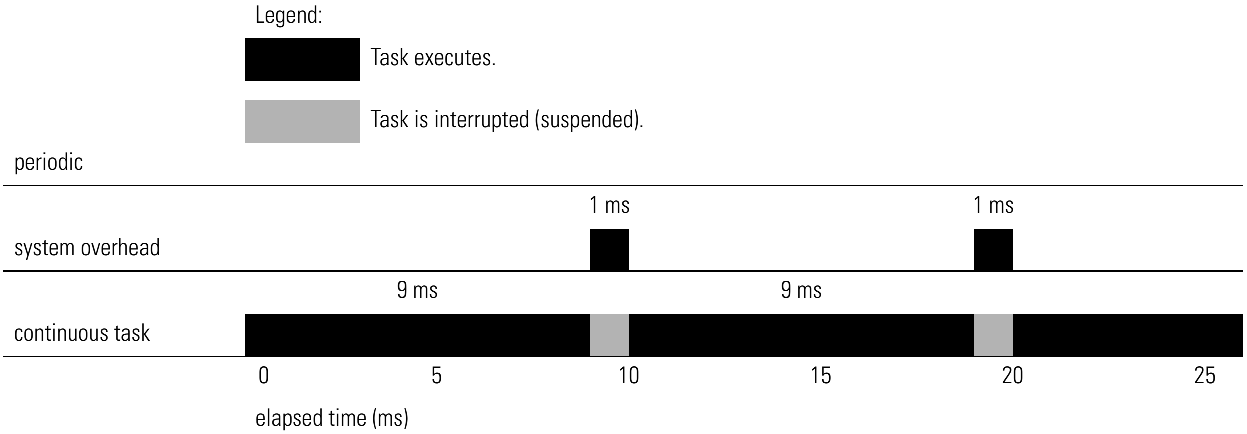
The controller performs system overhead functions for up to 1 ms at a time. If the controller completes the overhead functions in less than 1 ms, it resumes the continuous task.

If communications are not completing fast enough, increase the system overhead percentage. As you increase the system overhead percentage, the overall program scan also increases.

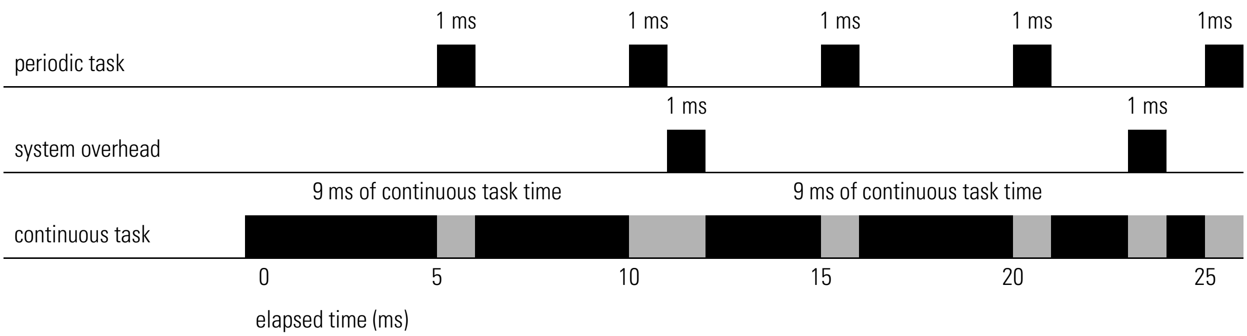
The following table shows the ratio between the continuous task and the system overhead functions:

At this time slice:	The continuous tasks runs for:	And then overhead occurs for up to:
10%	9 ms	1 ms
20%	4 ms	1 ms
33%	2 ms	1 ms
50%	1 ms	1 ms

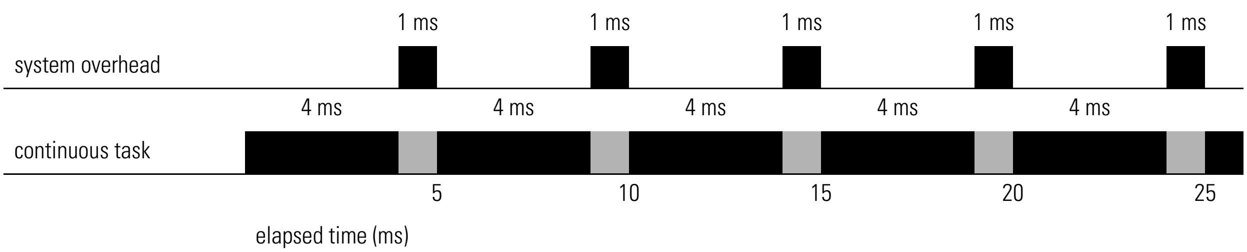
At the default time slice of 10%, system overhead interrupts the continuous task every 9 ms (of continuous task time), as illustrated below.



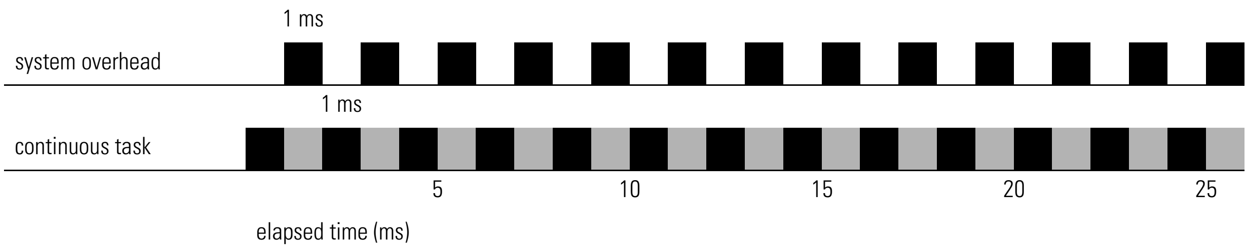
The interruption of a periodic task increases the elapsed time (clock time) between the execution of system overhead, as shown below.



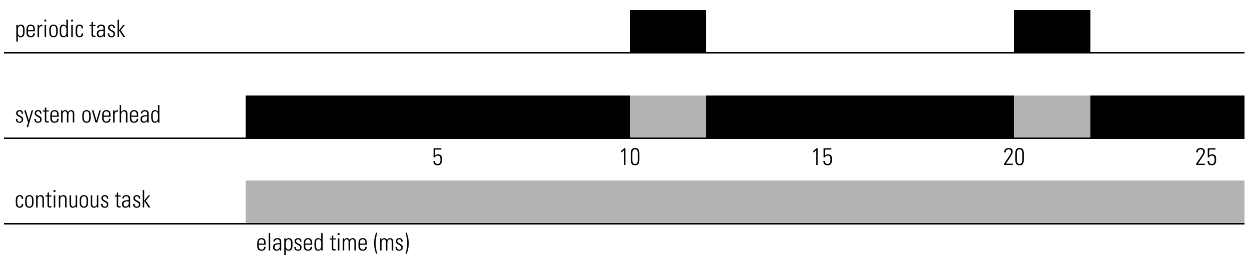
If you increase the time slice to 20%, the system overhead interrupts the continuous task every 4 ms (of continuous task time).



If you increase the time slice to 50%, the system overhead interrupts the continuous task every 1 ms (of continuous task time).



If the controller only contains a periodic task(s), the system overhead timeslice value has no effect. System overhead runs whenever a periodic task is not running.



Placing, Configuring, and Monitoring Local I/O

Using This Chapter

For information about:	See page
Placing local I/O modules	3-1
Verifying I/O layout by adding total words of backplane memory used **Important:** You must verify backplane memory use to make sure that the controller can support the proposed system.	3-4
Determining when the controller updates local I/O	3-5
Configuring a DIN rail	3-5
Configuring local I/O modules	3-7
Inhibiting I/O module operation	3-9
Accessing I/O data	3-11
Monitoring I/O modules	3-14
Configuring modules using the 1769 Generic Profile	3-17

Placing Local I/O Modules

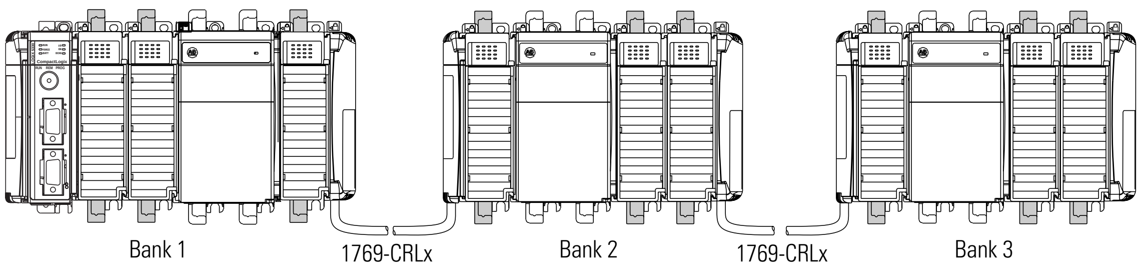
The 1769-L20 controller supports as many as 8 local I/O modules on the CompactBus. The 1769-L30 controller supports up to 16 local I/O modules.

You can also use the 1769-CRR1/-CRR3 or 1769-CRL1/-CRL3 cable to split the system into banks of I/O modules. You can split a bank right after the power supply or after any I/O module. Each bank must contain one power supply. An end cap/terminator must be used on the last I/O bank. 1769-L20 supports a maximum of two banks. 1769-L30 supports a maximum of three banks.

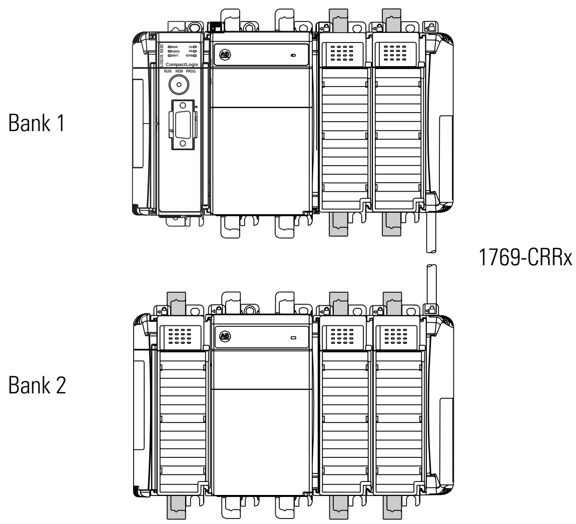
The first bank includes the CompactLogix controller in the far left position. The controller must be located within 4 positions of the bank's power supply. Only one controller may be used in a CompactLogix system.

Each I/O module also has a power supply distance rating (the number of modules from the power supply). The distance rating is printed on each module's label. Each module must be located within its distance rating.

Horizontal Orientation



Vertical Orientation



ATTENTION

CompactLogix does not support Removal and Insertion Under Power (RIUP). While the CompactLogix system is under power, any break in the connection between the power supply and the processor (i.e. removing the power supply, processor, or an I/O module) will clear processor memory (including the user program).

IMPORTANT

While under power, the removal of an end cap or a module (without breaking the connection between the processor and power supply) will fault the controller.

If the controller was in Program mode, cycle power in order to go to run successfully.

If the controller was in Run mode, cycle power. When the I/O LED is on (green, steady) and the OK LED is flashing red, turn the keyswitch from Program to Run and back to Program. When the OK LED turns on (green, steady), turn the keyswitch to Run.

System Power Budget Calculation and Considerations

To validate your system, the total 5V dc current and 24V dc current consumed must be considered. The I/O modules must be distributed such that the current consumed from the left or right side of the power supply never exceeds 2.0A at 5V dc and 1.0A at 24V dc.

Catalog Number	Number of Modules	Module Current Requirements		Calculated Current = (Number of Modules) x (Module Current Requirements)	
		at 5V dc (in mA)	at 24V dc (in mA)	at 5V dc (in mA)	at 24V dc (in mA)
1769-IA8I		90	0		
1769-IA16		115	0		
1769-IM12		100	0		
1769-IQ16		115	0		
1769-IQ6XOW4		105	50		
1769-OA8		145	0		
1769-OA16		225	0		
1769-OB16		200	0		
1769-OB16P		180	0		
1769-OV16		200	0		
1769-OW8		125	100		
1769-OW8I		125	100		
1769-OW16		205	180		
1769-IF4 (A)		120	150		
1769-IF4 (B)		120	60		
1769-OF2 (A)		120	200		
1769-OF2 (B)		120	120		
1769-IF4XOF2		120	160		
1769-IR6		100	45		
1769-IT6		100	40		
1769-HSC		425	0		
1769-L20		600	0		
1769-L30		800	0		
1769-ECR[1]		5	0		
1769-ECL[1]		5	0		
Total Modules:			**Total Current Required[2]:**		

(1) One 1769-ECR or 1769-ECL end cap/terminator is required in the system. The end cap/terminator used is dependent on your configuration.

(2) This number must not exceed the Power Supply Current Capacity listed below.

Power Supply Current Capacity

Specification	1769-PA2	1769-PB2	1769-PA4	1769-PB4
Output Bus Current Capacity (0°C to +55°C)	2A at 5V dc and 0.8A at 24V dc		4A at 5V dc and 2A at 24V dc	
24V dc User Power Capacity (0°C to +55°C)	250 mA (maximum)	not applicable		

Verifying I/O Layout by Adding Total Words of Backplane Memory Used

Each module in a CompactLogix system uses a set amount of backplane memory, in addition to the data that the module stores or transfers. Some modules require a considerable amount of backplane memory. Take this into account when designing your system because it affects how many modules a controller can support.

Each CompactLogix controller supports 256, 16-bit words of backplane data. This table shows how many backplane words each module uses.

Catalog Number:	Number of Modules:	Number of words used:	Calculated number of words:
1769-IA8I		8	
1769-IA16		8	
1769-IM12		8	
1769-IQ16		8	
1769-IQ6XOW4		12	
1769-OA8		12	
1769-OA16		12	
1769-OB16		12	
1769-OB16P		12	
1769-OV16		12	
1769-OW8		12	
1769-OW8I		12	
1769-OW16		12	
1769-IF4		14	
1769-OF2		14	
1769-IF4XOF2		20	
1769-IR6		14	
1769-IT6		16	
1769-HSC		187 (35 words input, 34 words output, 118 words configuration)	
1769-SDN		66 plus total words in scanlist	
system overhead (per controller)		34	34
		Total Words Required:(1)	

(1) The total words required cannot exceed 256 words.

Determining When the Controller Updates I/O

The controller continually scans the control logic. One scan is the time it takes the controller to execute the logic once. Input data transfers to the controller and output data transfers to output modules are asynchronous to the logic scan.

TIP

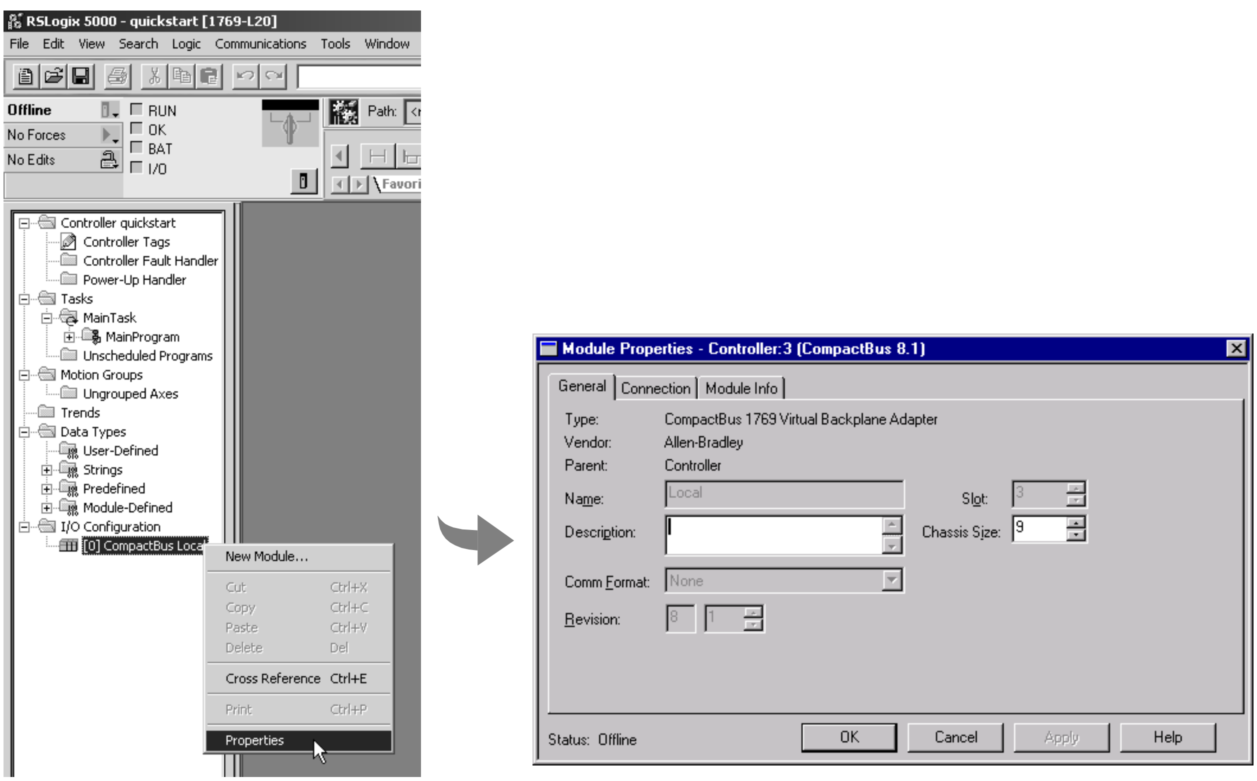
If you need to synchronize I/O to the logic scan, you can use the Synchronous Copy instruction (CPS) to buffer the I/O data.

Refer to the *Logix5000 Controllers Common Procedures Programming Manual*, publication number 1756-PM001B-EN-P for examples of I/O buffering or to the *Logix5000 Controllers General Instruction Set Reference Manual,* publication number 1756-RM003C-EN-P for information on the CPS instruction.

Configuring the CompactBus

When you create a CompactLogix project, the programming software automatically creates the local CompactBus. You must configure the CompactBus.

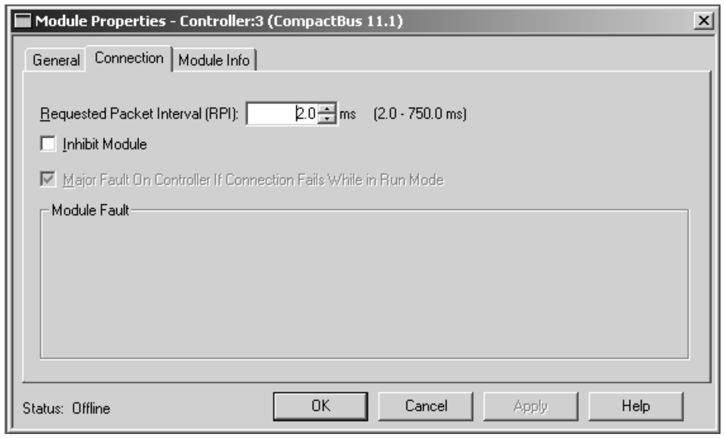
1. In the Controller Organizer, select either the CompactBus. Right-click and select Properties.



On the General tab, specify the size of the Chassis. Enter the number of modules you plan to install. Include the CompactLogix controller in this total, along with a maximum of 8 (1769-L20) or 16 (1769-L30) I/O modules, not including the power supply.

The Comm Format for the CompactBus is automatically set to None and cannot be changed because the controller uses direct connections to each I/O module.

Using the Connection tab, you can specify the RPI for the systems and choose to inhibit or uninhibit the CompactBus.



The RPI you specify here is the RPI for every 1769 module in this controller's system. Specify an RPI from 2-750ms for the system. You do not specify individual RPI values for each module.

By inhibiting and uninhibiting the CompactBus, you can write new configuration data to the entire system at once.

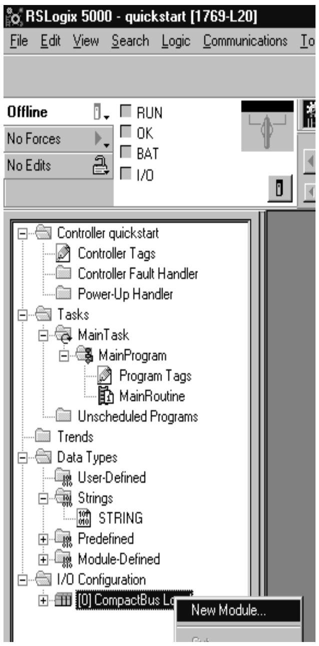
The controller's response to a CompactBus connection failure is fixed to always fault the controller. It is not configurable.

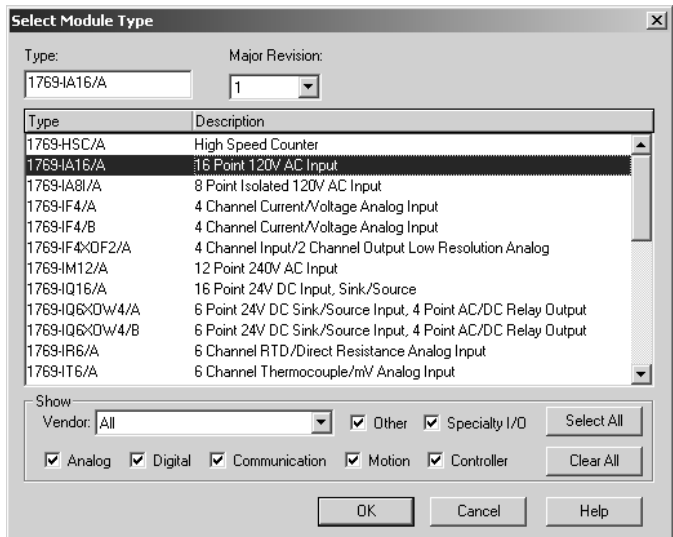
Configuring Local I/O Modules

Use your programming software to configure the I/O modules for the controller.

1. In the Controller Organizer, select the CompactBus. Right-click the selected rail and select New Module.

2. Select the module (1769-IA16 in this example).

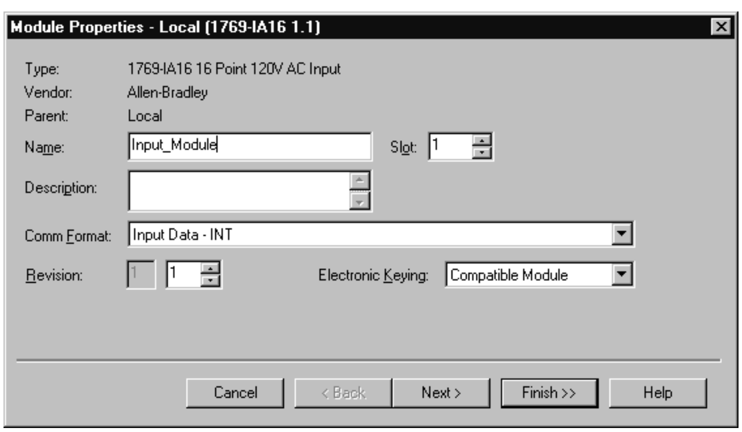




Click OK.

3. Configure the module. Use the module wizard to specify characteristics for the module. Click Next to continue through the wizard.

Click Finish when you are done. The completed module appears in the Controller Organizer.



The selection you make for the Comm Format determines the connections required for the I/O module. Once you complete adding a module, you cannot change this selection. See page 3-8.

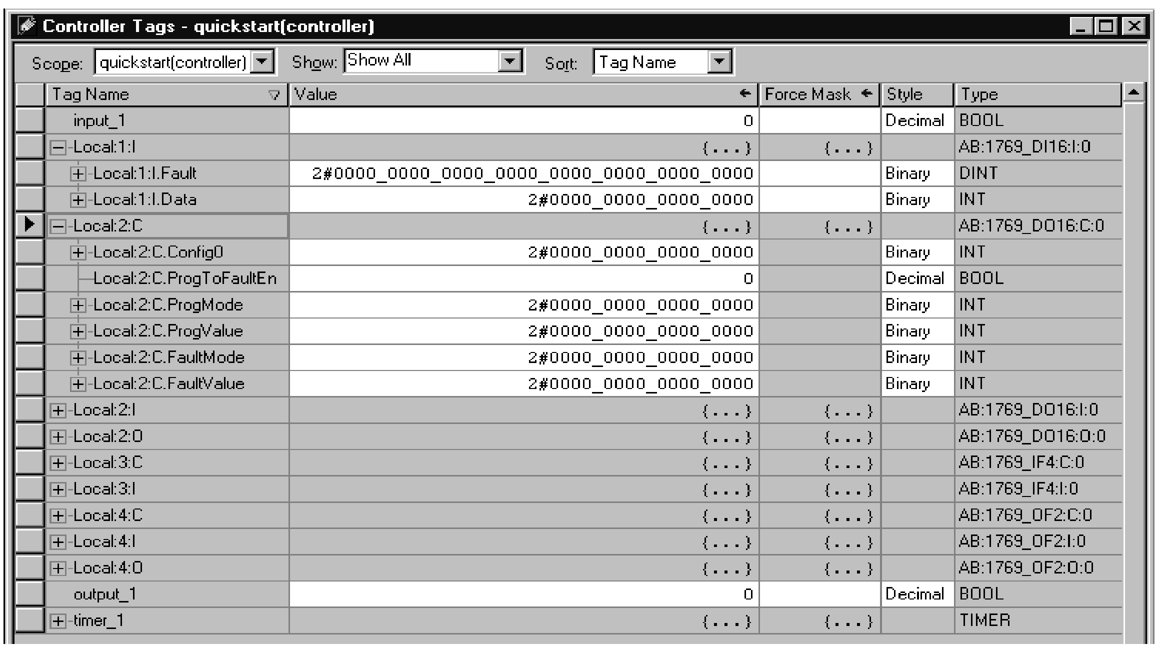
Communication Formats

The communication format determines the data structure the I/O module uses. Each format supports a different data structure. Presently, the CompactLogix system supports two data formats:

- Input Data – INT (for 1769 input modules)
- Data – INT (for 1769 output modules)

TIP The CompactLogix controller must own its local I/O modules. No other Logix-based controller can own the local CompactLogix I/O.

The communication format determines the tag structure that is created for the module. Assume that a 1769-IA16 Input module is in slot 1. The software creates the appropriate tags using the slot number to differentiate the tags for this example module from any other module.


Controller Tags - quickstart(controller)

Scope: quickstart(controller) Show: Show All Sort: Tag Name

Tag Name	Value	Force Mask	Style	Type
input_1	0		Decimal	BOOL
-Local:1:I	{...}	{...}		AB:1769_DI16:I:0
+Local:1:I.Fault	2#0000_0000_0000_0000_0000_0000_0000_0000		Binary	DINT
+Local:1:I.Data	2#0000_0000_0000_0000		Binary	INT
-Local:2:C	{...}	{...}		AB:1769_DO16:C:0
+Local:2:C.Config0	2#0000_0000_0000_0000		Binary	INT
Local:2:C.ProgToFaultEn	0		Decimal	BOOL
+Local:2:C.ProgMode	2#0000_0000_0000_0000		Binary	INT
+Local:2:C.ProgValue	2#0000_0000_0000_0000		Binary	INT
+Local:2:C.FaultMode	2#0000_0000_0000_0000		Binary	INT
+Local:2:C.FaultValue	2#0000_0000_0000_0000		Binary	INT
+Local:2:I	{...}	{...}		AB:1769_DO16:I:0
+Local:2:O	{...}	{...}		AB:1769_DO16:O:0
+Local:3:C	{...}	{...}		AB:1769_IF4:C:0
+Local:3:I	{...}	{...}		AB:1769_IF4:I:0
+Local:4:C	{...}	{...}		AB:1769_OF2:C:0
+Local:4:I	{...}	{...}		AB:1769_OF2:I:0
+Local:4:O	{...}	{...}		AB:1769_OF2:O:0
output_1	0		Decimal	BOOL
+timer_1	{...}	{...}		TIMER

Inhibiting I/O Module Operation

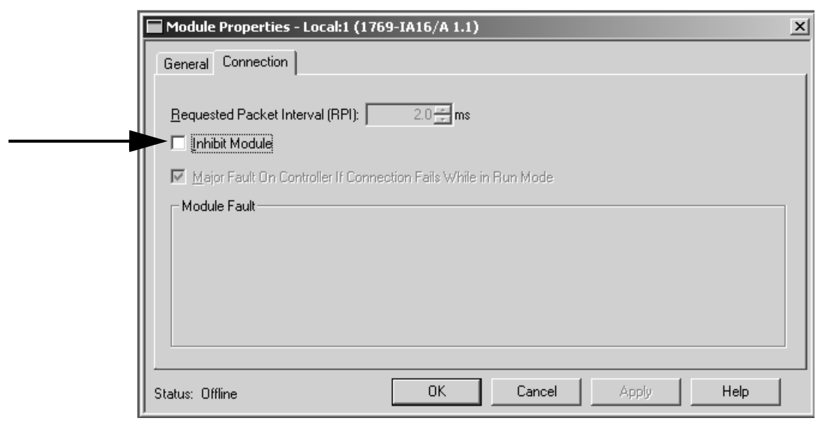
In some situations, such as when initially commissioning a system, it is useful to disable portions of a control system and enable them as you wire up the control system. The controller lets you inhibit individual modules or groups of modules, which prevents the controller from trying to communicate with these modules. Inhibiting a module shuts down the connection from the controller to that module.

When you create an I/O module, it defaults to being not inhibited. You can change an individual module's properties to inhibit a module.

ATTENTION

Inhibiting a module closes the connection to the module and prevents communication of I/O data.

On the Connection tab of the Module Properties dialog, you can select to inhibit that specific module.



TIP

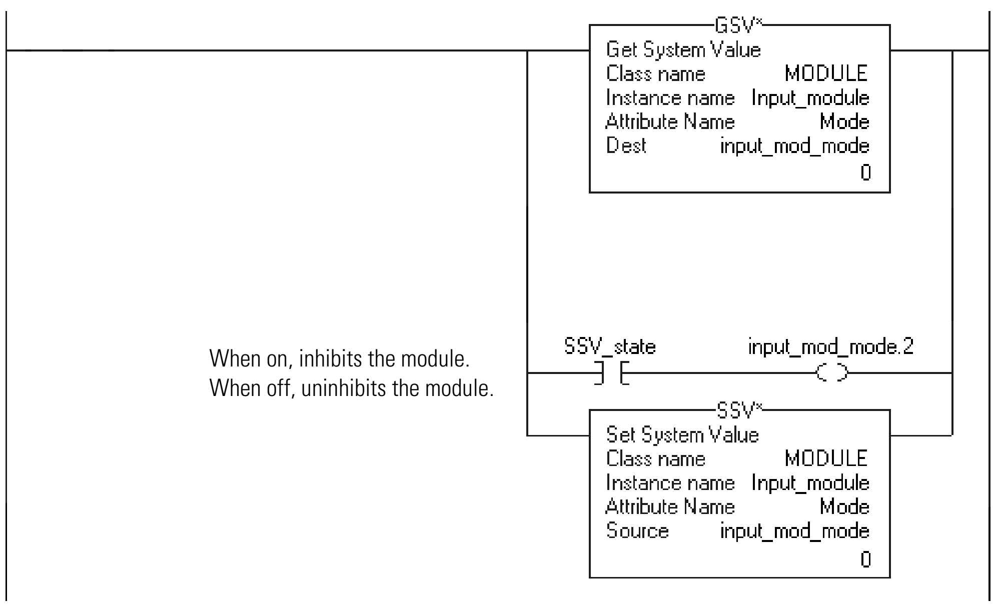
To easily inhibit all local I/O modules, you can inhibit the CompactBus, which in turn inhibits all the modules on that bus. See Configuring the CompactBus on page 3-5.

When you select to inhibit a module, the controller organizer displays a yellow attention symbol ⚠ over the module.

If you are:	**Inhibit a module to:**
offline	put a place holder for a module you are configuring. The inhibit status is stored in the project. When you download the project, the module is still inhibited.
online	stop communication to a module. If you inhibit a module while you are connected to the module, the connection to the module is closed. The module's outputs turn off. If you inhibit a module but a connection to the module was not established (perhaps due to an error condition or fault), the module is inhibited. The module status information changes to indicate that the module is inhibited and not faulted. If you uninhibit a module (clear the check box), and no fault condition occurs, a connection is made to the module and the module is dynamically reconfigured with the configuration you created for that module. If you uninhibit the module and a fault condition occurs, a connection is not made to the module. The module status information changes to indicate the fault condition.

To inhibit a module from logic, you must first read the Mode attribute for the module using a GSV instruction. Set bit 2 to the inhibit status (1 to inhibit or 0 to uninhibit). Use a SSV instruction to write the Mode attribute back to the module. For example:

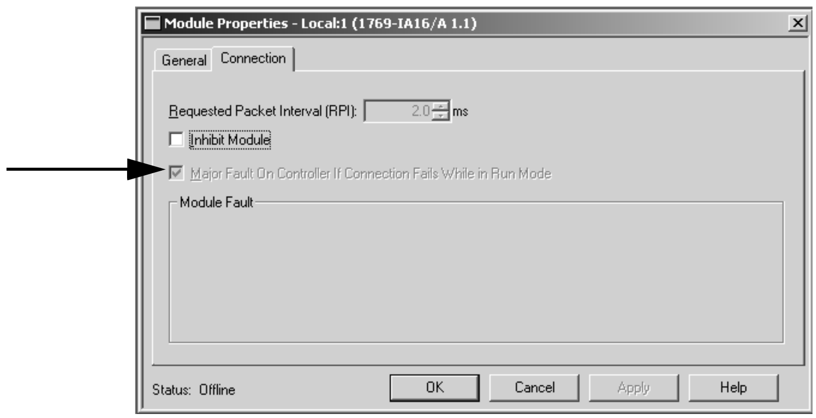
The GSV instruction gets the current status of the module named "input_module." The SSV instruction sets the state of "input_module" as either inhibited or uninhibited.



Configuring the Module's Response to a Connection Failure

Using the Connection tab in the ControlLogix and FlexLogix systems, you can also configure modules to generate a major fault in the controller if they lose their connection with the controller. This feature, however, is not available in a CompactLogix system because the controller's response to a CompactBus connection failure is fixed to always fault the controller. The CompactBus setting supersedes the individual module's setting.

IMPORTANT The controller's response to a connection failure of any I/O module is fixed to always fault the controller.



Accessing I/O Data

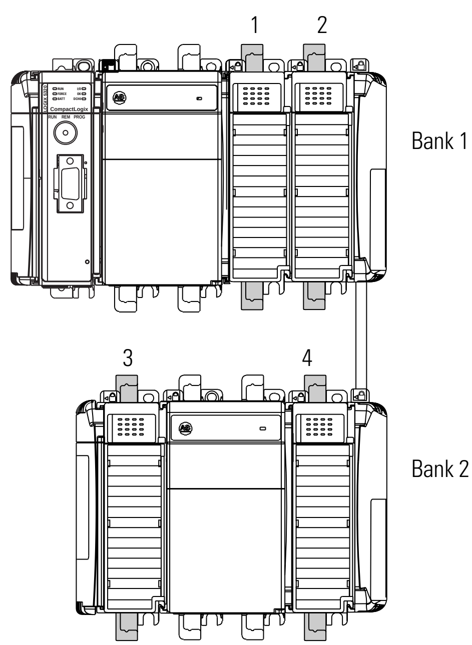
The programming software displays I/O data as structures of multiple tags that depend on the specific features of the I/O module. The names of the data structures are based on the location of the I/O module. The programming software automatically creates the necessary structures and tags when you configure the module. Each tag name follows this format:

Location:SlotNumber:Type.MemberName.SubMemberName.Bit where:

This address variable:	Is:
Location	Identifies network location LOCAL = local chassis
SlotNumber	Slot number of I/O module in its chassis
Type	Type of data I = input O = output C = configuration
MemberName	Specific data from the I/O module; depends on the type of data the module can store For example, Data and Fault are possible fields of data for an I/O module. Data is the common name for values that are sent to or received from I/O points.
SubMemberName	Specific data related to a MemberName.
Bit (optional)	Specific point on the I/O module; depends on the size of the I/O module (0-31 for a 32-point module)

The following examples show addresses for data in a CompactLogix system.

EXAMPLE I/O module on the local CompactBus utilizing two banks



Sample tag names for this example:

Location:	Example Tag Name:
input module in slot 1, LOCAL Bank 1	Local:1:C Local:1:I
output module in slot 2, LOCAL Bank 1	Local:2:C Local:2:I Local:2:O
analog input module in slot 3, LOCAL Bank 2	Local:3:C Local:3:I
analog output module in slot 4, LOCAL Bank 2	Local:4:C Local:4:I Local:4:O

Using Aliases to Simplify Tag Names

An alias lets you create a tag that represents another tag. This is useful for defining descriptive tag names for I/O values. For example:

Example:		Description:
I/O structure	Local:1:I:Data[0].0 Local:1:I:Fault.0	The aliases describe the specific I/O points.
alias	light_on = Local:1:I:Data[0].0 light_off = Local:1:I:Fault.0	

Direct Connections for I/O Modules

The CompactLogix system uses direct connections to transmit I/O data. Each local I/O module utilizes a direct connection to the CompactLogix controller. A direct connection is a real-time, data transfer link between the controller and an I/O module. The controller maintains and monitors the connection between the controller and the I/O module. Any break in the connection, such as a module fault, causes the controller to set fault status bits in the input data area associated with the module.

ATTENTION

CompactLogix does not support Removal and Insertion Under Power (RIUP). While the CompactLogix system is under power, any break in the connection between the power supply and the processor (i.e. removing the power supply, processor, or an I/O module) will clear processor memory (including the user program).

Monitoring I/O Modules

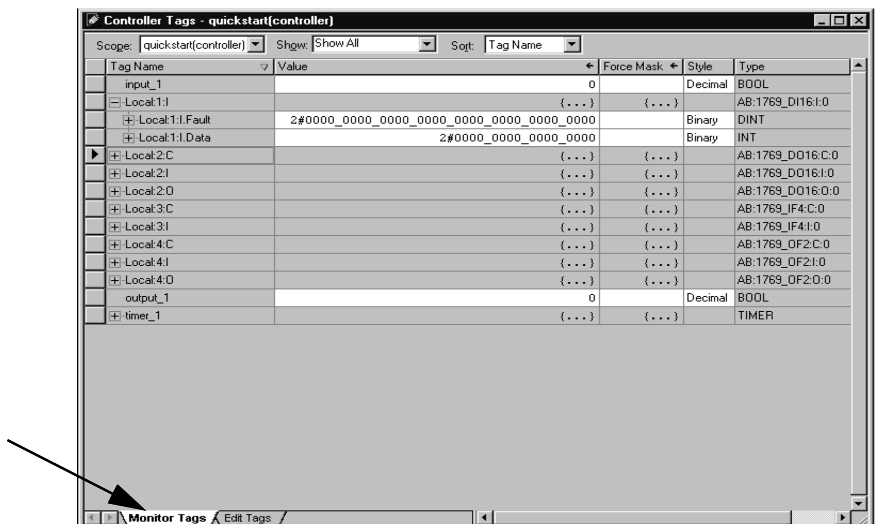
The CompactLogix controller offers different levels at which you can monitor I/O modules. You can:

- use the programming software to display fault data (See Displaying Fault Data on page 3-14)
- program logic to monitor fault data so you can take appropriate action (Refer to *Logix5000 Controllers Common Procedures Programming Manual*, publication number 1756-PM001B-EN-P, for examples.)

Displaying Fault Data

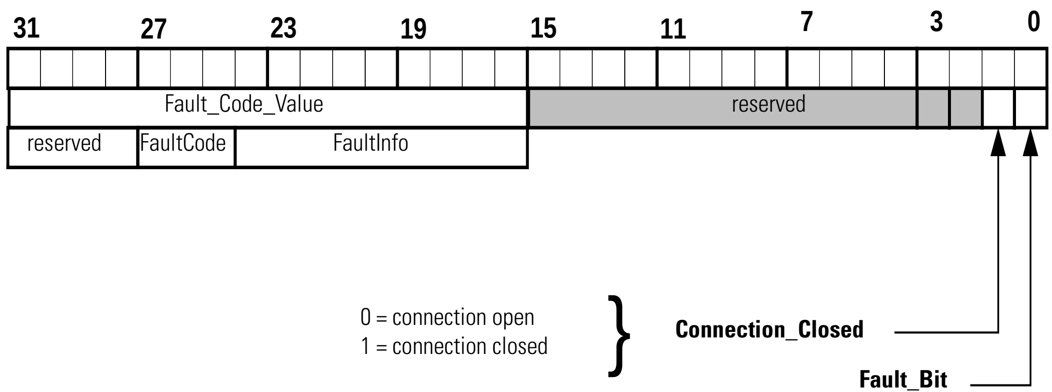
Fault data for certain types of module faults can be viewed through the programming software.

To view this data, select Controller Tags in the Controller Organizer. Right-click to select Monitor Tags.



The display for the fault data defaults to decimal. Change it to Hex to read the fault code.

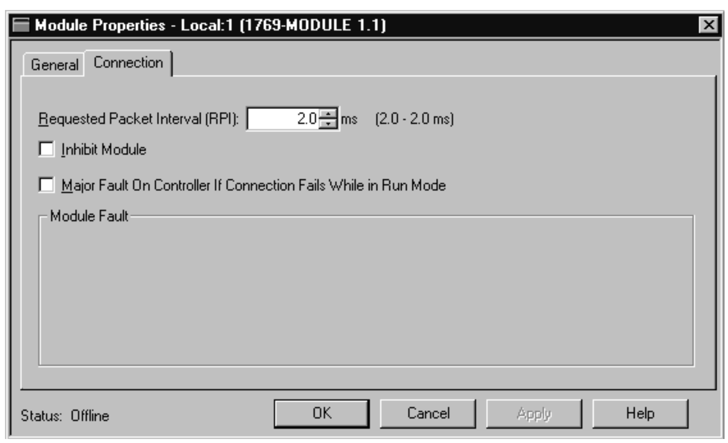
If the module faults, but the connection to the controller remains open, the controller tags database displays the fault value 16#0E01_0001. The fault word uses this format:



Where:

Bit	Description
Fault_Bit	This bit indicates that at least one bit in the fault word is set (1). If all the bits in the fault word are cleared (0), this bit is cleared (0).
Connection_Closed	This bit indicates whether the connection to the module is open (0) or closed (1). If the connection is closed (1), the Fault_Bit it set (1).

You can also view module fault data on the Connection tab of the Module Properties screen.



See your 1769 module's user documentation for a description of module faults. To recover from module faults, correct the module fault condition and send new data to the module by downloading the user program with configuration data, inhibiting and then uninhibiting the module, or cycling power.

End-Cap Detection and Module Faults

If a module that is not adjacent to an end cap experiences a fault and the connection to the controller is not broken, only the module enters the fault state.

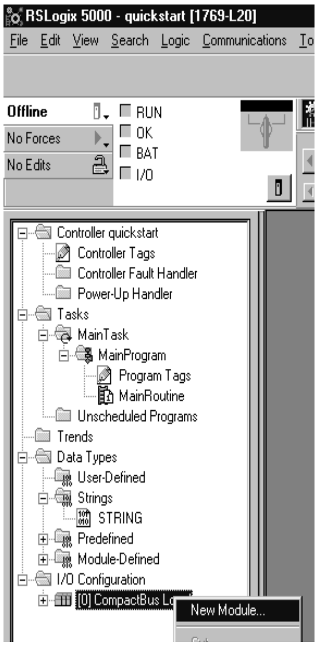
If a module that is adjacent to an end cap experiences a fault, both the module and the controller transition to the fault state.

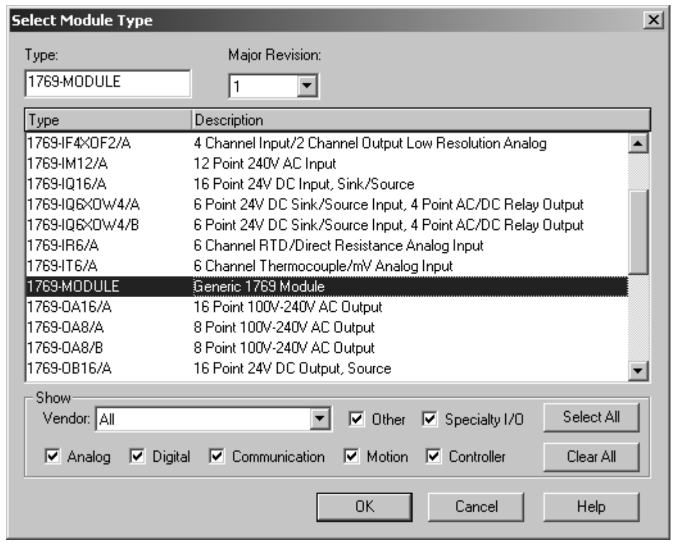
Configuring I/O Modules Using the Generic Profile

Use the Generic 1769 Module profile only when a 1769 I/O module does not appear in the list of modules to add to the Controller Organizer. To configure a 1769 I/O module for a CompactLogix controller using the generic profile:

1. In the Controller Organizer, select the CompactBus. Right-click the selected rail and select New Module.

2. Select the 1769-MODULE (Generic 1769 Module profile).

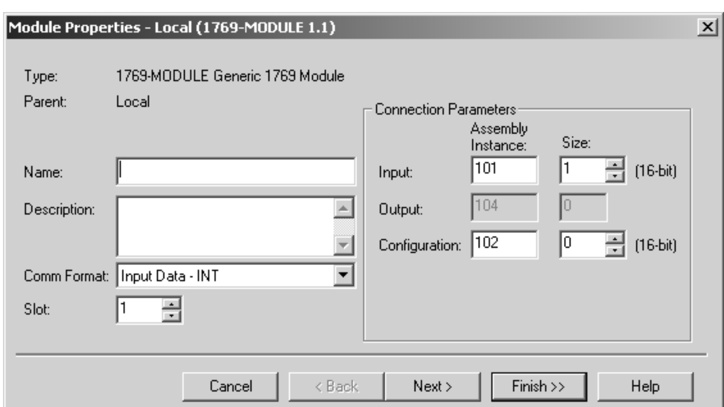




Click OK.

3. Configure the module. Use the module wizard to specify characteristics for the module. Click Next to continue through the wizard.

Click Finish when you are done. The completed module appears in the Controller Organizer.



The generic profile requires you to specify more parameters of the module.

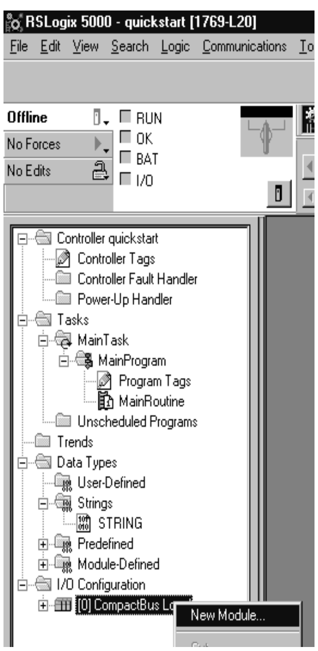
On the generic profile screen, you define the parameters of the module.

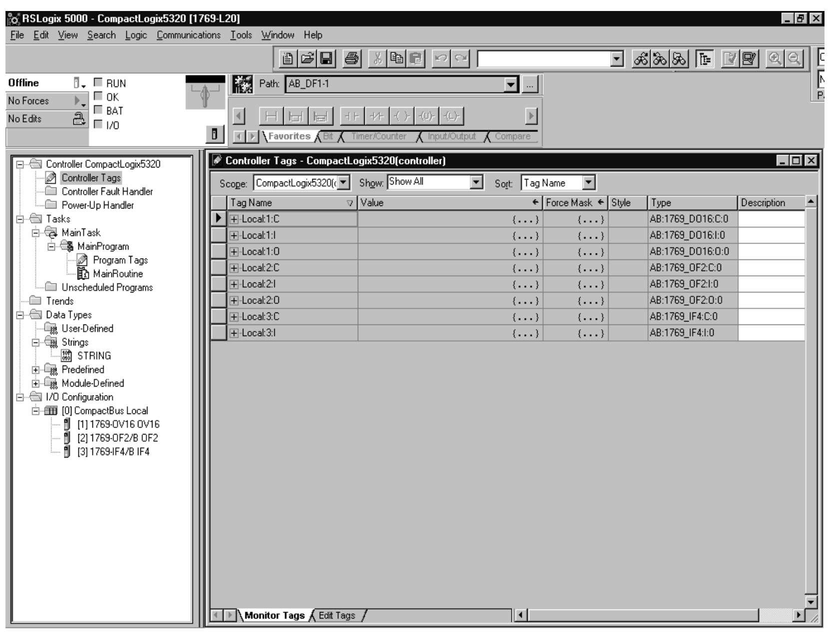
In this field:	Specify:
Name	name of the module
Description	(optional) provide more details about the module
Comm Format	communication format 1769 analog output modules, digital output modules, analog combination modules, and digital combination modules, use Data – INT. 1769 analog input modules and digital input modules use Input Data – INT.
Slot	slot placement of the module on the CompactBus
Connection Parameters Input Output Configuration	connection information unique to the module The documentation for module should list the assembly instance and size numbers for the input, output, and configuration parameters.

Entering the Configuration Information for the Module

Once you have configure a module using the generic profile, you must enter the configuration information for the module into the tag database. The configuration information is downloaded to the module at program download, power up, and whenever a module is inhibited and then uninhibited.

1. In the Controller Organizer, double-click on Controller Tags.
2. Edit the tags for the module so that the tags contain the appropriate configuration information.





RSLogix 5000 programming software automatically create tags for configured I/O modules. All local I/O addresses are preceded by the word Local. These addresses have the following format:

- Input Data: Local:*s*:I
- Output Data: Local:*s*:O
- Configuration Data: Local:*s*:C

Where *s* is the slot number assigned the I/O module.

Open the configuration tag for that module by clicking on the plus sign to the left of its configuration tag in the tag database. The configuration information for the generic profile depends on the module. See the documentation on the I/O module for the appropriate configuration information.

Notes:

Communicating with Devices on a Serial Link

Using This Chapter

For information about:	See page
Default communication configuration	4-1
Configuring your system for a serial link	4-3
Example 1: Workstation directly connected to a CompactLogix controller	4-11
Example 2: Workstation remotely connected to a CompactLogix controller	4-12
Example 3: CompactLogix controller communicating with a bar code reader	4-16

Default Communication Configuration

The CompactLogix controllers have the following default communication configurations.

Parameter	Channel 0 Default	Channel 1 Default (CompactLogix5330 only)
Baud Rate	19.2K	19.2K
Parity	none	none
Station Address	0	0
Control Lines	no handshaking	no handshaking
Error Detection	BCC	BCC
Embedded Responses	auto detect	auto detect
Duplicate Packet (Message) Detect	enabled	enabled
ACK Timeout	50 counts	50 counts
NAK Receive Limit	3 retries	3 retries
ENQ Transmit Limit	3 retries	3 retries
Data Bits	8	8
Stop Bits	1	1
Protocol	DF1 full-duplex	DF1 full-duplex

TIP

Node Address is part of the default configuration. Changing the node address will result in the DCH0 LED turning off.

System Protocol Options

The table below shows the system modes supported by Channels 0 and 1.

Channel 0	Channel 1 (CompactLogix5330 only)
DF1 full-duplex	DF1 full-duplex
DF1 master	DF1 half-duplex master
DF1 slave	DF1 half-duplex slave
DH-485	DH-485
ASCII	

TIP

When using MSG instructions to send commands out the CompactLogix serial ports, Channel 0 is Port 2 and Channel 1 is Port 3. This information is required on the Path tab for the MSG instruction.

Using the Channel 0 Default Communication Push Button

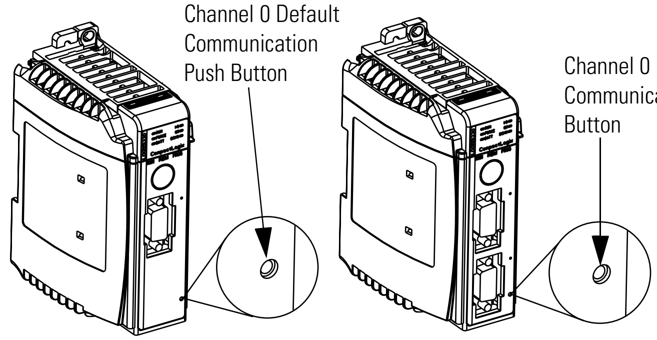
Use the Channel 0 Default Communication Push Button to change from the user-defined communication configuration to the default communications configuration. *Hold the button* until the Channel 0 Default Communications (DCH0) LED turns on (green, steady) showing that the default communication configuration is active.

TIP

Before pressing the Default Communication Push Button, be sure to note the present communication configuration for Channel 0. Pushing the Default Communication Push Button resets all configured parameters back to their default settings. To return the channel to its user-configured parameters, you must enter them manually while online with the controller or download them as part of a Logix Project file.

To accomplish this online, enter the Controller Properties screen under the Serial Port, System Protocol and User Protocol tabs.

The Channel 0 Default Communication Push Button is located on the front of the controller in the lower right corner.



The Channel 0 Default Communication Push Button is recessed. It only resets Channel 0 (which is the top serial connection on a 1769-L30 controller).

Configuring Your System for a Serial Link

For the CompactLogix controller to operate on a serial network, you need:

- a workstation with a serial port
- RSLinx software to configure the serial communication driver
- RSLogix5000 programming software to configure the serial port of the controller

IMPORTANT Limit the length of serial (RS-232) cables to 15.2m (50 ft.).

Step 1: Configure the Hardware

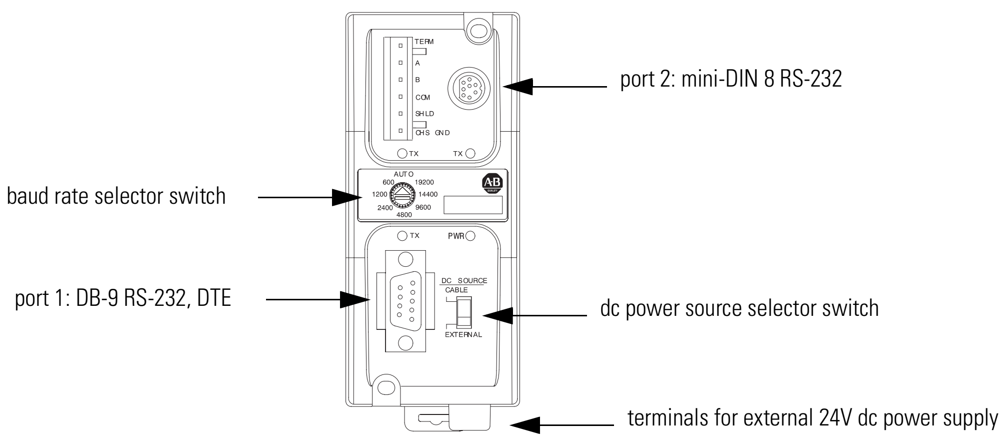
The Channel 0 RS-232 port on a CompactLogix controller is a non-isolated serial port built-in to the front of the controller. The Channel 1 RS-232 port on the 1769-L30 controller is isolated.

1. Determine whether you need an isolator.

   If you connect Channel 0 to a modem or an ASCII device, consider installing an isolator between the controller and modem or ASCII device. An isolator is also recommended when connecting Channel 0 directly to a programming workstation.

   TIP If you connect to Channel 1 of the CompactLogix5330, an isolator is not needed.

   One possible isolator is the 1761-NET-AIC interface converter.



WARNING

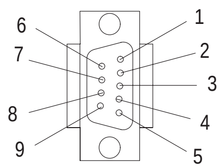
EXPLOSION HAZARD

The 1761-NET-AIC is rated Class I, Division 2. An external power supply must be used in hazardous locations, and the DC Power Source selector switch must be in the EXTERNAL position before connecting the power supply to the AIC+.

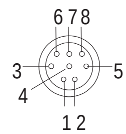
Refer to the Advanced Interface Converter (AIC+) User Manual, publication 1761-6.4 for installation requirements, especially if operating in a hazardous area.

2. Select the appropriate cable.

If you are using an isolator:	Use this cable:
yes	The 1761-CBL-AP00 cable (right-angle bend connector to controller) or the 1761-CBL-PM02 cable (straight connector to the controller) attaches the controller to port 2 on the 1761-NET-AIC isolator. The 8-pin mini-DIN connector is not commercially available, so you cannot make this cable.



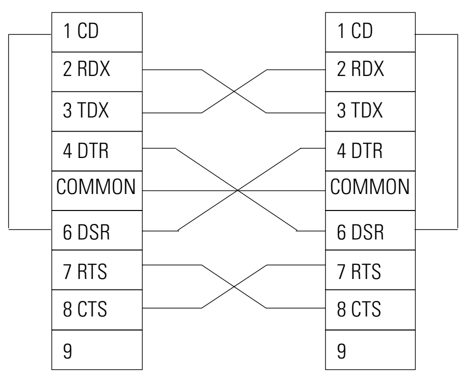
DB-9 right-angle or straight cable end



8-pin, mini-DIN cable end

Pin:	DB-9 end:	Mini-DIN end:
1	DCD	DCD
2	RxD	RxD
3	TxD	TxD
4	DTR	DTR
5	ground	ground
6	DSR	DSR
7	RTS	RTS
8	CTS	CTS
9	na	na

If you are using an isolator:	Use this cable:
no	The 1756-CP3 cable attaches the controller directly to the RS-232 device.



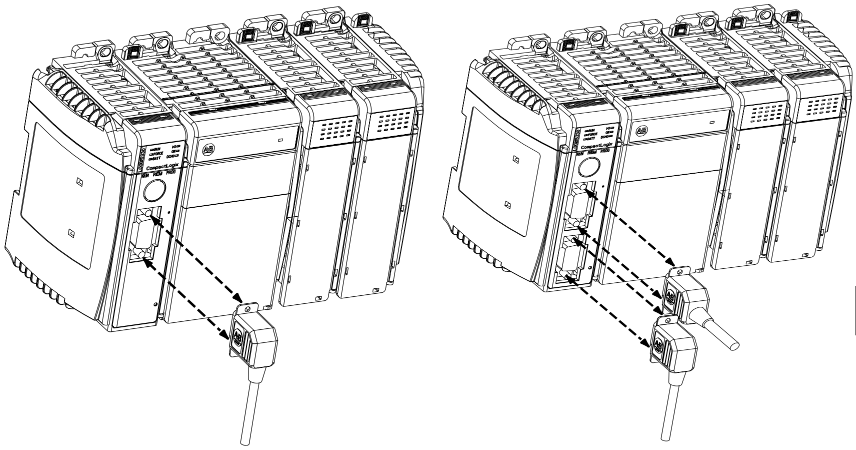
straight cable end

right-angle cable end

If you make your own cable, it must be shielded and the shield must be tied to the metal shell (that surrounds the pins) on the end of the cable.

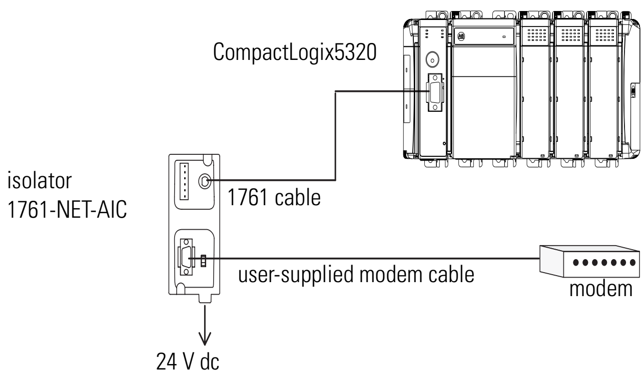
You can also use a 1747-CP3 cable from the SLC product family. This cable has a larger right-angle connector than the 1756-CP3 cable.

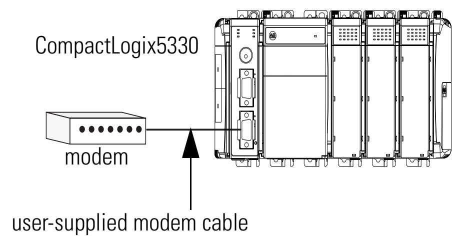
3. Connect the appropriate cable to the serial port(s) on the controller. To connect two cables to the CompactLogix5330, attach the straight-end connector to Channel 0.



TIP The Channel 0 port is locally grounded.

4. If necessary, attach the controller to the isolator.





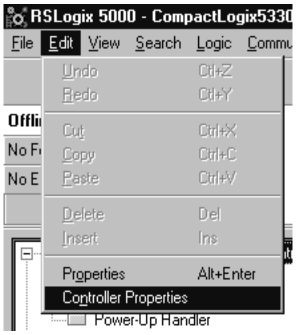
TIP Channel 1 on the CompactLogix5330 is isolated, so the 1761-NET-AIC is not required.

ATTENTION The CompactLogix controller is grounded through its DIN rail or mounting foot. It is important that you understand the workstation's grounding system before connecting it to the controller. An isolator is recommended between Channel 0 of the controller and the workstation.

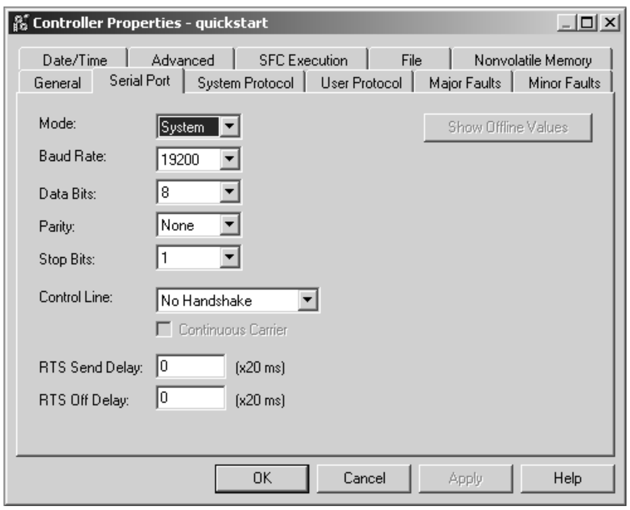
Step 2: Configure the Serial Port(s) of the Controller

1. In RSLogix 5000 programming software, select the Edit ⟶ Controller folder.

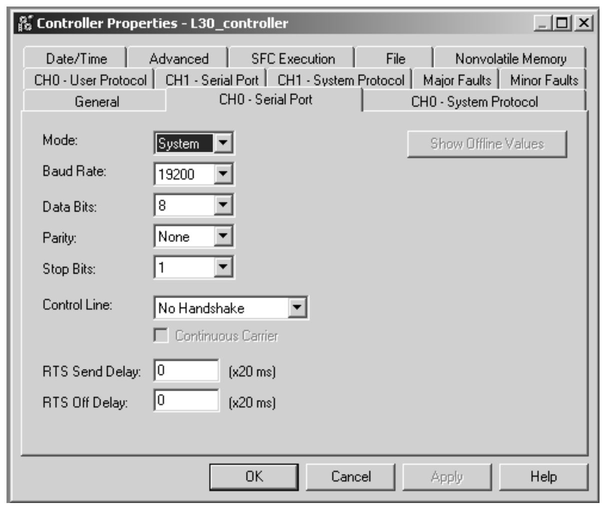
2. On the Serial Port tab, specify the appropriate serial communication configuration.



1769-L20 with one serial port.



1769-L30 with two serial ports.



3. On the System Protocol tab, select the appropriate DF1 communication mode for point-to-point or master/slave communications. Or on the User Protocol tab, select ASCII to communicate with an ASCII device.

Specifying Serial Port Characteristics

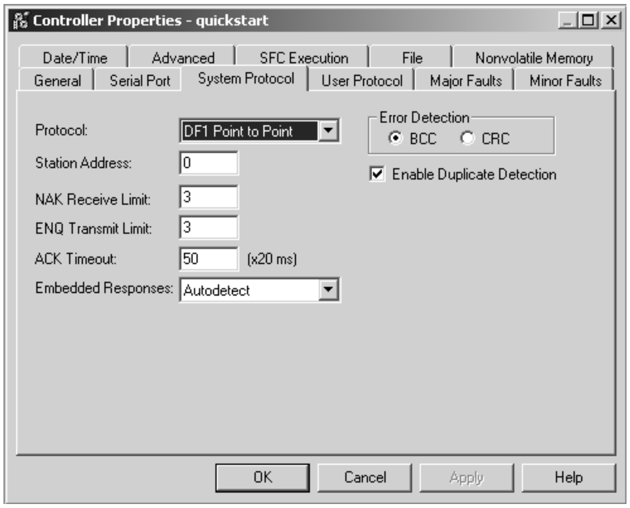
Specify these characteristics on the Serial Port tab(s) (default values are shown in bold):

Characteristic:	**Description (default is shown in bold):**
Mode	Select **System** (for DF1 and DH485 communication) or User mode (for ASCII communication). User mode is not available for Channel 1 at this time.
Baud rate	Specifies the communication rate for the serial port. Select a baud rate that all devices in your system support. Select 110, 300 600, 1200, 2400, 4800, 9600, **19200**, 38400 Kbps. Note: 38400 Kbps only in DF1 mode
Parity	Specifies the parity setting for the serial port. Parity provides additional message-packet error detection. Select **None** or Even.
Data bits	Specifies the number of bits per message packet. Select **8**.
Stop bits	Specifies the number of stop bits to the device with which the controller is communicating. Select **1** or 2.
Control line	Specifies the mode in which the serial driver operates. Select **No Handshake**, Full-Duplex, Half-Duplex with Continuous Carrier, or Half-Duplex without Continuous Carrier. If you are not using a modem, select No Handshake. If both modems in a point-to-point link are full-duplex, select Full-Duplex for both controllers. If the master modem is full-duplex and the slave modem is half-duplex, select Full-Duplex for the master controller and select Half-Duplex with Continuous Carrier for the slave controller. If all the modems in the system are half-duplex, select Half-Duplex without Continuous Carrier for the controller.
RTS send delay(1)	Enter a count that represents the number of 20 ms periods of time that elapse between the assertion of the RTS signal and the beginning of a message transmission. This time delay lets the modem prepare to transmit a message. The CTS signal must be high for the transmission to occur. The range is **0** to +32767 periods.
RTS off delay(1)	Enter a count that represents the number of 20 ms periods of time that elapse between the end of a message transmission and the de-assertion of the RTS signal. This time delay is a buffer to make sure the modem successfully transmits the entire message. The range is **0** to +32767 periods. Normally leave this setting at zero.

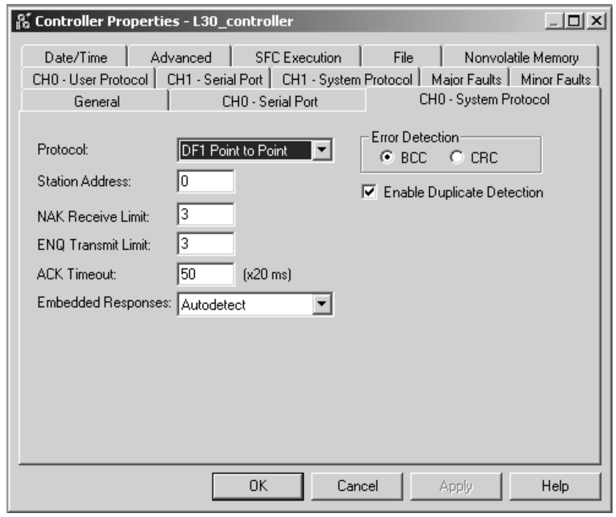
(1) This parameter is especially useful for communicating via radio modems.

Specifying System Protocol Characteristics

CompactLogix5320 with one serial port.



CompactLogix5330 with two serial ports.

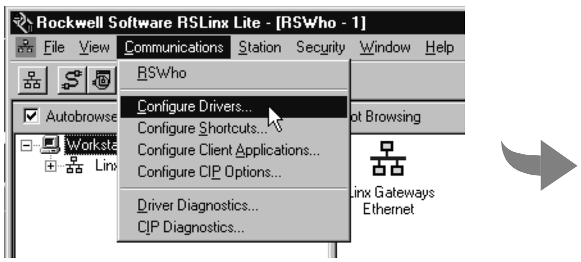


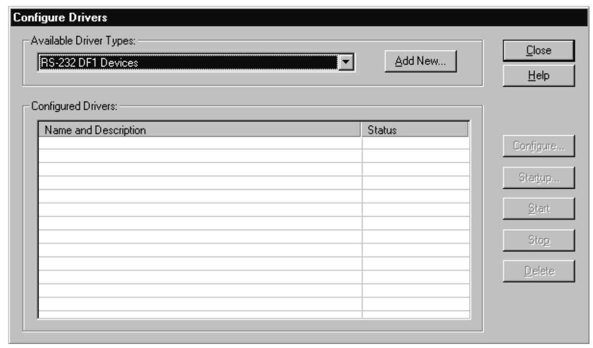
The available system modes are:

Use this mode:	For:	See page:
DF1 point-to-point	communication between the controller and one other DF1-protocol-compatible device. This is the default system mode. This mode is typically used to program the controller through its serial port.	4-11
DF1 master mode	control of polling and message transmission between the master and slave nodes. The master/slave network includes one controller configured as the master node and as many as 254 slave nodes. Link slave nodes using modems or line drivers. A master/slave network can have node numbers from 0 to 254. Each node must have a unique node address. Also, at least 2 nodes must exist to define your link as a network (1 master and 1 slave station are the two nodes).	4-14
DF1 slave mode	using a controller as a slave station in a master/slave serial communication network. When there are multiple slave stations on the network, link slave stations using modems or line drivers to the master. When you have a single slave station on the network, you do not need a modem to connect the slave station to the master. You can configure the control parameters for no handshaking. You can connect 2 to 255 nodes to a single link. In DF1 slave mode, a controller uses DF1 half-duplex protocol. One node is designated as the master and it controls who has access to the link. All the other nodes are slave stations and must wait for permission from the master before transmitting.	4-14
User mode (Channel 0 only)	communicating with ASCII devices. This requires your program logic to use the ASCII instructions to read and write data from and to an ASCII device.	4-16
DH-485	communicating with other DH-485 devices multi-master, token passing network allowing programming and peer-to-peer messaging.	5-1

Step 3: Configure the Serial Communication Driver

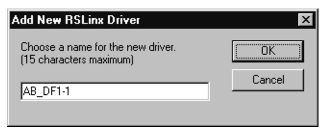
1. In RSLinx software, select Communication → Configure Driver. From the Available Driver Types list, select“RS-232 DF1 Devices“.



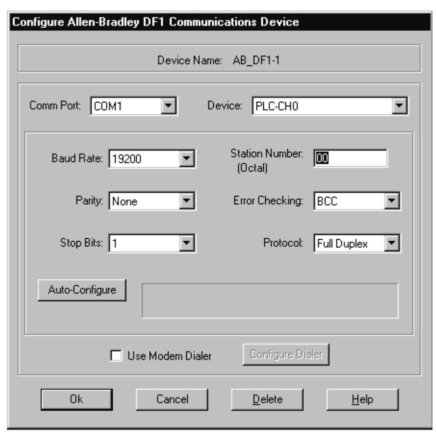


Click Add New.

2. Specify a name for the driver.



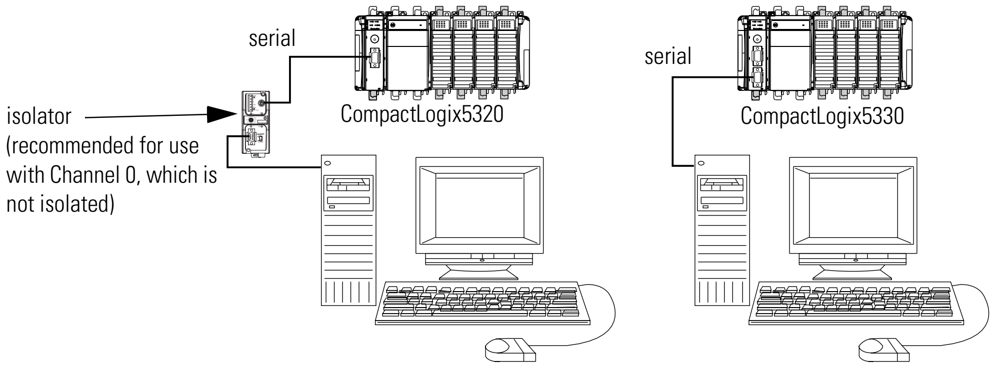
3. Specify the appropriate communication settings.
Select the “Logix5550/CompactLogix” and specify the COM port. Click Autoconfigure to have the software determine the remaining serial settings.



Click OK.

Example 1: Workstation Directly Connected to a CompactLogix Controller

In the following example, a workstation directly connects to a CompactLogix controller over a serial link. This is useful for downloading a controller project directly to the controller.



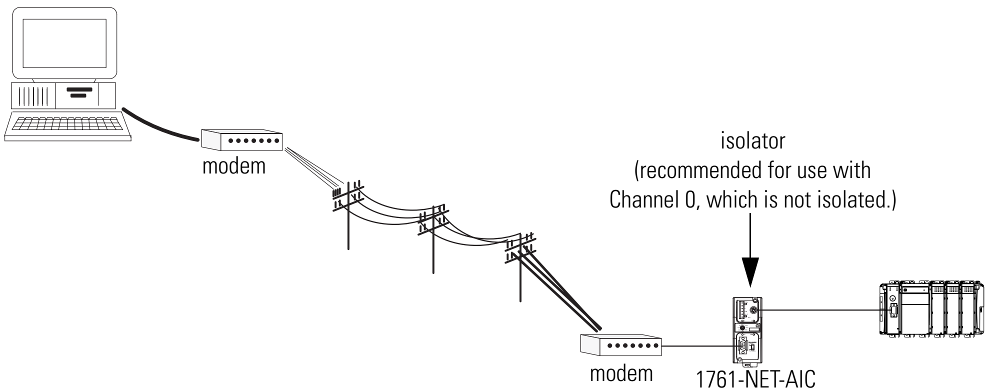
This type of protocol supports simultaneous transmission between two devices in both directions. The DF1 point-to-point protocol controls message flow, detects and signals errors, and retries if errors are detected.

Configuring a DF1 Point-to-Point Station

This field:	Description:
Station address	The station address for the serial port on the DF1 point-to-point network. Enter a valid DF1 address (0 to 254). Address 255 is reserved for broadcast messages. The default is 0.
NAK receive limit	Specifies the number of NAKs the controller can receive in response to a message transmission. Enter a value 0 to 127. The default is 3.
ENQ transmit limit	Specifies the number of inquiries (ENQs) you want the controller to send after an ACK timeout. Enter a value 0 to 127. The default is 3.
ACK timeout	Specifies the amount of time you want the controller to wait for an acknowledgment to its message transmission. Enter a value 0 to 32767. Limits are defined in 20 ms intervals. The default is 50 (1000 ms).
Embedded response	Specifies how to enable embedded responses. Select Autodetect (enabled only after receiving one embedded response) or Enabled. The default is Autodetect.
Error detection	Select BCC or CRC error detection. Configure both stations to use the same type of error checking. **BCC:** the controller sends and accepts messages that end with a BCC byte for error checking. BCC is quicker and easier to implement in a computer driver. This is the default. **CRC:** the controller sends and accepts messages with a 2-byte CRC for error checking. CRC is a more complete method.
Enable duplicate detection	Select whether or not the controller should detect duplicate messages. The default is duplicate detection enabled.

Example 2: Workstation Remotely Connected to a CompactLogix Controller

In the following example, a workstation remotely connects to a CompactLogix controller over a serial link. A modem is connected to the controller to provide remote access.



If you use a modem to remotely connect the controller to one workstation, use DF1 point-to-point (full-duplex) protocol, as in the previous example.

Master/Slave Communication Methods

Half-duplex DF1 Protocol

Half-duplex master/slave protocol is a SCADA protocol, consisting of 1 master and up to 254 slaves. Typically, the master polls all of the slaves for data in a round-robin fashion, using RF modems, leased-line modems, or any similar media.

A master station can communicate with a slave station in two ways:

Name:	This method:	Benefits:
standard communication mode	initiates polling packets to slave stations according to their position in the polling array(s). Polling packets are formed based on the contents of the normal poll array and the priority poll array.	This communication method is most often used for point-to-multipoint configurations. This method provides these capabilities: • slave stations can send messages to the master station (polled report-by-exception) • slave stations can send messages to each other via the master (slave-to-slave transfers) • master maintains an active station array The poll array resides in a user-designated data file. You can configure the master: • to send messages during its turn in the poll array *or* • for between-station polls (master transmits any message that it needs to send before polling the next slave station) In either case, configure the master to receive multiple messages or a single message per scan from each slave station.
message-based communication mode	initiates communication to slave stations using only user-programmed message (MSG) instructions. Each request for data from a slave station must be programmed via a MSG instruction. The master polls the slave station for a reply to the message after waiting a user-configured period of time. The waiting period gives the slave station time to formulate a reply and prepare the reply for transmission. After all of the messages in the master's message-out queue are transmitted, the slave-to-slave queue is checked for messages to send.	If your application uses satellite transmission or public switched-telephone-network transmission, consider choosing message-based communication. Communication to a slave station can be initiated on an as-needed basis. Also choose this method if you need to communicate with non-intelligent remote terminal units (RTUs).

Configuring a DF1 Slave Station

This field:	Description:
Station address	The station address for the serial port on the DF1 slave. Enter a valid DF1 address (0 to 254). Address 255 is reserved for broadcast messages. The default is 0.
Transmit retries	The number of times the remote station retries a message after the first attempt before the station declares the message undeliverable. Enter a value 0 to 127. The default is 3.
Slave poll timeout	Specifies the amount of time the slave station waits to be polled by a master before indicating a fault. Enter a value 0 to 32767. Limits are defined in 20 ms intervals. The default is 3000 (60,000 ms).
EOT suppression	Select whether or not to suppress sending EOT packets in response to a poll. The default is *not* to suppress sending EOT packets.
Error detection	Select BCC or CRC error detection. Configure both stations to use the same type of error checking. **BCC:** the controller sends and accepts messages that end with a BCC byte for error checking. BCC is quicker and easier to implement in a computer driver. This is the default. **CRC:** the controller sends and accepts messages with a 2-byte CRC for error checking. CRC is a more complete method.
Enable duplicate detection	Select whether or not the controller should detect duplicate messages. The default is duplicate detection enabled.

Configuring a DF1 Master Station

This field:	Description:
Station address	The station address for the serial port on the DF1 master. Enter a valid DF1 address (0 to 254). Address 255 is reserved for broadcast messages. The default is 0.
Transmit retries	Specifies the number of times a message is retried after the first attempt before being declared undeliverable. Enter a value 0 to 127. The default is 3.
ACK timeout	Specifies the amount of time you want the controller to wait for an acknowledgment to its message transmission. Enter a value 0 to 32767. Limits are defined in 20ms intervals. The default is 50 (1000 ms).
Reply message wait	**Message-based polling mode only** Specifies the amount of time the master station waits after receiving an ACK to a master-initiated message before polling the slave station for a reply. Enter a value 0 to 65535. Limits are defined in 20ms intervals. The default is 5 (100 ms).
Polling mode	Select one of these: • Message Based (slave cannot initiate messages) • Message Based (slave can initiate messages) - default • Standard (multiple message transfer per node scan) • Standard (single message transfer per node scan)
Master transmit	**Standard polling modes only** Select when the master station sends messages: • between station polls (default) • in polling sequence

This field:	Description:
Normal poll node tag	**Standard polling modes only** An integer tag array that contains the station addresses of the slave stations. Create a single-dimension array of data type INT that is large enough to hold all the normal station addresses. The minimum size is three elements. This tag must be controller-scoped. The format is: *list[0]* contains total number of stations to poll *list[1]* contains address of station currently being polled *list[2]* contains address of first slave station to poll *list[3]* contains address of second slave station to poll *list[n]* contains address of last slave station to poll
Normal poll group size	**Standard polling modes only** The number of stations the master station polls after polling all the stations in the priority poll array. Enter 0 (default) to poll the entire array.
Priority poll node tag	**Standard polling modes only** An integer tag array that contains the station addresses of the slave stations you need to poll more frequently. Create a single-dimension array of data type INT that is large enough to hold all the priority station addresses. The minimum size is three elements. This tag must be controller-scoped. The format is: *list[0]* contains total number of stations to be polled *list[1]* contains address of station currently being polled *list[2]* contains address of first slave station to poll *list[3]* contains address of second slave station to poll *list[n]* contains address of last slave station to poll
Active station tag	**Standard polling modes only** An array that stores a flag for each of the active stations on the DF1 link. Both the normal poll array and the priority poll array can have active and inactive stations. A station becomes inactive when it does not respond to the master's poll. Create a single-dimension array of data type SINT that has 32 elements (256 bits). This tag must be controller-scoped.
Error detection	Select BCC or CRC error detection. Configure both stations to use the same type of error checking. **BCC:** the controller sends and accepts messages that end with a BCC byte for error checking. BCC is quicker and easier to implement in a computer driver. This is the default. **CRC:** the controller sends and accepts messages with a 2-byte CRC for error checking. CRC is a more complete method.
Enable duplicate detection	Select whether or not the controller should detect duplicate messages. The default is duplicate detection enabled.

If You Choose One of the Standard Polling Modes

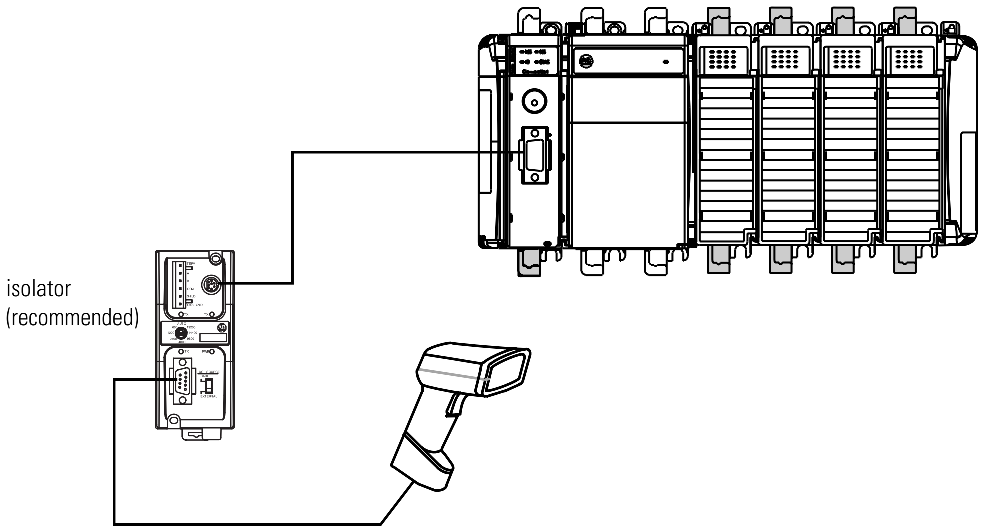
The master station polls the slave stations in this order:

1. all stations that are active in the priority poll array
2. one station that is inactive in the priority poll array
3. the specified number (normal poll group size) of active stations in the normal poll array
4. one inactive station, after all the active stations in the normal poll array have been polled

Use the programming software to change the display style of the active station array to binary so you can view which stations are active.

Example 3: CompactLogix Controller Connected to a Bar Code Reader

In the following example, a workstation connects to a bar code reader. A bar code reader is an ASCII device, so you configure the serial port differently than in the previous examples. Configure the serial port for User mode, rather than the system mode.



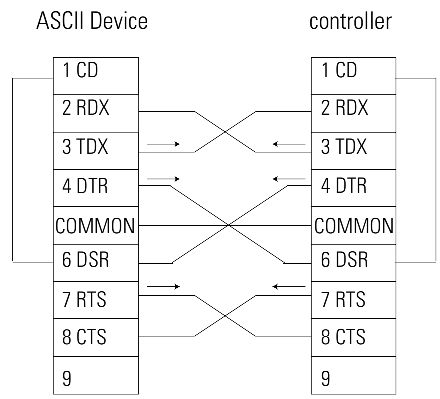
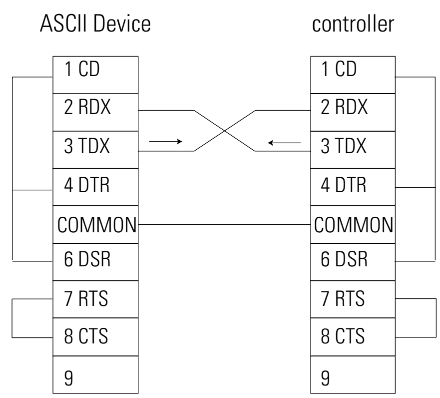
IMPORTANT You must use Channel 0 when connecting to an ASCII device. Channel 1 on the CompactLogix5330 does not support ASCII at this time.

Connect the ASCII Device to the Controller

To connect the ASCII device to the Channel 0 serial port of the controller:

1. For the serial port of the ASCII device, determine which pins send signals and which pins receive signals.

2. Connect the sending pins to the corresponding receiving pins and attach jumpers:

If the communications hardware handshaking is:	Then wire the connectors as follows:
enabled	
disabled	

3. Attach the cable shield to both connectors and tie the cable to both connectors.

4. Connect the cable to the controller and the ASCII device.

The following table lists the default serial port configuration settings for the ASCII protocol. You specify these settings on the User Protocol tab under Controller Properties.

Configuring User Mode

This field:	Description:
Buffer size	Specify the maximum size (in bytes) of the data array you plan to send and receive. The default is 82 bytes.
Termination characters	Specify the characters you will use to designate the end of a line. The default characters are '$r' and '$FF'.
Append characters	Specify the characters you will append to the end of a line. The default characters are '$r' and '$l'.(1)
XON/XOFF	Select whether or not to regulate the flow of incoming data. The default is disabled.
Echo mode	Select whether or not to echo data back to the device from which it was sent. The default is disabled.
Delete mode	Select Ignore, CTR, or Printer for the delete mode. The default is Ignore.

(1) IEC 1131-3 representation for carriage return and line feed.

Programming ASCII Instructions

Both of the CompactLogix controllers support ASCII instructions on Channel 0. However, at this time, the CompactLogix5330 controller *does not* support ASCII instructions on Channel 1. ASCII instructions are used to communicate with ASCII devices. Your RSLogix5000 programming software CDROM includes programming examples using ASCII instructions.

For information about using these examples, see the *Logix5000 Controllers General Instruction Set Reference Manual*, publication 1756-RM003.

Communicating with Devices on a DH-485 Link

Using This Chapter

The DH-485 protocol uses RS-485 half-duplex as its physical interface. (RS-485 is a definition of electrical characteristics; it is *not* a protocol.) You can configure the RS-232 port of the CompactLogix controller to act as a DH-485 interface. By using a 1761-NET-AIC and the appropriate RS232 cable (1756-CP3 or 1747-CP3), a CompactLogix controller can send and receive data on a DH-485 network.

For information about:	See page
Configuring your system for a DH-485 link	5-2
Planning a DH-485 network	5-5
Installing a DH-485 network	5-7
Example: Messaging on DH-485 with SLC 5/03, CompactLogix Controllers	5-9

IMPORTANT A DH-485 network consists of multiple cable segments. Limit the total length of all the segments to 1219m (4000 ft.).

Configuring Your System for a DH-485 Link

For the CompactLogix controller to operate on a DH-485 network, you need:

- a 1761-NET-AIC interface converter for each CompactLogix controller you want to put on the DH-485 network.

  You can have two controllers per one 1761-NET-AIC converter, but you need a different cable for each controller. Connect one controller to port 1 (9-pin connector) and one controller to port 2 (mini-DIN connector).

- RSLogix 5000 programming software to configure the serial port of the controller for DH-485 communications.

Excessive traffic on the DH-485 network may make it impractical to connect to your CompactLogix controller with RSLogix 5000 programming software. Program upload/download, monitoring and online editing of programs via DH-485 can be accomplished when the system is not running and the controllers are in Program mode.

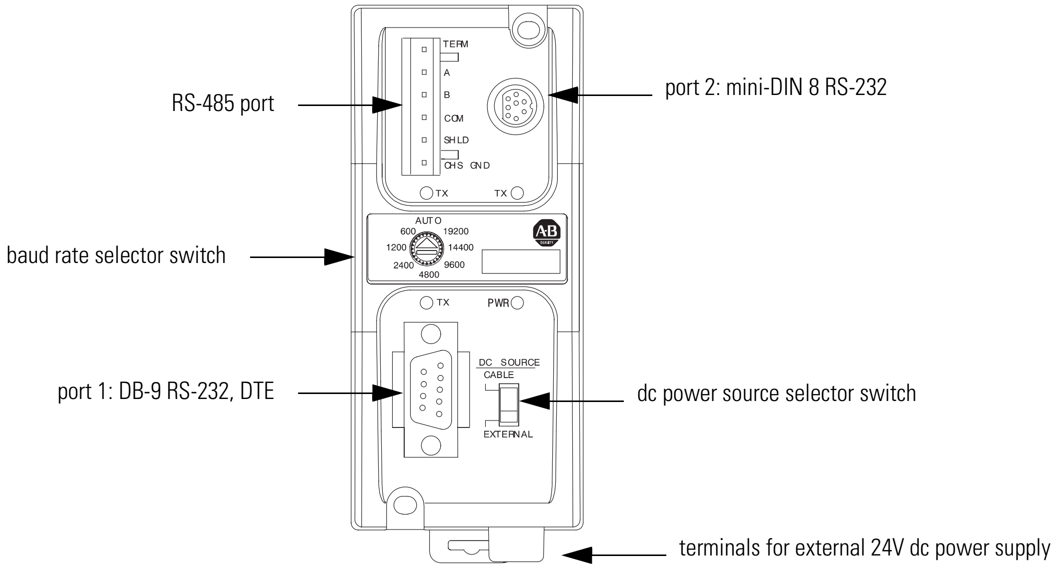
In addition, when attempting to go online or upload/download a program using the Communications/Who Active window in RSLogix 5000 software, disable the Autobrowse feature to minimize traffic from RSLogix 5000 software on the DH-485 network.

The DH-485 network is not recommended for new applications using CompactLogix controllers. CompactLogix controllers should be used on DH-485 networks only when adding these controllers to an existing DH-485 network. For new applications with CompactLogix controllers, DeviceNet and Ethernet are the recommended networks.

Step 1: Configure the Hardware

The channel 0 RS-232 port is a non-isolated serial port built-in to the front of the CompactLogix controller. Channel 1 on CompactLogix5330 is an isolated RS-232 port. The RS-232 port(s) support the requirements you need for the DH-485 network connection.

Connect the controller to an RS-232-to-RS-485 isolator. One possible isolator is the 1761-NET-AIC interface converter.



Connect the serial port of the CompactLogix controller to either port 1 or port 2 of the 1761-NET-AIC converter. Use the RS-485 port to connect the converter to the DH-485 network.

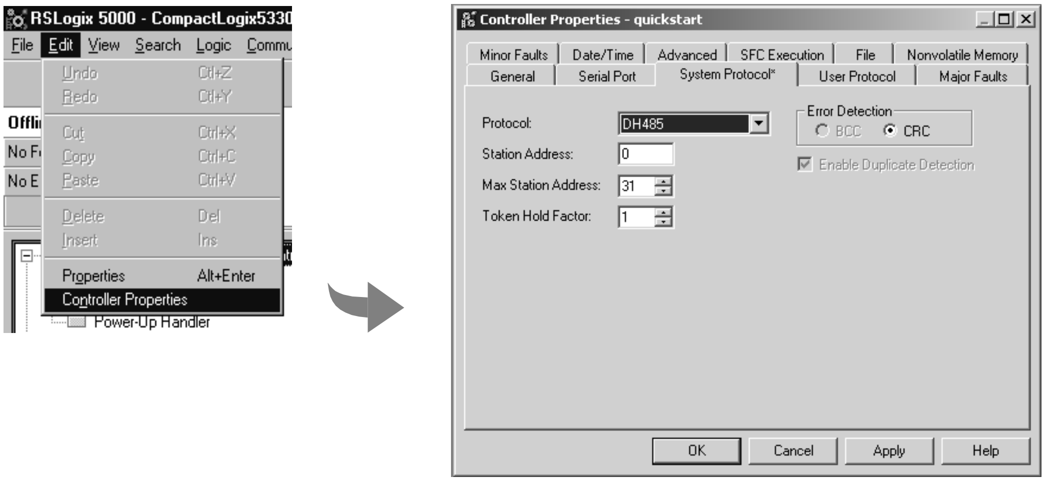
The cable you use to connect the controller depends on the port you use on the 1761-NET-AIC converter.

If you connect to this port:	Use this cable:
port 1 DB-9 RS-232, DTE connection	1747-CP3 or 1761-CBL-AC00
port 2 mini-DIN 8 RS-232 connection	1761-CBL-AP00 or 1761-CBL-PM02

Step 2: Configure the DH-485 Port of the Controller

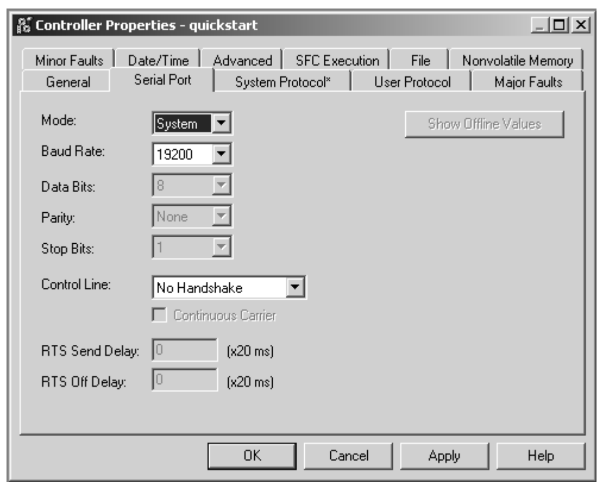
1. In RSLogix 5000 programming software, select the Controller folder. Right-click to select Properties.

2. On the System Protocol tab, specify the appropriate serial communication configuration.



3. On the Serial Port tab, specify the appropriate communication settings.

The grayed out settings are selections that do not apply to a DH-485 network.



Specify these characteristics on the Serial Port tab (default values are shown in bold):

Characteristic:	Description (default is shown in bold):
Baud Rate	Specifies the communication rate for the DH-485 port. All devices on the same DH-485 network must be configured for the same baud rate. Select 9600 or **19200** Kbps.
Node Address	Specifies the node address of the CompactLogix controller on the DH-485 network. Select a number **1**-31 decimal, inclusive. To optimize network performance, assign node addresses in sequential order. Initiators, such as personal computers, should be assigned the lowest address numbers to minimize the time required to initialize the network.
Token Hold Factor	Specifies the number of messages sent per token possession. Select a number **1**-4, inclusive.
Maximum Node Address	Specifies the maximum node address of all the devices on the DH-485 network. Select a number 1-**31** decimal, inclusive. To optimize network performance, make sure: • the maximum node address is the highest node number being used on the network • that all the devices on the same DH-485 network have the same selection for the maximum node address.

Planning a DH-485 Network

The DH-485 network offers:

- interconnection of 32 devices
- multi-master capability
- token passing access control
- the ability to add or remove nodes without disrupting the network
- maximum network length of 1219 m (4000 ft.)

The DH-485 protocol supports two classes of devices: initiators and responders. All initiators on the network get a chance to initiate message transfers. The DH-485 protocol uses a token-pass algorithm to determine which initiator has the right to transmit.

DH-485 Token Rotation

A node holding the token can send any valid packet onto the network. As a default, each node gets only one transmission (plus two retries) each time it receives the token. After a node sends one message packet, it attempts to give the token to its successor by sending a “token pass” packet to its successor.

If no network activity occurs, the initiator sends the token pass packet again. After two retries (a total of three tries) the initiator attempts to find a new successor.

IMPORTANT	The maximum address that the initiator searches for before starting again with zero is the value in the configurable parameter “maximum node address.” The default and maximum value for this parameter is 31 for all initiators and responders.

The allowable range of the node address of a initiator is 0 to 31. The allowable address range for all responders is 1 to 31. There must be at least one initiator on the network.

Network Initialization

The network requires at least one initiator to initialize it. Network initialization begins when a initiator on the network detects a period of inactivity that exceeds the time of a link dead timeout. When the link dead timeout is exceeded, usually the initiator with the lowest address claims the token. When a initiator has the token it will begin to build the network.

Building a network begins when the initiator that claimed the token tries to pass the token to the successor node. If the attempt to pass the token fails, or if the initiator has no established successor (for example, when it powers up), it begins a linear search for a successor starting with the node above it in the addressing.

When the initiator finds another active node, it passes the token to that node, which repeats the process until the token is passed all the way around the network to the initial node. At this point, the network is in a state of normal operation.

Number of Nodes and Node Addresses

The number of nodes on the network directly affects the data transfer time between nodes. Unnecessary nodes (such as a second programming terminal that is not being used) slow the data transfer rate. The maximum number of nodes on the network is 32.

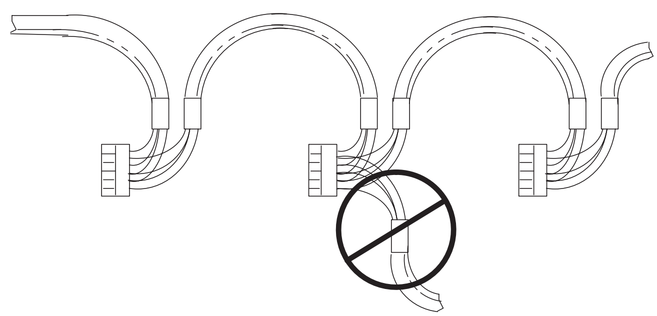
If the node addresses for controllers are assigned in sequence, starting at node 1 (with node 0 left for a programming terminal), it is as efficient to leave the maximum node address at 31 as it is to decrease it to the highest node address on the network. Then, adding devices to the network at a later time will not require modifying the maximum node address in every device on the network. The maximum node address should be the same for all devices on a DH-485 network for optimal operation.

The best network performance occurs when node addresses start at 0 and are assigned in sequential order. The controller defaults to node address 1. Initiators, such as personal computers, should be assigned the lowest numbered addresses to minimize the time required to initialize the network.

Installing a DH-485 Network

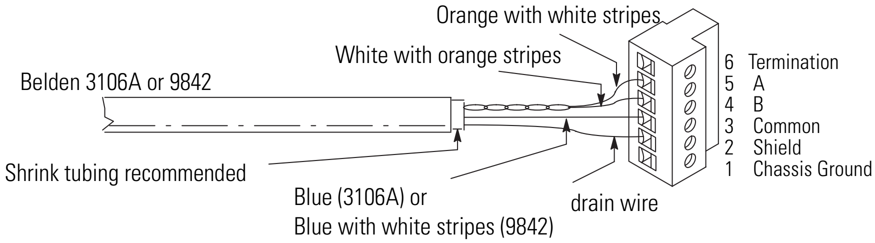
A DH-485 network consists of a number of cable segments daisy-chained together. The total length of the cable segments cannot exceed 1219 m (4000 ft).

IMPORTANT Use shielded, twisted-pair cable, either Belden 3106A or Belden 9842. A daisy-chained network is recommended.

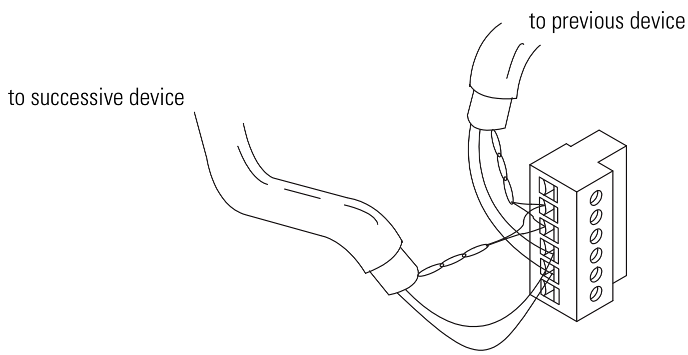


When cutting cable segments, make them long enough to route them from one link coupler to the next with sufficient slack to prevent strain on the connector. Allow enough extra cable to prevent chafing and kinking in the cable.

Single Cable Connection



Multiple Cable Connection



The table below shows wire/terminal connections for Belden 3106A.

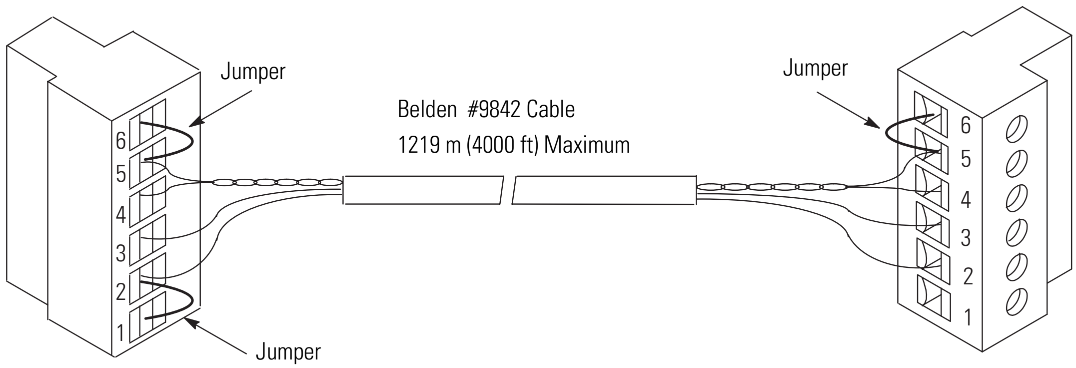
For this Wire/Pair	Connect this Wire	To this Terminal
shield/drain	non-jacketed	2 - Shield
blue	blue	3 - (Common)
white/orange	white with orange stripe	4 - (Data B)
	orange with white stripe	5 - (Data A)

The table below shows wire/terminal connections for Belden 9842.

For this Wire/Pair	Connect this Wire	To this Terminal
shield/drain	non-jacketed	2 - Shield
blue/white	white with blue stripe	cut back - no connection(1)
	blue with white stripe	3 - (Common)
white/orange	white with orange stripe	4 - (Data B)
	orange with white stripe	5 - (Data A)

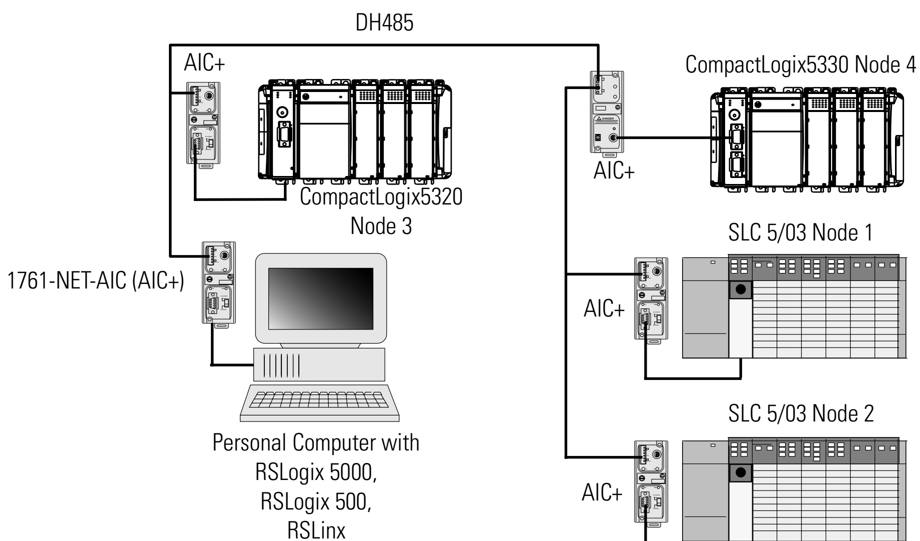
(1) To prevent confusion when installing the communication cable, cut back the white with blue stripe wire immediately after the insulation jacket is removed. This wire is not used by DH-485.

Grounding and Terminating a DH-485 Network



Example: Messaging on DH-485 with SLC 5/03, CompactLogix Controllers

This example shows how to expand an existing SLC 5/03 control system with two CompactLogix controllers. The SLC 5/03 controllers are connected together on a DH485 network, for messaging between them and for program upload/download and program monitoring with RSLogix 500 software. The CompactLogix controllers also connect to the existing DH485 network for messaging between them, as well as for messaging to the SLC controllers and for program upload/download and monitoring with RSLogix 5000 software.



The 1761-NET-AIC (AIC+) at the computer can be eliminated if the computer is near enough to any one of the controller AIC+ units to connect with a 1747-CP3 cable.

The SLC 5/03 controllers are sending a message to each other, each reading 50 words from the other. The 1769-L20 controller will write 50 words of data to the 1769-L30 controller and will read 50 words of data from SLC 5/03, node 1. The CompactLogix5330 controller will write 50 words of data to the CompactLogix5320 controller as well as read 50 words of data from SLC 5/03, node 2.

Each CompactLogix controller sends one message at a time, to keep the traffic on the DH-485 network to a minimum. This allows the programming software to access the controllers more readily while the system is running.

When online with one of the CompactLogix controllers with RSLogix 5000 software, message throughput between controllers on the DH-485 network decreases. If your system cannot tolerate decreased message throughput, you can:

- do online edits and uploading/downloading when the system is not running and the controllers are in Program mode.
- use a 1769-L30 controller, which has two serial ports, so you can access the controller through one serial port while the other is dedicated to DH-485 communications.

All messages for this example use SLC 500 Typed Read and Write commands.

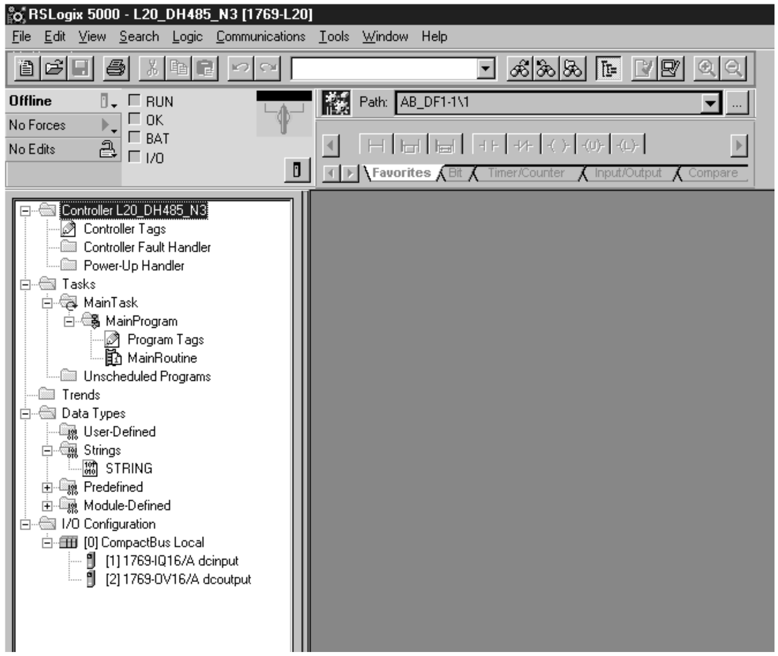
Configuring and Programming the 1769-L20 Controller

The ladder programs for the two CompactLogix controllers contain two MSG rungs. Also, each controller's channel 0 serial port is configured for DH-485 communications.

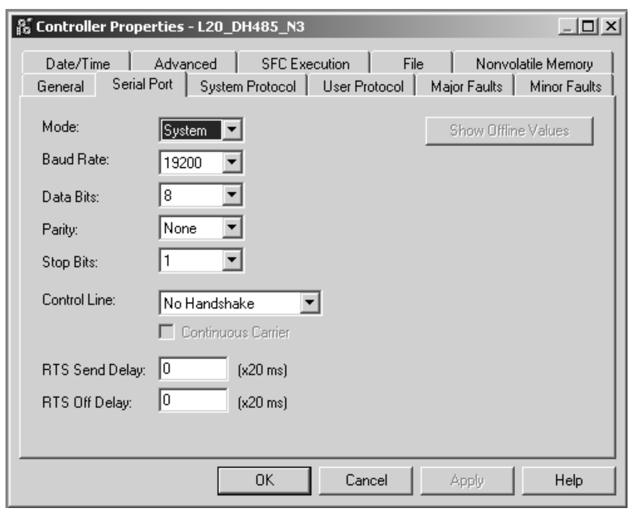
The messages sent and received by the 1769-L20 controller are:

- Send a MSG to read 50 integer words from SLC 5/03 controller, node 1.
- Send a MSG to write 50 integer words to the 1769-L30 controller, node 4.
- Receive a MSG write of 50 integer words, from the 1769-L30 controller, to file 11. File 11 will be mapped to tag "data_from_L30_N4" in the 1769-L20 controller.

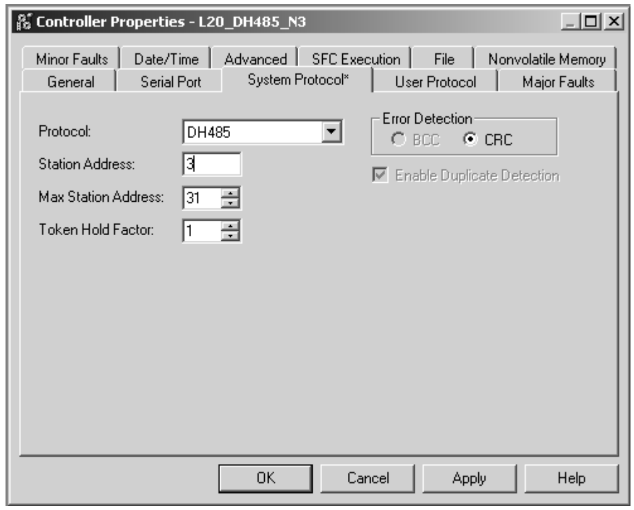
1. Create an RSLogix 5000 project named "L20_DH485_N3".



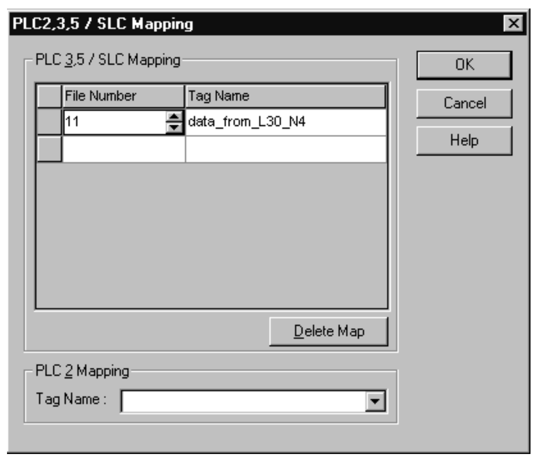
2. Right click on the Controller and select Properties. In the Properties screen, click the Serial Port tab. The Serial Port configuration should be as follows:



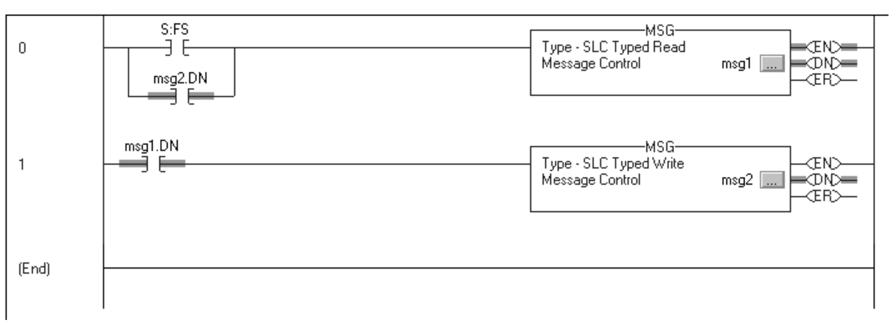
3. Click on the System Protocol tab. Select DH485 and a Station Address of 3. The System Protocol screen should be as follows:



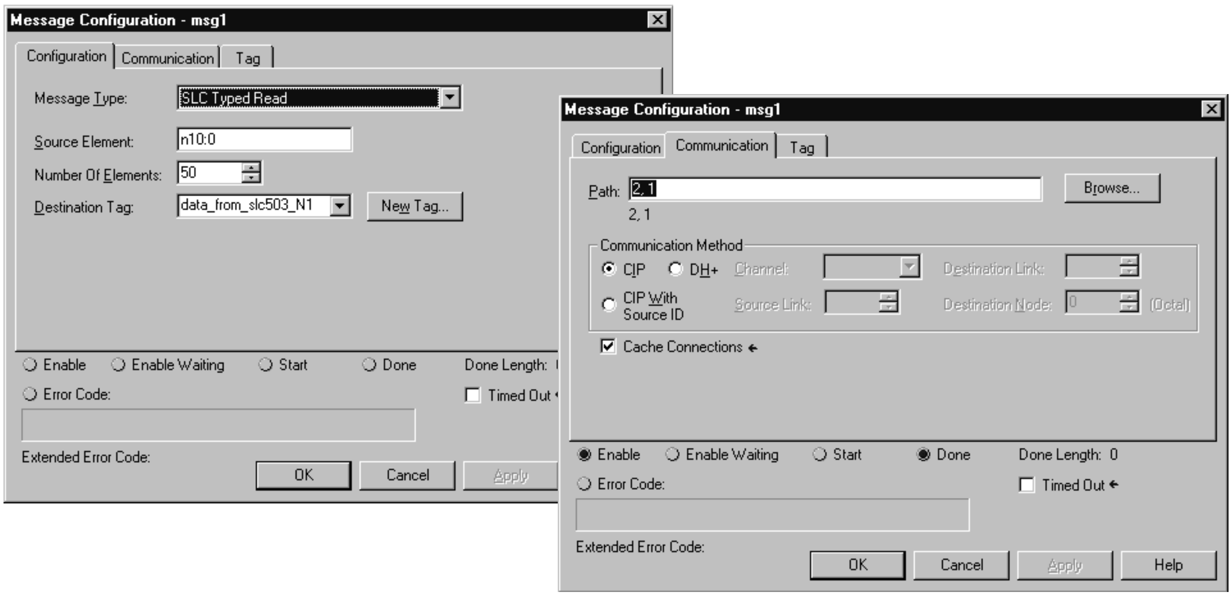
4. Create a tag named “data_from_L30_N4”. Then, map file 11 to that tag. File 11 is the address used in the SLC 500 message from the 1769-L30 controller.



5. Next, enter the MSG instructions:



The two messages toggle, allowing one message to be sent at a time. When it completes, the other message is triggered and so on. Configure the MSG instructions as follows:

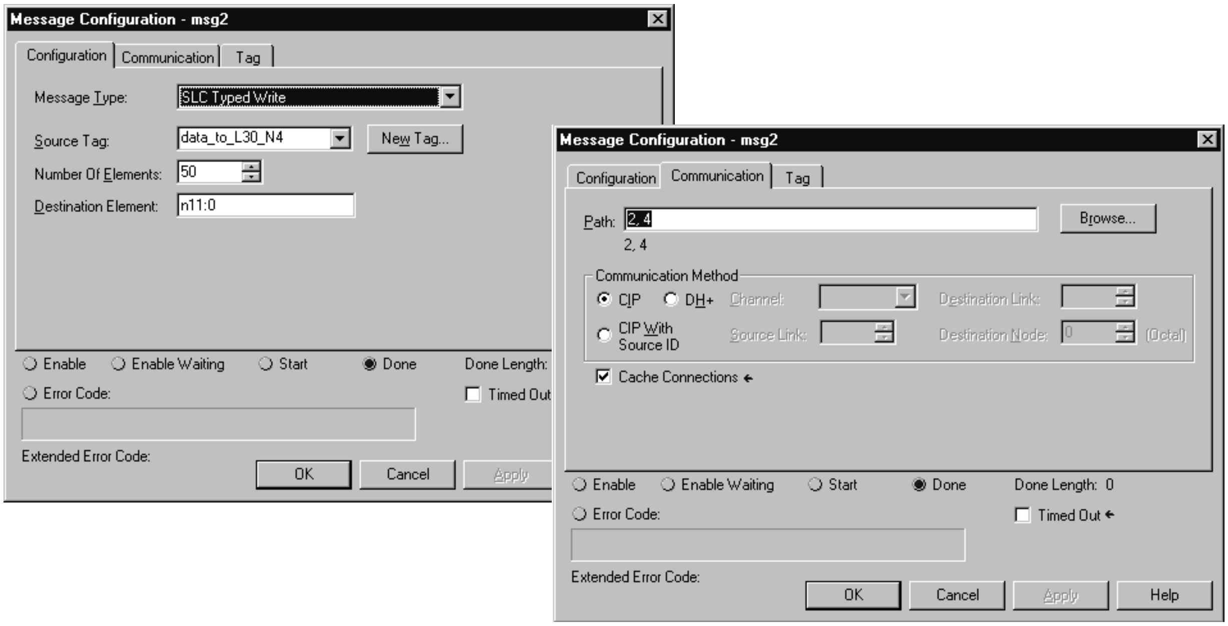


There are 50 words of data read from integer file N10:0 in the SLC 5/03 controller and stored in tag “data_from_slc503_N1”. The path on the Communication tab is “2,1”. Channel 0 is port 2 from the controller’s perspective and the SLC 5/03 controller the MSG is being sent to is node 1.

TIP

The 2 in the path directs the MSG to channel 0 of the CompactLogix controller. Use 3 for channel 1 of the 1769-L30 controller.

The second MSG instruction writes 50 integer words of data to the 1769-L30 controller at node 4. Its two MSG Instruction screens are as follows:



An SLC Typed Write message type writes 50 words of data to the 1769-L30 controller. The Destination Element of N11:0 is mapped to a tag in the 1769-L30 controller, as in the 1769-L20 controller. The data sent to the 1769-L30 controller is from tag "Data_to_L30_N4" in the 1769-L20 controller. The path on the Communication tab is "2,4", where the 2 represents channel 0 and the 4 is the DH-485 node address of the 1769-L30 controller.

6. Save the program and download it to the 1769-L20 controller using the default protocol on this controller of full duplex DF1.

   In RSLinx, create a full duplex DF1 driver and use the Auto Config. feature to establish communications. Download the program from RSLogix 5000 software. When you are prompted to apply the serial configuration changes, click Yes. The software displays a communication error, and you will be offline. The serial port is configured for DH-485 and node address 3. Now connect the 1769-L20 controller's serial port to a 1761-NET-AIC, which in turn can be connected to the DH-485 network with the SLC 5/03 controllers.

Configuring and Programming the 1769-L30 Controller

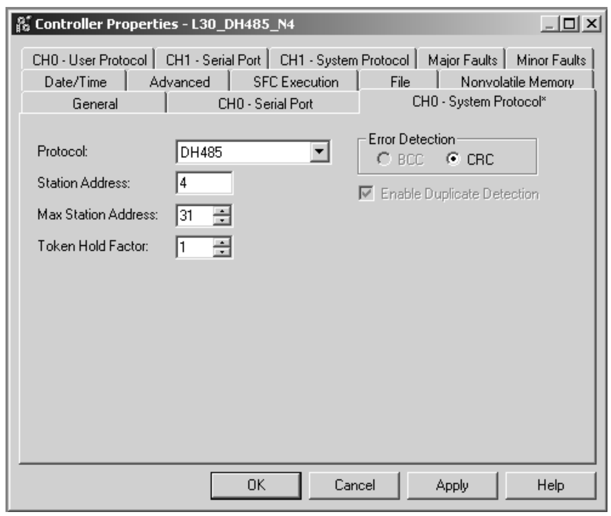
Configure the 1769-L30 controller for the DH-485 network using the same steps as for the 1769-L20 controller.

The messages sent and received by the 1769-L30 controller are:

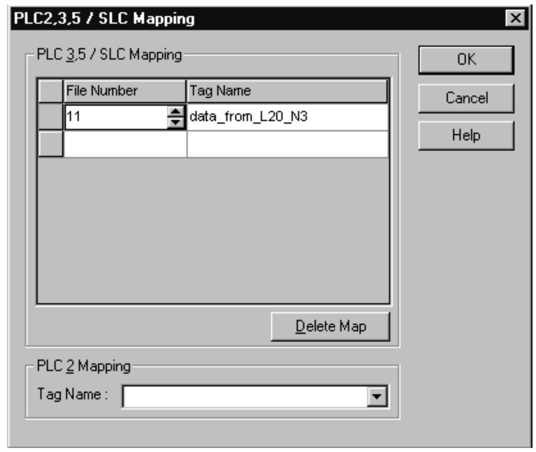
- Send a MSG to read 50 integer words from SLC 5/03 controller, node 2.
- Send a MSG to write 50 integer words to the CompactLogix5320 controller, node 3.
- Receive a MSG to write 50 integer words, from the CompactLogix5320 controller, to file 11. File 11 will be mapped to tag “Data_From_L20_N3” in the CompactLogix5330 controller.

1. Create an RSLogix 5000 project named “L30_DH485_N3”.

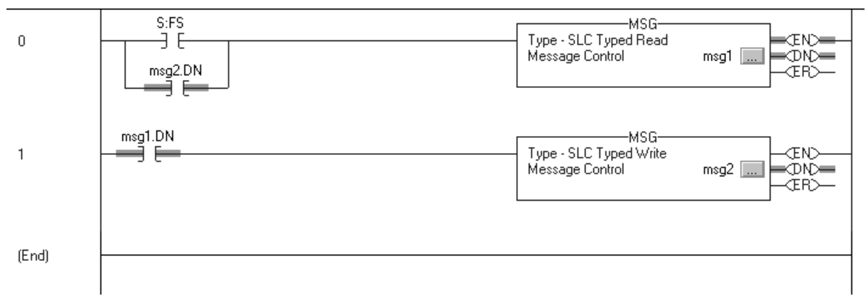
2. The channel 0 System Protocol tab should be as follows:



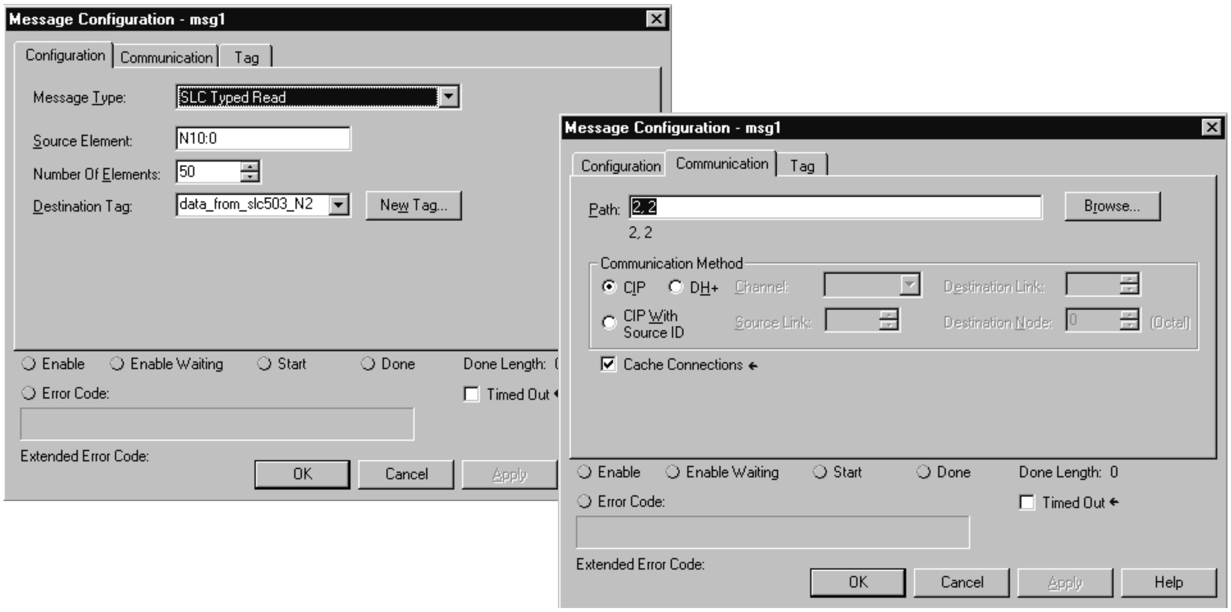
3. Create a tag named “data_from_L20_N3”. Then, map file 11 to that tag. File 11 is the address used in the SLC 500 message from the 1769-L20 controller.

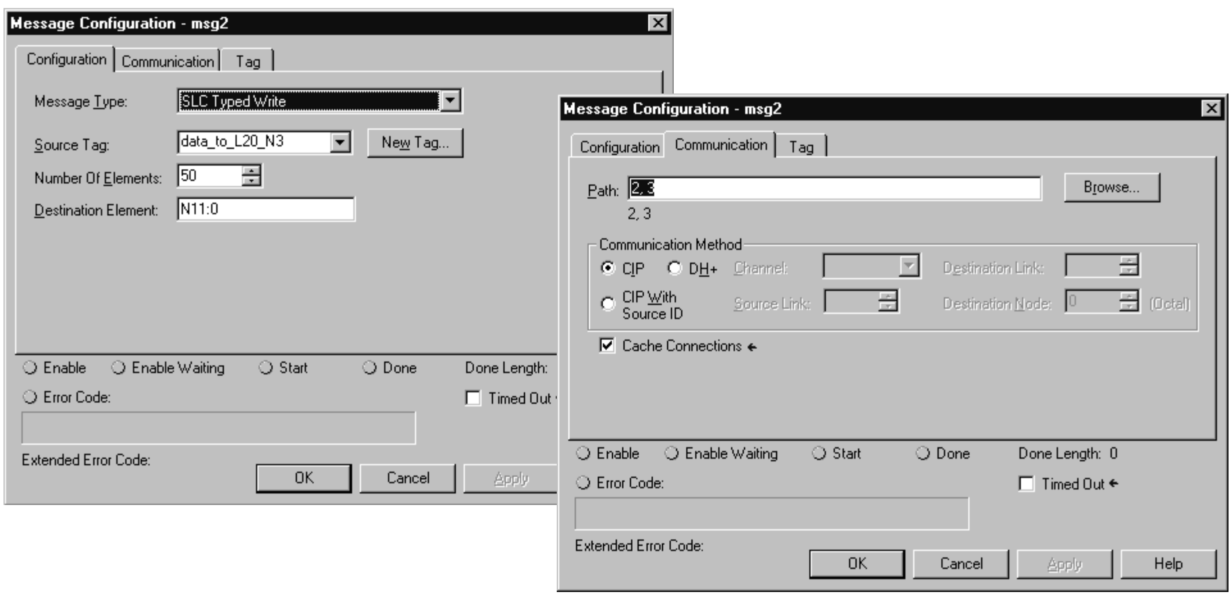


4. Next, enter the MSG instructions:



The MSG instructions in the 1769-L30 controller should be as follows:





5. Save the program and download it to the 1769-L30 controller using the default protocol on this controller of full duplex DF1.

   In RSLinx, create a full duplex DF1 driver and use the Auto Config. feature to establish communications. Download the program from RSLogix 5000 software. If you download to channel 0, you will be prompted to apply the serial configuration changes. Click Yes. The software displays a communication error, and you will be offline. The channel 0 serial port will be configured for DH-485 and node address 4. You can now connect the 1769-L30 controller's channel 0 serial port to a 1761-NET-AIC, which in turn may be connected to the DH-485 network with the SLC 5/03 and 1769-L20 controllers.

TIP

If the combination of node count and network traffic on DH-485 makes it difficult or impractical to connect to your CompactLogix controllers with RSLogix 5000 while the system is running, it may be prudent to do program maintenance while the system is not running. Program upload/download, monitoring and online editing of programs on DH-485 may be more readily accomplished when the system is not running, i.e. the controllers are in Program mode.

In addition, when attempting to go online or upload/download a program using the Communications/Who Active window in RSLogix 5000, disable the Autobrowse feature to minimize traffic from RSLogix 5000 on the DH-485 network.

Other alternatives are to switch to different networks:

- DeviceNet network using 1761-NET-DNI modules in place of the 1761-NET-AIC modules.
- EtherNet/IP network using 1761-NET-ENI modules in place of the 17610NET-AIC modules.

Communicating with Devices on a DeviceNet link

Using This Chapter

For information about:	See page
Configuring your system for a DeviceNet link	6-1
Example 1: Controlling DeviceNet devices	6-2
Example 2: Sending messages over DeviceNet using a 1761-NET-DNI interface converter	6-13

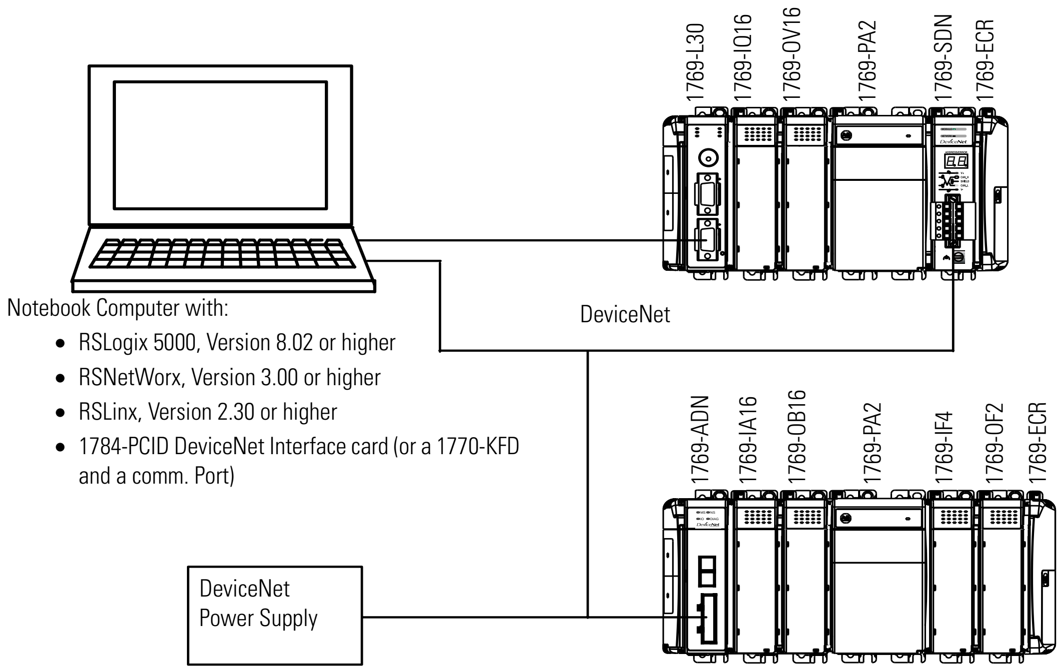
Configuring Your System for a DeviceNet Link

Select the appropriate DeviceNet interface depending on the application and how the controller interacts with the devices:

If your application:	Select this interface:	Description:
• communicates with other DeviceNet devices (I/O data transfer only) • uses the controller as a master or slave on DeviceNet • uses the controller serial port for other communications	1769-SDN DeviceNet scanner module	The scanner acts as an interface between DeviceNet devices and the CompactLogix controller. The scanner lets the controller: • read inputs from slave devices • write outputs to slave devices
• accesses remote Compact I/O over a DeviceNet network • sends remote I/O data for as many as 30 modules back to scanner or controller	1769-ADN DeviceNet adapter module	The adapter: • interfaces with as many as 30 Compact I/O modules • communicates to other network system components (typically a controller or scanner and/or programming terminals) over the DeviceNet network
• communicates with other DeviceNet devices (messaging only) • uses the controller as a slave on DeviceNet • does not use the controller serial port for other communications	1761-NET-DNI interface	The interface links the CompactLogix controller to other devices on a DeviceNet network to: • download configuration data to a device • monitor operational status of a device • communicate with peer devices (messaging) • upload/download programs

Example 1: Controlling DeviceNet Devices

This example uses a 1769-SDN scanner module in the local CompactLogix system to control the I/O attached to a 1769-ADN adapter module.



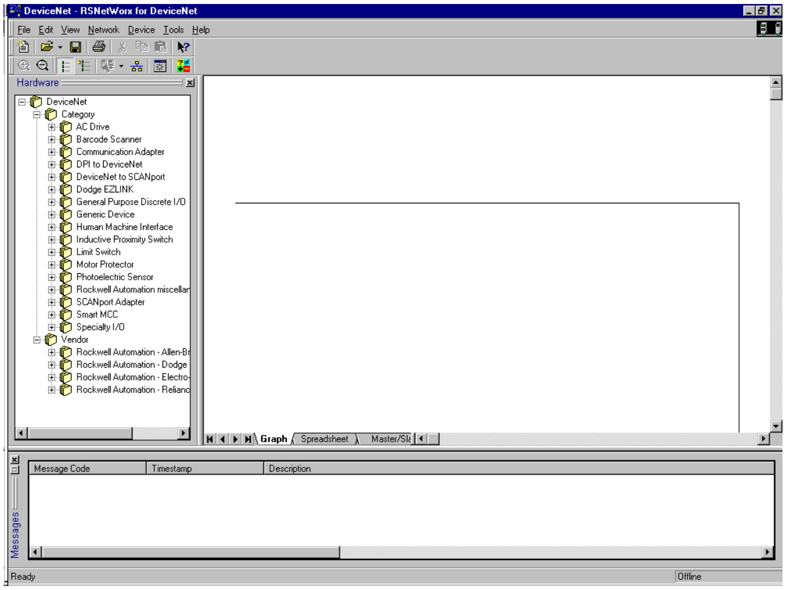
This example describes:

- using RSNetWorx for DeviceNet to assign node addresses to the 1769-SDN and the 1769-ADN and map the adapter's image into the scanner
- creating a CompactLogix project including the necessary configuration for the 1769-SDN DeviceNet scanner module
- controlling outputs and reading inputs with the distributed I/O via DeviceNet

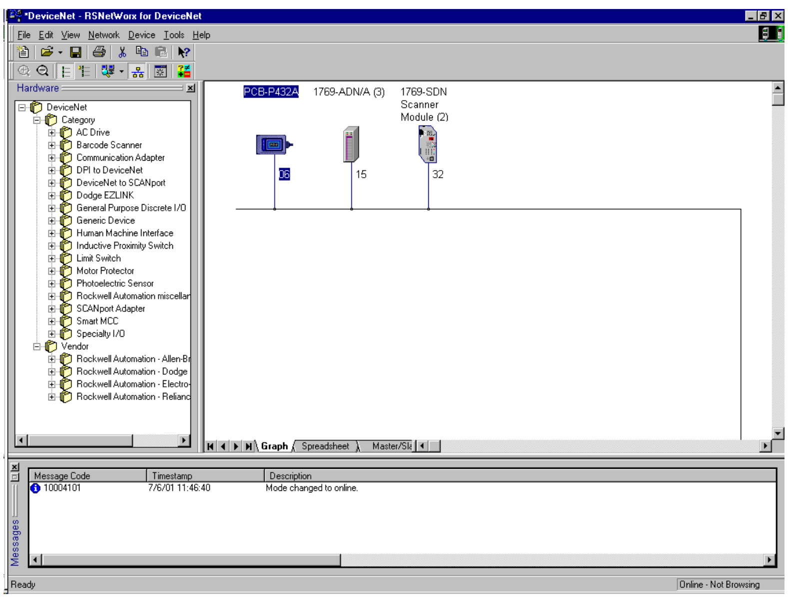
Step 1: Configuring the 1769-ADN Adapter

1. Start RSNetWorx.

2. Select Network → Online. The RSLinx communication driver screen appears. Choose the 1784-PCD-1, DeviceNet driver, or if you are using a 1770-KFD, choose its driver. This example assumes that one of these drivers is already configured in RSLinx.

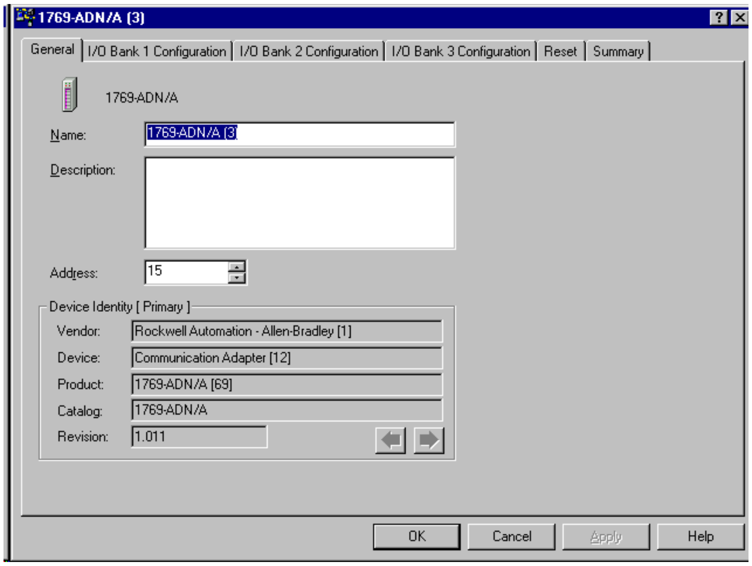


3. The software then prompts you to either upload or download. Choose upload. RSNetWorx browses the network for valid devices. The online screen should look like the following, where the 1784-PCID card (computer) is node 6, the 1769-ADN is node 15, and the 1769-SDN is node 32 for this example.



continued

4. Right click on the 1769-ADN and choose Properties



5. Click on the I/O Bank 1 Configuration tab, then choose upload when prompted. The actual 1769-ADN I/O layout appears. From this screen you can configure the I/O modules in the 1769-ADN system by simply clicking on the slot number box associated with each I/O module.

6. When the I/O modules are configured, click on the Summary tab. Note the number of bytes of input and output data. This will be used later when adding the adapter to the 1769-SDN's scanlist.

7. Click Apply, then OK to save the configuration and download it to the adapter.

For this example, you only configure the two analog modules. For more information about analog modules, see the *Compact I/O Analog Modules User Manual*, publication 1769-UM002. Only analog and specialty modules are configurable. Discrete I/O modules, power supplies, and end caps are not configurable.

TIP

Configuration changes made to the adapter or any of its I/O modules with RSNetWorx will not be saved or downloaded to the adapter once the adapter is configured in a scanner's scanlist.

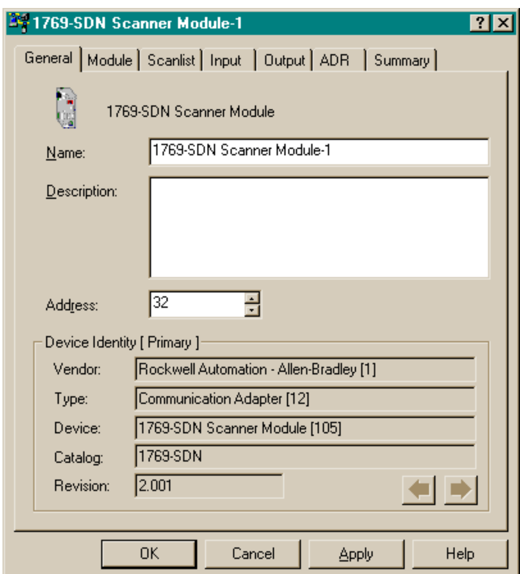
To make configuration changes, the controller must be placed into the Program mode and the adapter must be temporarily removed from the scanner's scanlist.

Step 2: Setting Up the 1769-SDN Scanlist

The 1769-SDN series B scanner supports automatic device recovery (ADR). An ADR tab appears in the scanlist window in RSNetWorx for DeviceNet for series B scanners so you can enable the ADR feature. This feature:

- automates the replacement of a failed slave device on a DeviceNet network by returning the device to the prior level of operation
- includes automatic address recovery which allows a slave device to be removed from the network and replaced with another identical slave device that is residing on the network at node 63 and is not in the scanlist.
- includes configuration recovery which allows a slave device to be removed from the network and replaced with an identical device with the same configuration

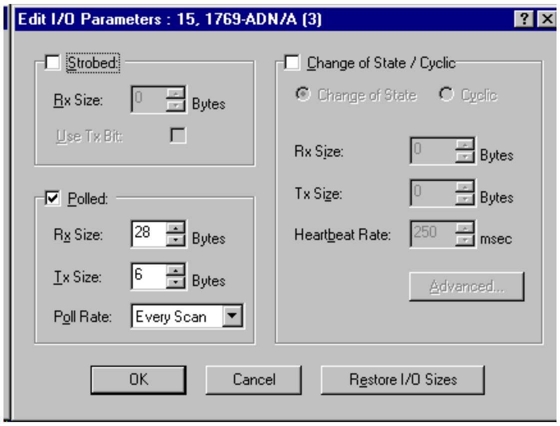
1. Right click on the 1769-SDN and choose Properties.



2. Click the Scanlist tab, then click Upload when prompted. The area on the left is called “Available Devices” and the area on the right is called “Scanlist”. The 1769-ADN adapter should be on the left.

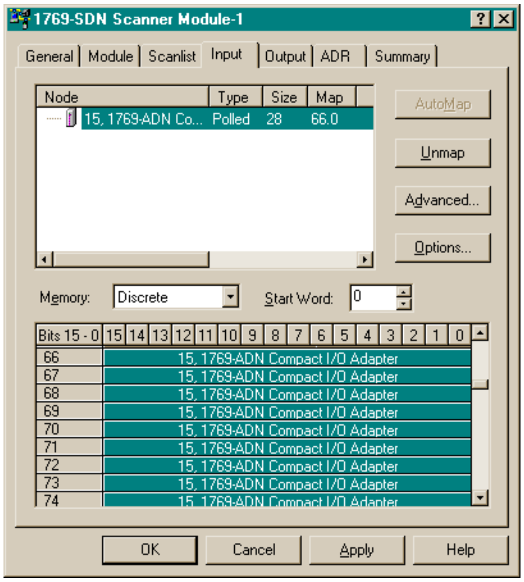
3. Click on the adapter, then click on the single arrow pointing to the right. This moves the adapter from Available Devices to the scanner’s scanlist.

4. Click on the Edit I/O Parameters button



5. Verify that the Rx Size and Tx Size are correct. The Tx (Transmit) and Rx (Receive) sizes correspond to the total number of output and input bytes noted from the adapter’s summary page. In this example, the scanner transmits 6 bytes to the adapter (output data) and receives 28 bytes from the adapter (input data). Click OK when finished with this screen.

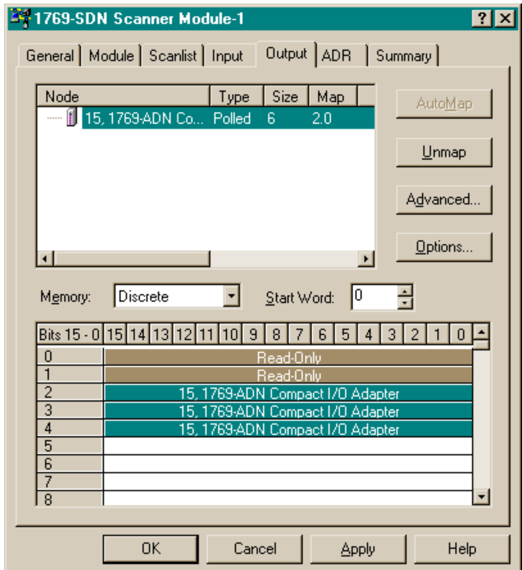
6. Click on the Input tab.



Click Apply and then click OK.

The first 66 words of input data (0 to 65) are read-only. For this example, 28 bytes or 14 words of input data will be mapped by the scanner to the 1769-L30 controller’s input tag, beginning with word 66. The 1769-ADN adapter also adds 2 words of status information to the input data. Therefore, the actual input data from the I/O modules in the 1769-ADN adapter’s system begins at word 68.

7. Click on the Output tab.



Click Apply and then click OK.

The first 2 words of output data (0 and 1) are read-only from the scanner’s point of view, but are control words for the controller. Bit 0 of output word 0 is used for control. It is the scanner’s Run bit. When set (1), it places the scanner into Run mode. When the scanner’s Run bit is a 0, the scanner is in Idle mode even if the controller is in Run mode.

The output data begins with word 2. This is where the actual output data goes for the output modules in the 1769-ADN adapter’s system.

TIP

The input and output data being exchanged by the scanner and adapter is packed data. This means that there is no special structure to it that makes it obvious which I/O module it is associated with.

To establish which data is from which module, you must list the number of input and output words each module has. Then, based on its position in the I/O bank, you can determine where any module’s data is in the controller’s I/O tags.

Transferring Data

There are 28 bytes of input data and 6 bytes of output data for this example. The I/O modules in the adapter's system are:

Module	Input	Output
ADN Status Information (added by the 1769-ADN)	2 words	0 words
1769-IA16	1 word	0 words
1769-OB16	1 word	1 word
1769-IF4	6 words	0 words
1769-OF2	4 words	2 words
Total Words	14 words	3 words
Total Bytes	28 bytes	6 bytes

The total is 14 input words or 28 input bytes. The first two input words are adapter status, leaving 12 words (24 bytes) for data. In this example, words 66 and 67 are the two words of status in the controller input tag for the scanner. The actual input data is then mapped to the controller's input data tag at the following word locations:

Location	Description
Words 66 and 67	1769-ADN Status Information
Word 68	1769-IA16 module's input word
Word 69	1769-OB16 module's input data (output data echo)
Words 70 to 75	1769-IF4 module's input data
Words 76 to 79	1769-OF2 module's input data

The output data can be determined in a similar manner. This data begins with word 2 of the output tag in the controller for the scanner as follows:

Location	Description
Word 0 and 1	See Module Command Array on page 6-9.
Word 2	1769-OB16 module's output word
Words 3 and 4	1769-OF2 module's output words

Module Command Array

The module command array is the primary control interface between your control program and the module.

Output Word	Bit	Description	Behavior
0	0	Run	This bit controls when the module scans its mapped slave devices. When set (1), the scanner will process I/O data as defined by its scanlist. To actually scan the network the Fault and Disable Network command bits must be clear (0).
	1	Fault	When set, the scanner's I/O mode will be Halt; messaging will still operate. The fault bit is primarily used to artificially set the slave devices into a fault state due to some event or condition within the control program.
	2	Disable Network	When set, the scanner is functionally removed from the network.
	3	Reserved(1)	na
	4	Reset	Restarts access to the DeviceNet network.
	5 to 15	Reserved(1)	na
1	16 to 31	Reserved(1)	na

(1) DO NOT manipulate Reserved Bits. Doing so may interfere with future compatibility.

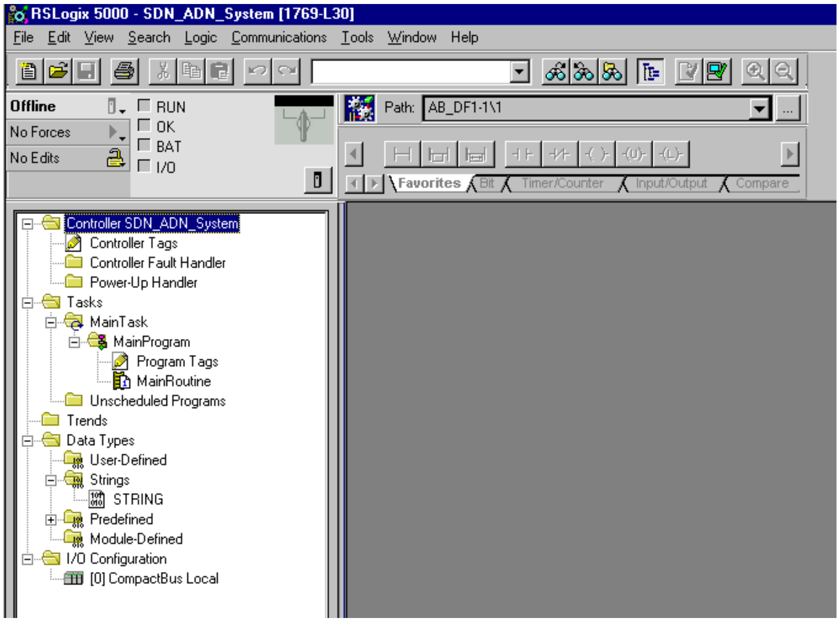
Download the scanner information to the 1769-SDN

After you configure the scanlist, you need to download that information to the 1769-SDN module.

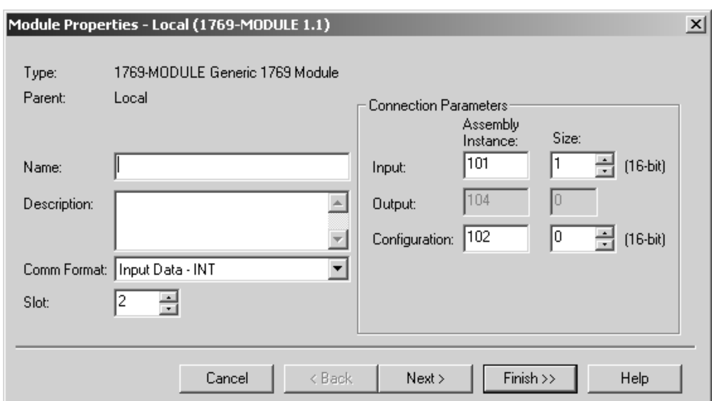
IMPORTANT You must use a DeviceNet connection to download scanner information to the 1769-SDN module. The CompactLogix controller does not support pass-through from its serial port to the 1769-SDN module.

Step 3: Creating a Project for the CompactLogix Controller

1. In RSLogix 5000 software, create a new project for a 1769-L30 CompactLogix controller.



2. Configure the local I/O in the Controller Organizer. The configure the 1769-SDN scanner. The scanner does not have a profile, so use the 1769 MODULE generic profile.



In the generic profile, select Data - INT for the communication format and specify these connection parameters:

Data	Assembly Instance	Size
Input	101	80
Output	100	5
Config	102	0

All tags for I/O modules are automatically created when the profiles for these modules are configured. Double click on Controller Tags in the controller organizer to view these tags. Each I/O module slot has Input, Output and Configuration tags created, if they apply. These Local I/O tags are structured as follows:

Tag	Definition
Local:s:I	s is the slot number I represents Input Data
Local:s:O	O represents Output Data
Local:s:C	C represents Configuration Data

Each of these tags can be further expanded by clicking the plus sign to the left of its entry. For example, click the **[+]** to the left of Local:3:I and the following tags appear:

Local:3:I.Fault
Local:3:I.Data

The Fault tag is a DINT, so clicking its plus sign only reveals its 32 bits. This is status information concerning the module's connection to the CompactLogix controller. Clicking the plus sign to the left of the Data tag reveals the 80 input words created when this value was entered into the Generic profile for the scanner. For this example, the input addresses for the scanner are broken down as follows:

Tag	Definition
Local:3:I.Data[0] through Local:3:I.Data[65]	Read Only Status
Local:3:I.Data[66] through Local:3:I.Data[67]	1769-ADN Status Information
Local:3:I.Data[68]	Input Data from 1769-IA16
Local:3:I.Data[69]	Input (output echo) Data from 1769-OB16
Local:3:I.Data[70] through Local:3:I.Data[75]	Input Data from 1769-IF4
Local:3:I.Data[76] through Local:3:I.Data[79]	Input Data from 1769-OF2

The Output tag created for the scanner is as follows:

Local:3:O

Click the **[+]** to its left to reveal Local:3:O.Data.

Click the **[+]** to its left and the 5 words of output data created when you created the Generic profile for the scanner will be displayed. This output data is broken down as follows for this example:

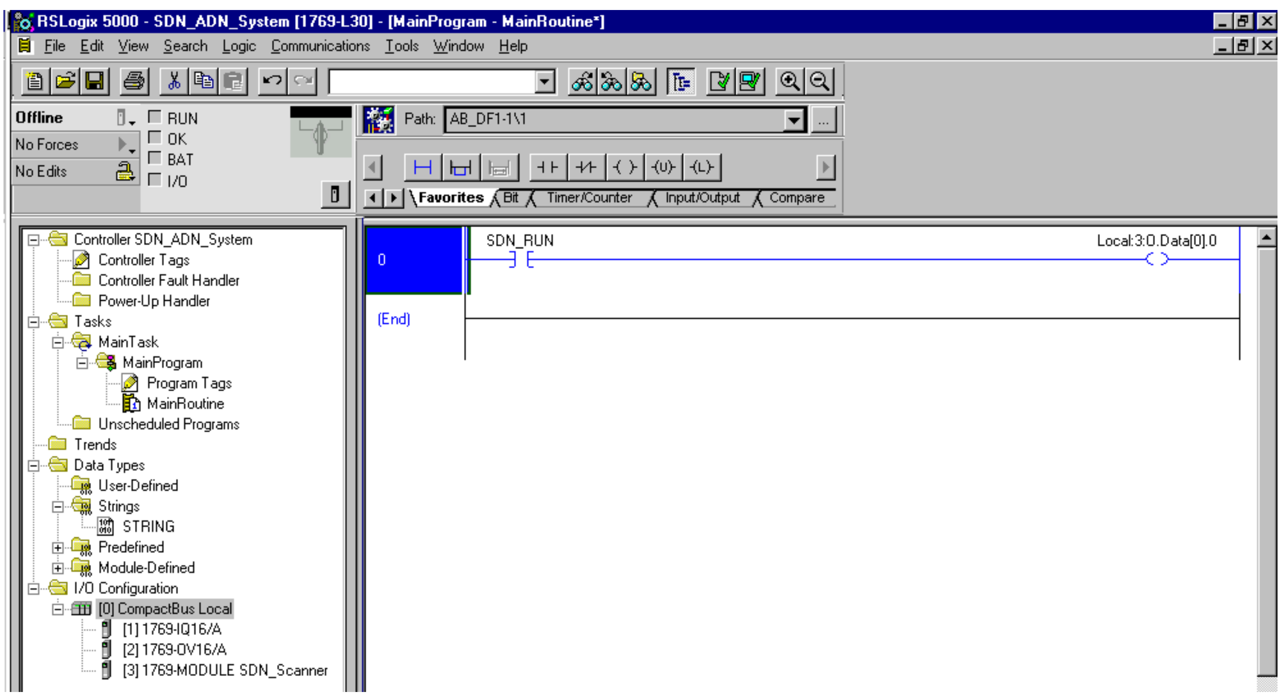
Tag	Definition
Local:3:O.Data[0] through Local:3:O.Data[1]	Control to 1769-SDN[(1)]
Local:3:O.Data[2]	Output data for 1769-OB16
Local:3:O.Data[3] through Local:3:O.Data[4]	Output data for 1769-OF2

(1) Bit 0 of output word 0 is the Run mode bit for the scanner. With the controller in Run mode, set this bit to a 1 to place the scanner into the Run mode. With the controller in Run and this bit a 0, the scanner will be in Idle mode and will not scan I/O. The Module Command Array is described in more detail on page 6-9.

Step 4: Enter Program Logic

The program for this example consists of a single rung that is used to place the scanner into the RUN mode.

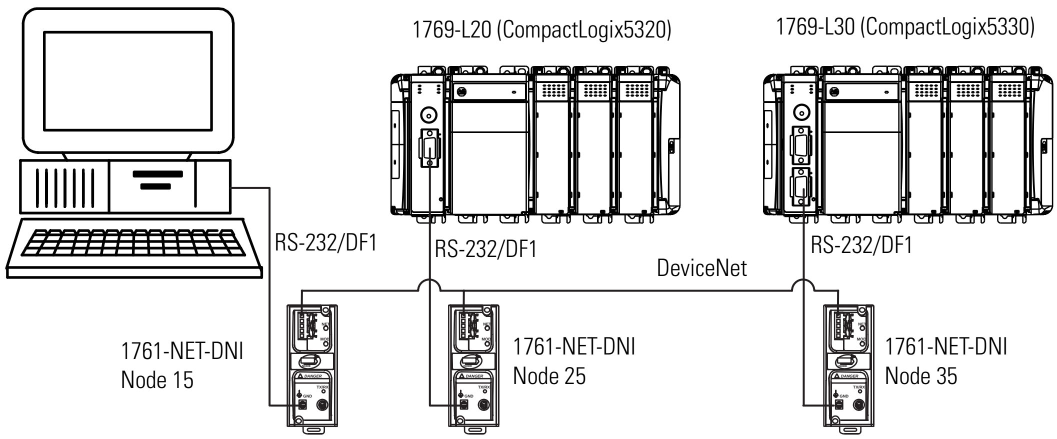
To place the scanner in the Run mode when the CompactLogix controller is in the Run mode, either set “SDN_RUN” to a 1, or remove it from the program. When “SDN_RUN” is removed, the scanner’s Run bit is always in Run when the controller is in Run.



When your program is written, verify and save it, then download it to your controller to run and test your system.

Example 2: Sending Messages Over DeviceNet Using a 1761-NET-DNI Interface Converter

.This example uses a 1761-NET-DNI interface converter to connect the CompactLogix controller to the DeviceNet network. Use the serial port(s) of the CompactLogix controller to connect to a DeviceNet network using the 1761-NET-DNI DeviceNet interface converter.



Step 1: Configure the Hardware

Connecting CompactLogix controllers on DeviceNet requires one 1761-NET-DNI per CompactLogix controller. The DNI converts RS-232 hardware connections and full duplex DF1 protocol to DeviceNet. A computer can also be connected to the DeviceNet network with another 1761-NET-DNI. The Full Duplex DF1 communication driver in RSLinx can be used to allow RSLogix 5000 programming software to upload/download and monitor programs in the CompactLogix controller over the DeviceNet network.

Step 2: Commissioning the 1761-NET-DNI Modules on DeviceNet

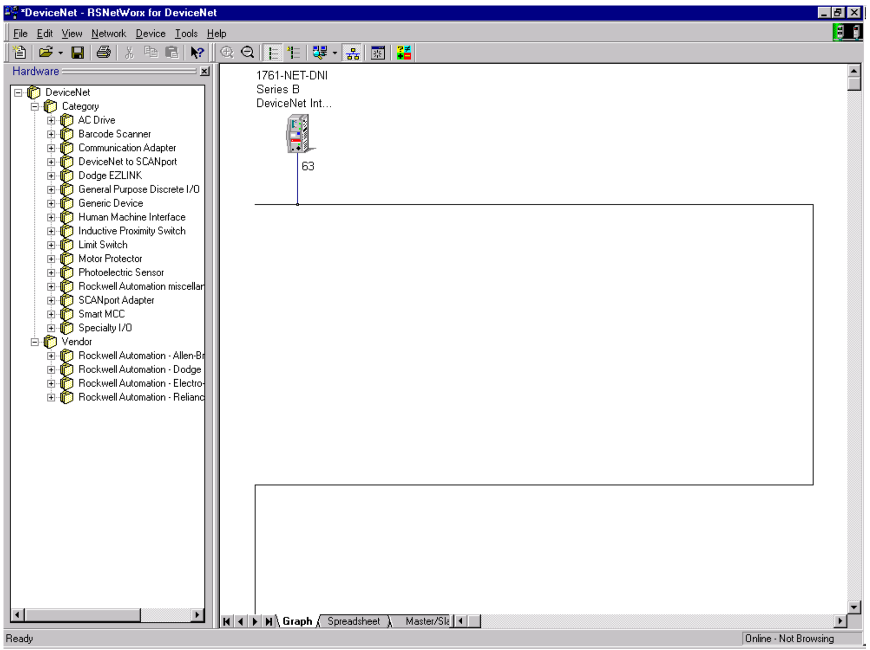
The DNI must be commissioned on the DeviceNet network via the RSNetworx for DeviceNet software or the DNI Configuration Utility, version 2.001.

TIP

The DNI Configuration Utility, a free tool for commissioning and configuring the DNI, is available for download at **www.ab.com/micrologix**.

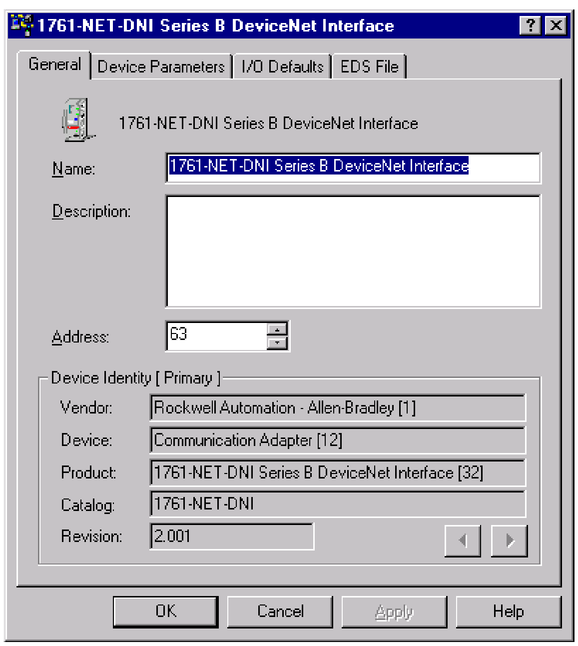
Commissioning assigns node addresses to the DNI modules. Each device on the network must have a unique node address. The DNI then routes DF1 messages from each CompactLogix controller to the other CompactLogix controller via DeviceNet and the other DNI module.

1. Start RSNetworx for DeviceNet and connect one of the 1761-NET-DNI modules to the DeviceNet network with your PC. When power is first applied to a DNI, it powers up at node 63. In RSNetworx, click on the online icon or click on the "Network" pull-down menu and select Online.



2. The first DNI module appeared as node 63. To change it to any other unique node address, right click on the DNI module and select Properties.

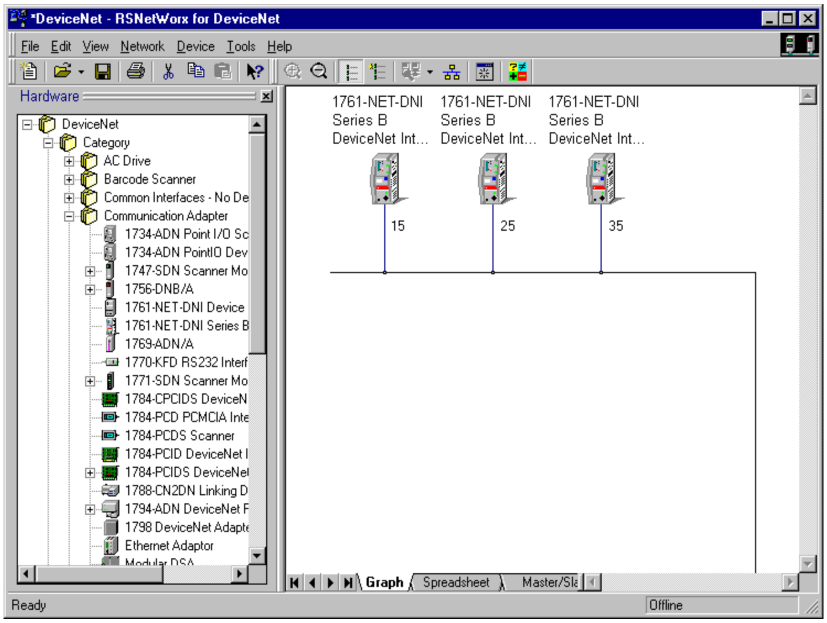
Enter a new node address into the Address field. For this example, addresses 15, 25, and 35 are used for the three DNI module's DeviceNet node addresses. Modify this DNI module's node address to 15. Click Apply and OK.



continued

3. Next, connect the second DNI module to the DeviceNet network. Click the Network pull down menu and select “Single Pass Browse”. The second DNI module should appear at node 63 if the unit is new. Modify its DeviceNet node address as we did with the first DNI module.

The DNI modules should now be at nodes 25 and 35.



You have completed the commissioning of your DNI modules with addresses 15, 25, and 35. You can go offline and exit the RSNetworx for DeviceNet software.

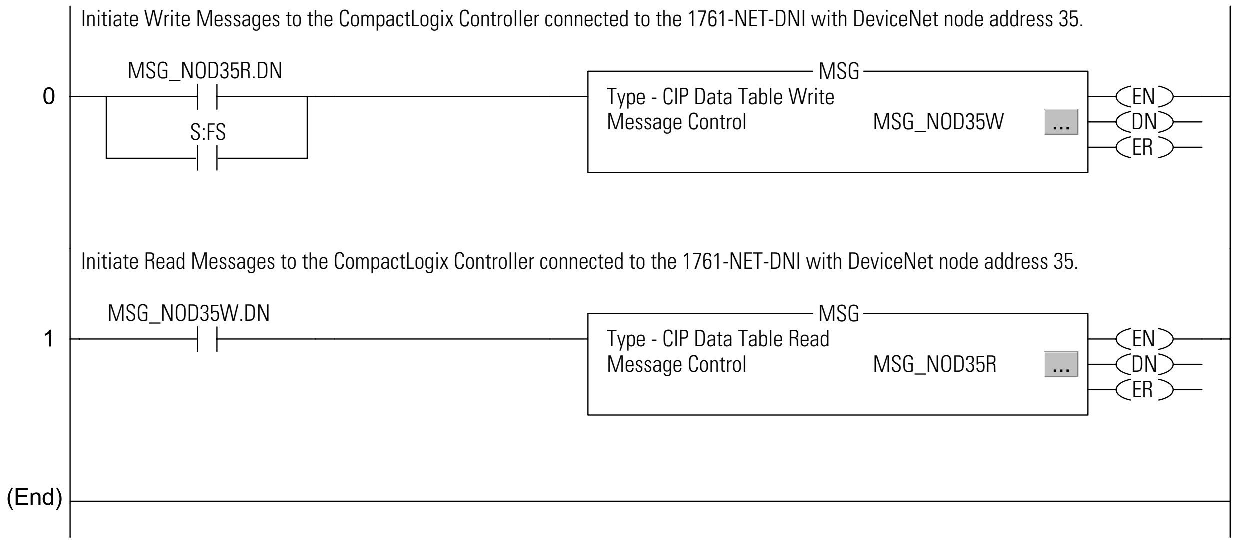
Step 3: Connecting the Controllers to the DeviceNet Network

Connect the serial channel of each CompactLogix controller to the round mini-din channel on each respective DNI module with a 1761-CBL-PMO2, series B RS-232 or 1761-CBL-AP00 cable.

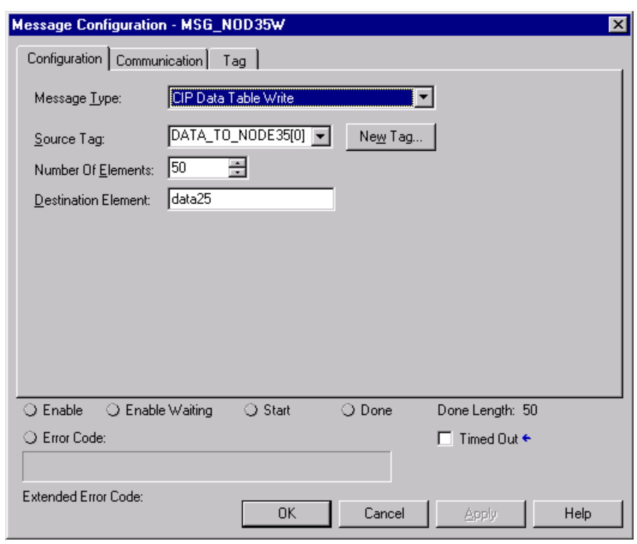
For this example, connect the 1769-L20 to DNI node 25 and connect the 1769-L30 to DNI node 35. Connect the PC to DNI node 15.

Step 4: Sending Messages

1. Start RSLogix 5000 and create a new project for the 1769-20 controller. Add two ladder rungs, each containing MSG Instructions; one to Write data to the controller and one to Read data from the other controller. For this example, the MSG instructions are executed alternately, initiated at power-up or going to Run mode with the first scan bit (S:FS). This allows each MSG Instruction to initiate the other MSG Instruction each time it completes.



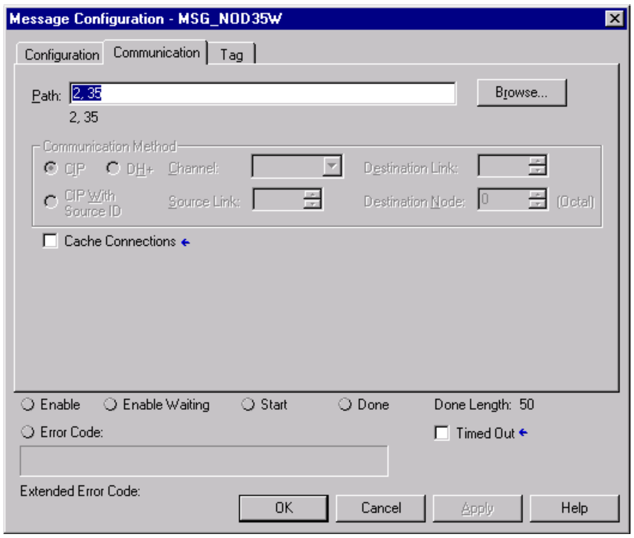
2. The Configuration tab for the MSG Write Instruction to write fifty 16-bit signed integer data words to the other controller, looks like the following:



continued

The message is a CIP Data Table Write message. In the controller connected to the DNI node 25, the source tag for the message is "DATA_TO_NODE35". The destination element is a tag in the 1769-L30 controller. For this example, this tag must be an array of at least 50 integer words in length. It is the destination where the data from this MSG Instruction will be sent.

3. The Communication tab for the MSG Write Instruction looks like the following:

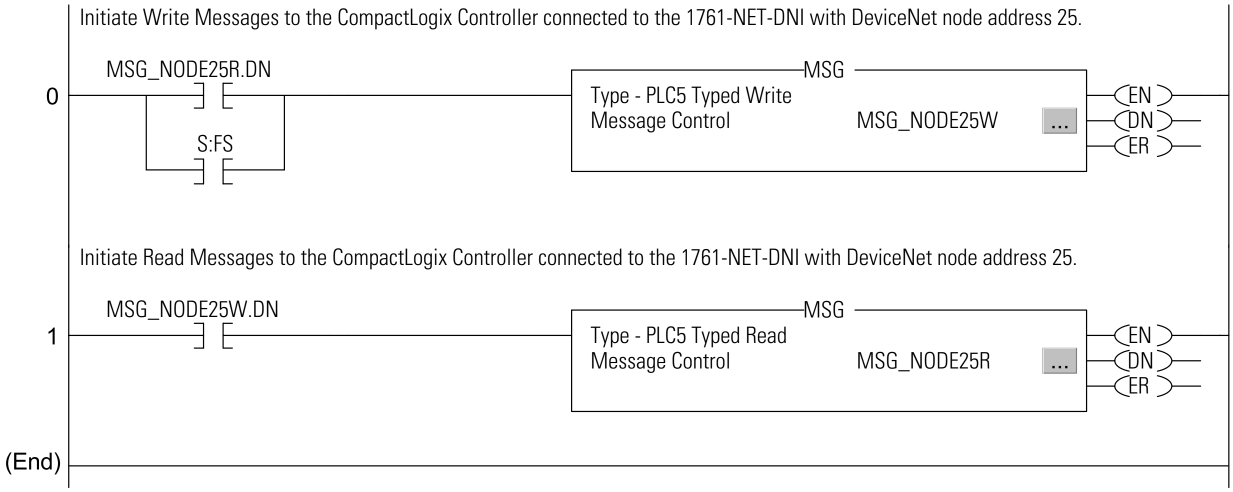


The path parameter is the only field to modify. The path shown above (2, 35) routes the message out port 2 (channel 0) of the local 1769-L20 controller and to a destination node address 35. This is the DeviceNet node address of the DNI to which the remote 1769-L30 controller is connected. The cache connections selection is not valid for serial port communications.

The second message in rung 1 of the ladder logic is very similar to the write message. The difference is that for the read message, the message type is CIP Data Table Read, the source is data25 and the destination is a tag in the sending controller called DATA_FROM_NODE35. The tag data25 tag is located in the 1769-L30 1769-L20 controller read and write data.

Before saving the project, be sure that the port parameters of the serial port match those of the DNI module's serial port. The default RS-232/full-duplex DF1 port parameters for the DNI modules and the CompactLogix controllers are identical, except for the error checking method. The DNI module uses the CRC check by default, while the CompactLogix controller's serial port defaults to BCC.

If you want the 1769-L30 controller send read and write messages to the 1769-L20 controller, add the same two message rungs to it and create the necessary tags.



TIP

If using Channel 1 on the CompactLogix5330 controller, this is Port 3 for each MSG instruction's Path.

Communications on Ethernet

Using This Chapter

For information about:	See page
Configuring your system for an Ethernet link	7-1
Configuring an Ethernet drive in RSLinx	7-8
Example 1: Create MSG programs for SLC 5/05 and ControlLogix controllers	7-9
Example 2: Staggering multiple messages	7-12

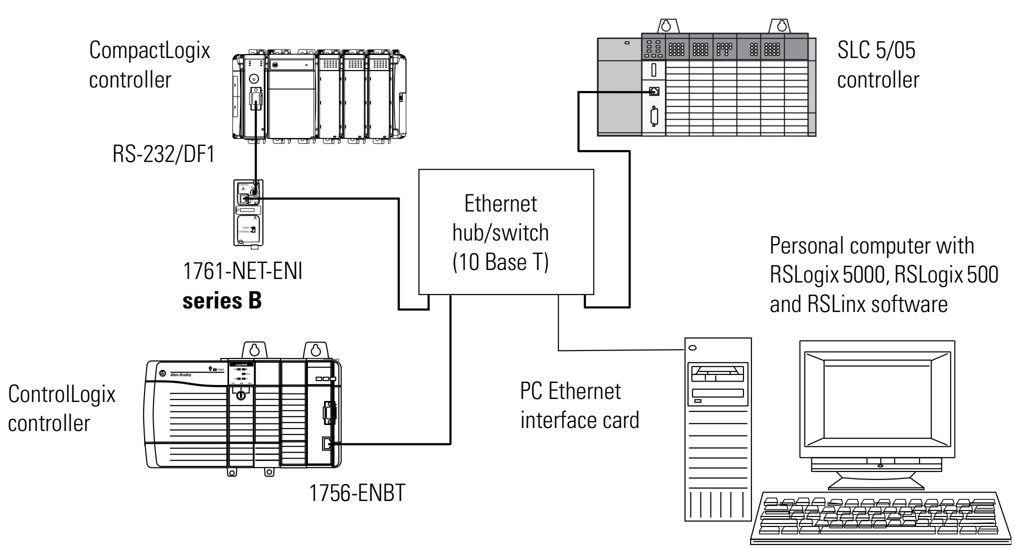
Configuring Your System for an Ethernet Link

Connecting CompactLogix controllers on Ethernet requires one 1761-NET-ENI per CompactLogix controller.

In this example, the Ethernet interface card in the computer connects directly to the:

- SLC 5/05 controller (channel 1)
- ControlLogix controller via the 1756-ENBT card
- CompactLogix controller via a 1761-NET-ENI interface.

  This example uses a 1761-NET-ENI series B module. If you connect a 1761-NET-ENI series A module to the CompactLogix controller, you must also connect a 1761-NET-ENI module to the personal computer.



The computer must have the following software:

- RSLogix 5000, version 7.00 or later for the 1769-L20 controller, version 8.00 or later for the 1769-L30 controller
- RSLinx, version 2.30.00 or later
- RSLogix 500 software
- ENI Configuration Tool

The 1769-L20 controller must contain firmware 7.17 or later. The 1769-L30 controller must contain firmware 8.14 or later.

Step 1: Assigning IP Addresses

Each Ethernet device requires a unique IP address. If your Ethernet network is isolated from the company-wide network, any valid IP addresses may be used. If your Ethernet hub is connected to a larger Ethernet network, contact your System Administrator for unique IP addresses. This example uses these IP addresses:

IP Address	Device
131.200.50.92	SLC 5/05 controller
131.200.50.93	1756-ENBT
131.200.50.94	1761-NET-ENI (CompactLogix controller)
131.200.50.96	computer's Ethernet card

The ENI can be configured with IP addresses assigned to node numbers 0 to 49. In the ENI, node addresses 45 through 49 are dedicated for Logix controllers. Node addresses 0 through 44 are used for all other Ethernet devices, such as other CompactLogix controllers connected to ENI modules and SLC 5/05 controllers.

The subnet mask for each Ethernet device is 255.255.0.0.

IMPORTANT

The RS-232/DF1 interface for the CompactLogix controller should be 38400 baud. This allows fast upload/download of programs.

On the CompactLogix controller, select 38400 baud. Also, set the number of Stop Bits in RSLinx and the controller to 2.

Verify that the baud rate for the ENI module is set to autobaud. In autobaud, the ENI assumes series B functionality and determines its actual baud rate from the controller it is connected to.

Step 2: Configuring the 1761-NET-ENI

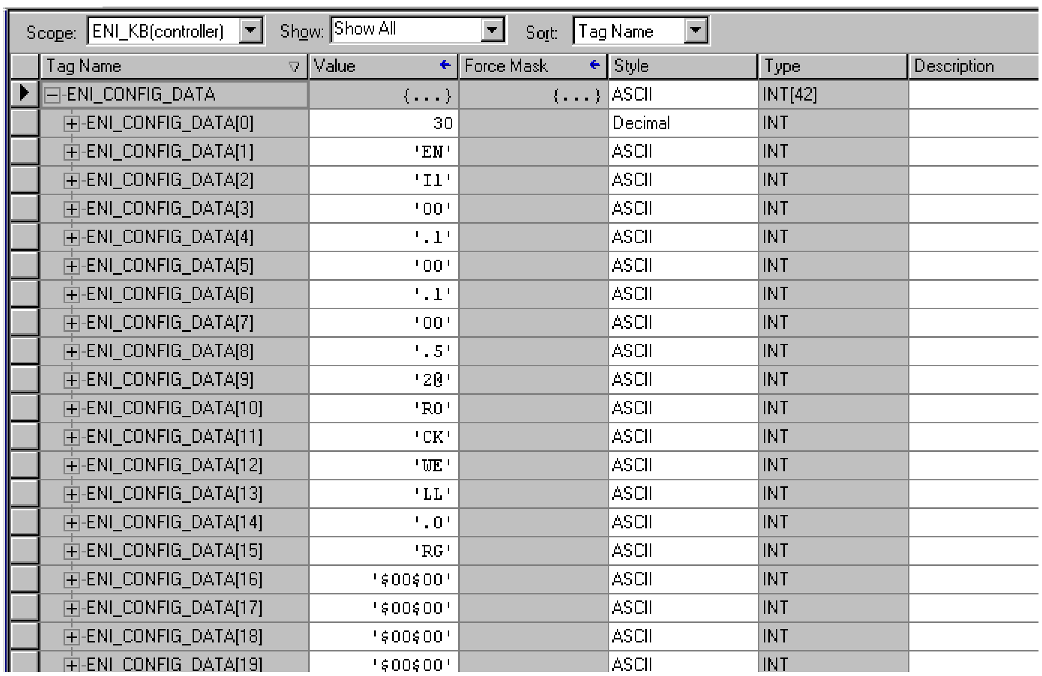
You can configure a 1761-NET-ENI module (node 248 to 254) using a MSG instruction. This example configures an ENI module as node 249. The default address is Eni0.0.0.0@eni1761.org.

The ENI is only compatible with the MicroLogix type string files. To configure the ENI module:

1. Create a tag to be the source element in your MSG write instruction.

- The data type must be integer
- The size must be 42. A MicroLogix string file contains 42 elements.

In this example, the tag name is ENI_CONFIG_DATA:



Scope: ENI_KB(controller) Show: Show All Sort: Tag Name

Tag Name	Value	Force Mask	Style	Type	Description
ENI_CONFIG_DATA	{...}	{...}	ASCII	INT[42]	
ENI_CONFIG_DATA[0]	30		Decimal	INT	
ENI_CONFIG_DATA[1]	'EN'		ASCII	INT	
ENI_CONFIG_DATA[2]	'I1'		ASCII	INT	
ENI_CONFIG_DATA[3]	'00'		ASCII	INT	
ENI_CONFIG_DATA[4]	'.1'		ASCII	INT	
ENI_CONFIG_DATA[5]	'00'		ASCII	INT	
ENI_CONFIG_DATA[6]	'.1'		ASCII	INT	
ENI_CONFIG_DATA[7]	'00'		ASCII	INT	
ENI_CONFIG_DATA[8]	'.5'		ASCII	INT	
ENI_CONFIG_DATA[9]	'2@'		ASCII	INT	
ENI_CONFIG_DATA[10]	'RO'		ASCII	INT	
ENI_CONFIG_DATA[11]	'CK'		ASCII	INT	
ENI_CONFIG_DATA[12]	'WE'		ASCII	INT	
ENI_CONFIG_DATA[13]	'LL'		ASCII	INT	
ENI_CONFIG_DATA[14]	'.O'		ASCII	INT	
ENI_CONFIG_DATA[15]	'RG'		ASCII	INT	
ENI_CONFIG_DATA[16]	'$00$00'		ASCII	INT	
ENI_CONFIG_DATA[17]	'$00$00'		ASCII	INT	
ENI_CONFIG_DATA[18]	'$00$00'		ASCII	INT	
ENI_CONFIG_DATA[19]	'$00$00'		ASCII	INT	

2. In ENI_CONFIG_DATA[0], change the style to decimal and enter the number of ASCII characters in the email address.

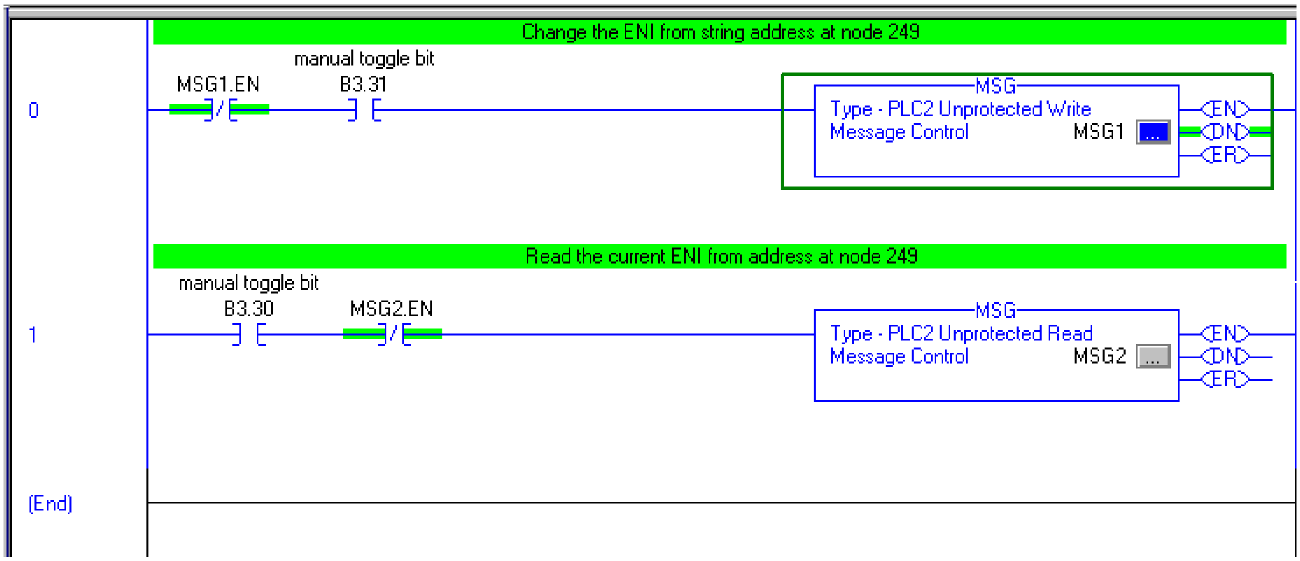
   In this example, the address is ENI100.100.100.52@rockwell.org, so enter 30 (for 30 characters).

3. In ENI_CONFIG_DATA[1], type in the entire email address.

   The email address uses two ASCII characters for each integer word in the array.

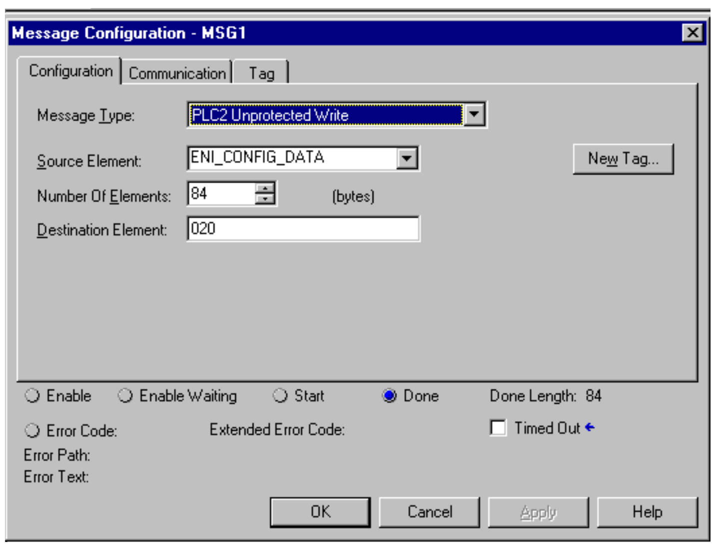
   For an odd number of characters, the last element should be 'x$00', where x is the last character in the email address.

4. Enter a PLC2 type unprotected write MSG instruction.



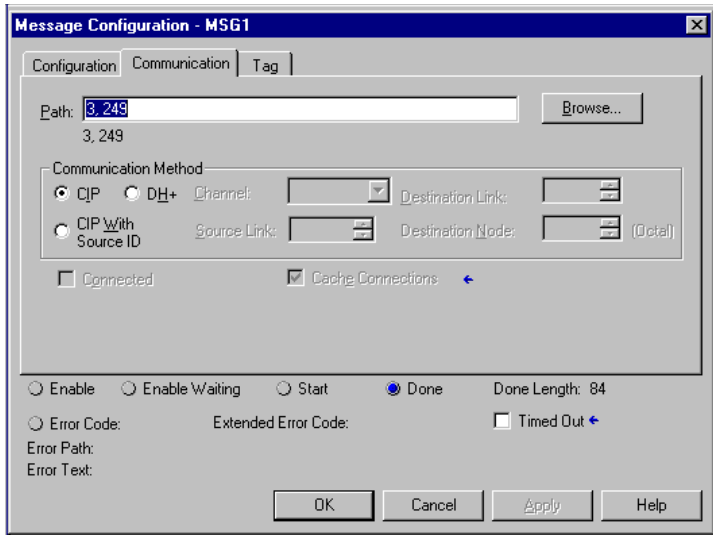
5. Configure the MSG instruction as shown below. The destination element can be any legal value.

   The number of elements is in byte and must equal 84.



6. Set the MSG communication path. The 3 represents channel 1 on a 1769-L30 controller and 249 is the ENI node address.

   If you use channel 0, enter 2.

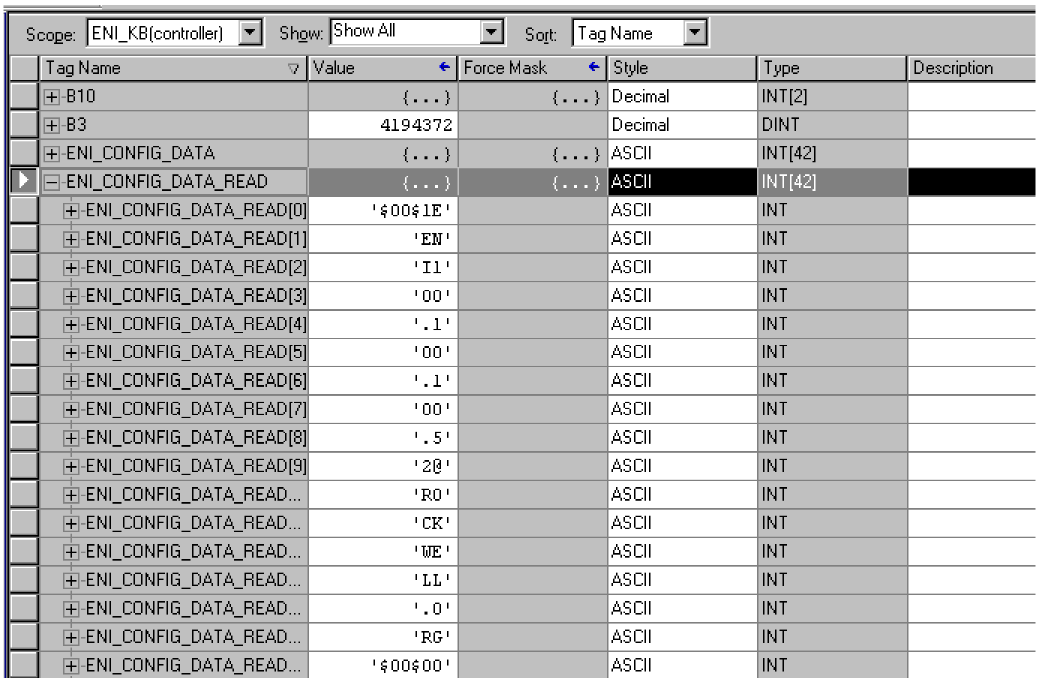


Confirming the configuration

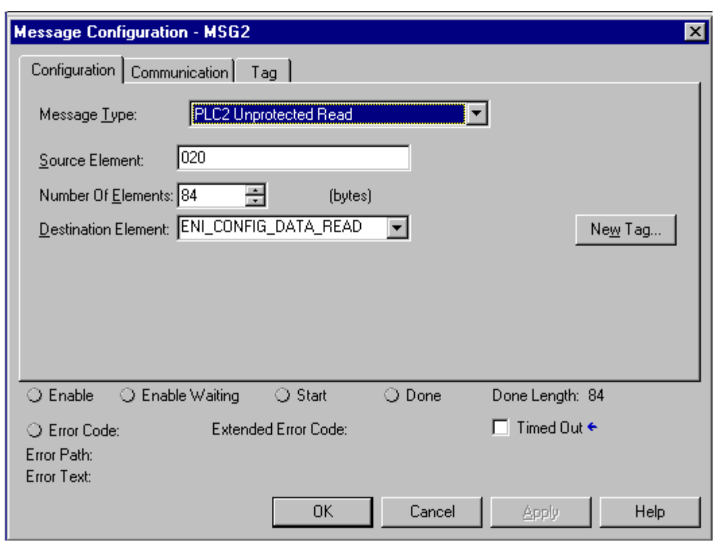
To confirm that the ENI was configured properly, use a MSG read instruction.

1. Create a tag to be used in the MSG read instruction.

In this example, the tag name is ENI_CONFIG_DATA_READ:



2. Configure the MSG instruction as shown below. The communication path is the same as the path you used to configure the ENI module.

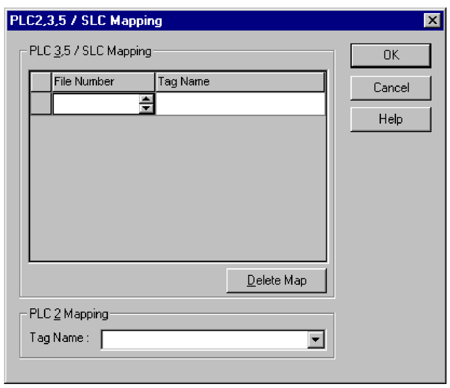


For more details on using the ENI module, see the *RS-232/DF1 Ethernet Interface (ENI) User Manual*, publication 1761-UM006.

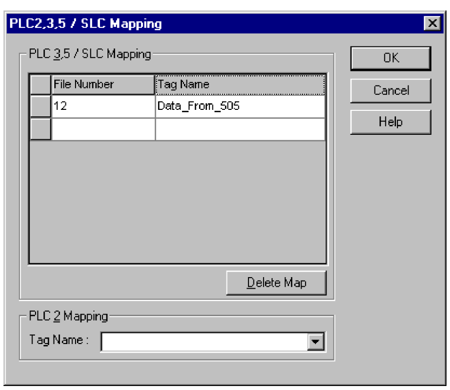
Step 3: Mapping Data Table Files

The CompactLogix controller does not use the structured data table addressing scheme used by PLC and SLC controllers. You must map PLC and SLC file numbers to CompactLogix tags. For example, a MSG sent by an SLC 5/05 controller to a CompactLogix controller uses a PLC-5 Typed Write command. The target data table address used is N12:0. This file 12 must be mapped to a valid tag name in the CompactLogix controller.

1. While offline in the CompactLogix controller project, click on the Logic pull-down menu and select Map PLC/SLC Messages.



2. In the File Number column, enter 12. Under the Tag Name, click on the right side in the white box to reveal your Controller Tags and select the tag name you created ("Data_From_505").You can map multiple entries.

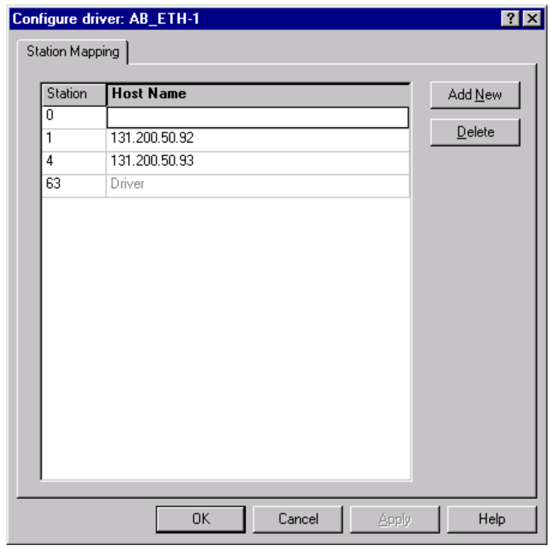


Configuring an Ethernet Driver in RSLinx

In order to download your programs to the any of the controllers via Ethernet, you must configure an Ethernet driver in RSLinx.

In RSLinx, click on the Communications pull-down menu and select Configure Drivers. Click on the arrow associated with the Available Driver Types box. Select Ethernet Devices, then click Add New.

Modify this screen to include the IP addresses of the SLC 5/05, 1756-ENBT module, and the ENI module.

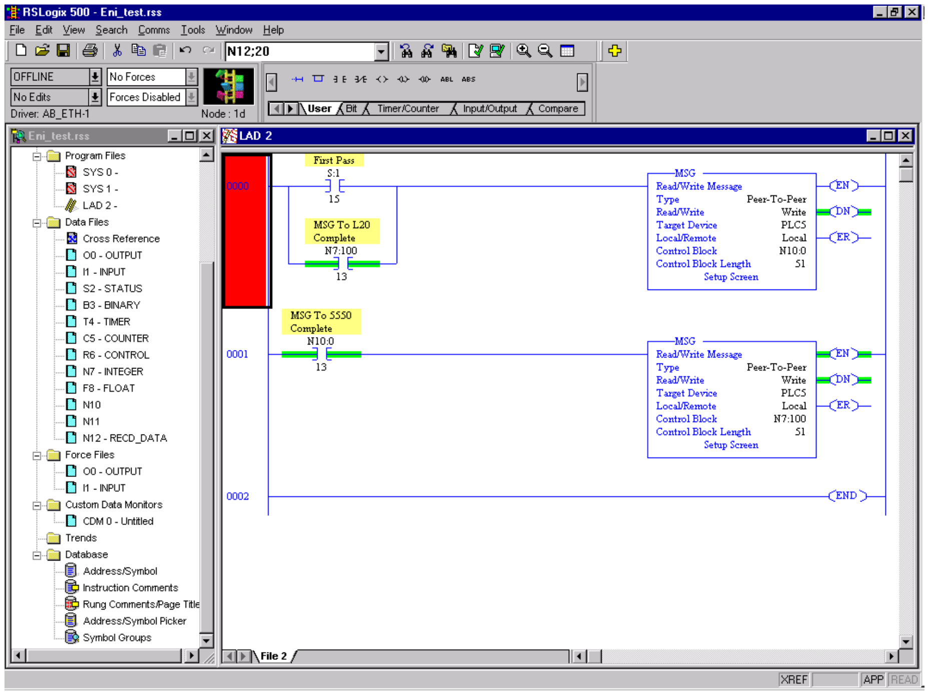


From the RSLogix 5000 software, you should be able to download your CompactLogix and ControlLogix programs. From the RSLogix 500 software, you should be able to download your SLC 5/05 program.

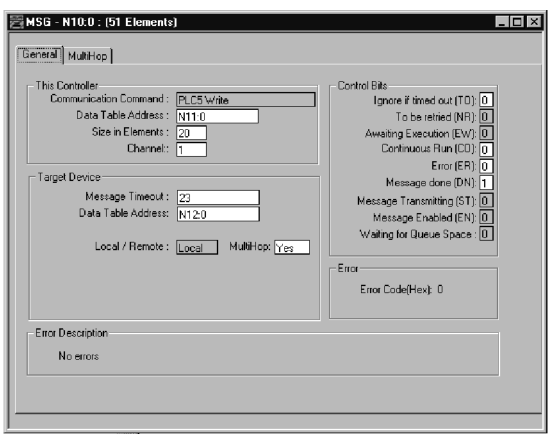
Once all programs are downloaded to their respective controllers, place each controller into the RUN mode, and a MSG from each controller will be sent to each of the other controllers. Each controller will only send one MSG at any given time. Go online with the controllers to verify the successful completion of the messages.

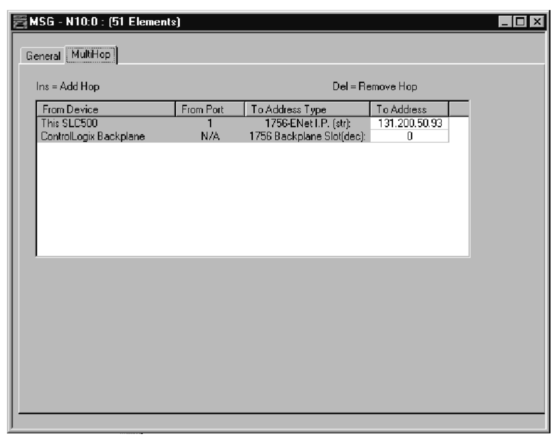
Example 1: Creating MSG Programs for SLC 5/05 and ControlLogix Controllers

The following is the ladder program is for the SLC 5/05 controller. The following RSLogix 500 screens show the tabs for each MSG instruction. Before saving your program, make sure to configure channel 1 with its IP address, subnet mask, and disable BOOTP. Then save the program.

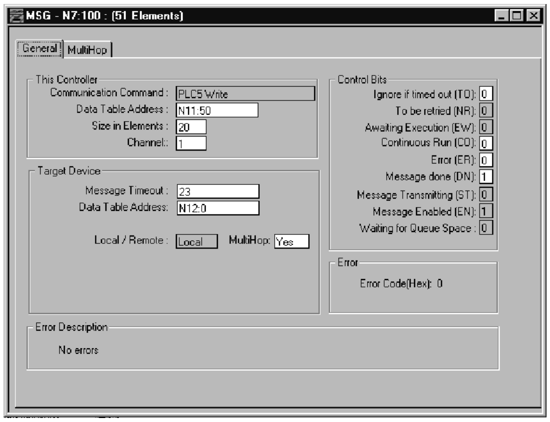


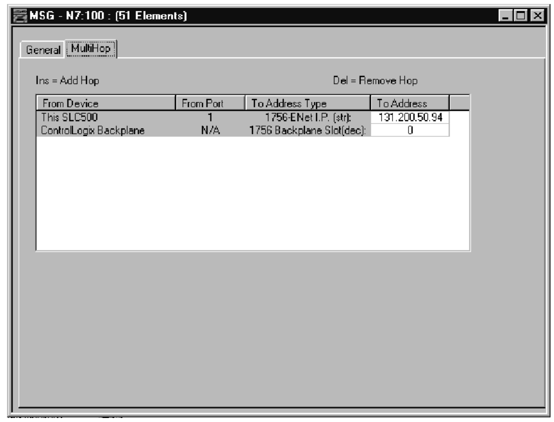
Configure the MSG instruction on rung 0 as follows:



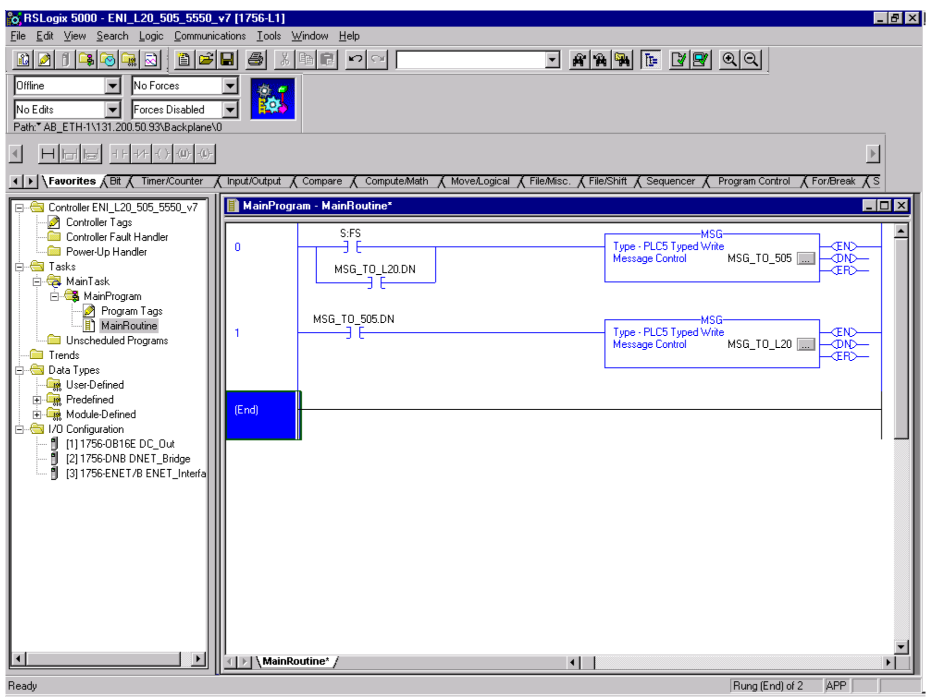


Configure the MSG instruction on rung 1 as follows:

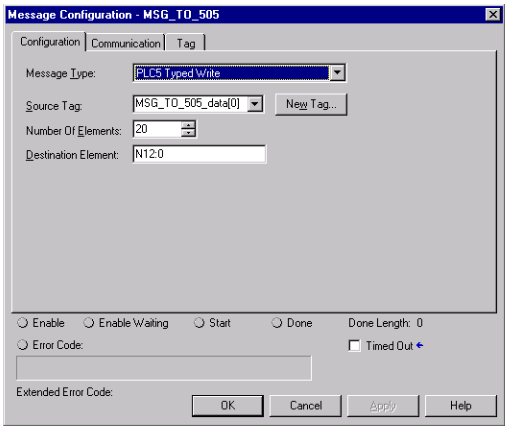


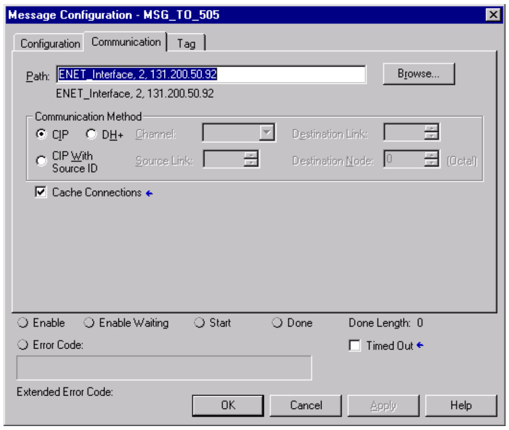


The following is the ladder program for the ControlLogix controller. The following RSLogix 5000 screens show the two tabs for each MSG Instruction. As part of the program, configure the Ethernet modules with the proper IP address, subnet mask, and disable BOOTP.

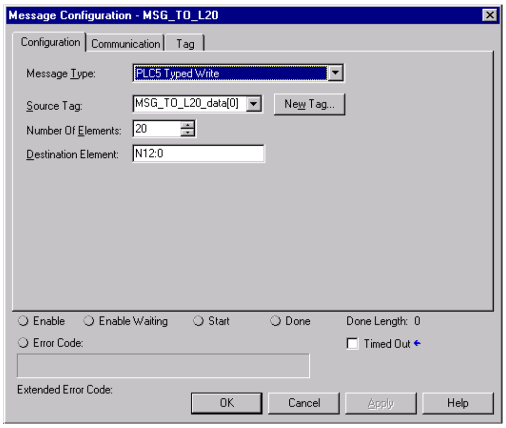


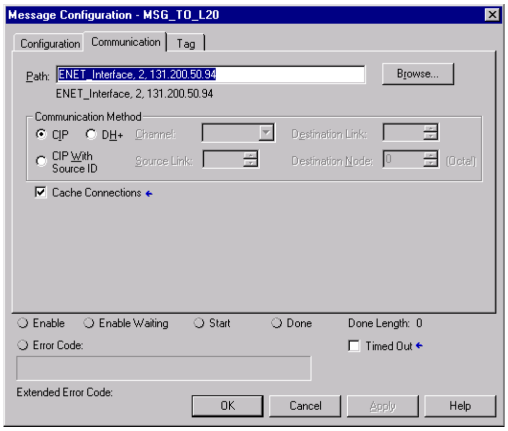
Configure the MSG instruction on rung 0 as follows:





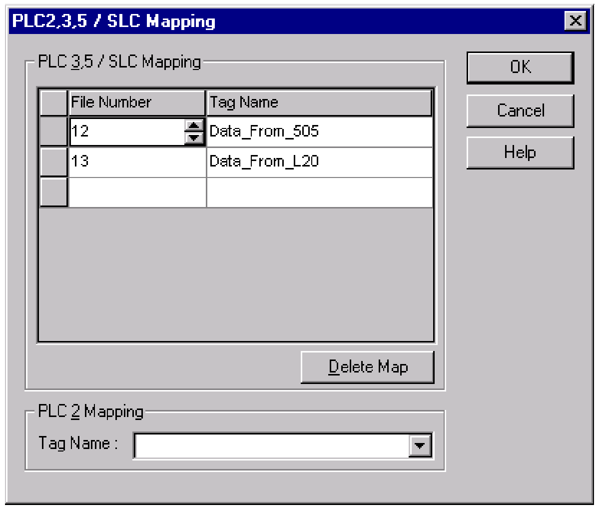
Configure the MSG instruction on rung 1 as follows:





The 1761-NET-ENI modules do not support CIP commands. Therefore, all commands between controllers used in this application example are PLC-5 Typed Write commands. These commands require a PLC-5 type address to send the data to the receiving controller. Such addresses do not exist in Logix controllers, so they must be mapped to existing tags in these controllers.

The mapped table for the ControlLogix program should look like:



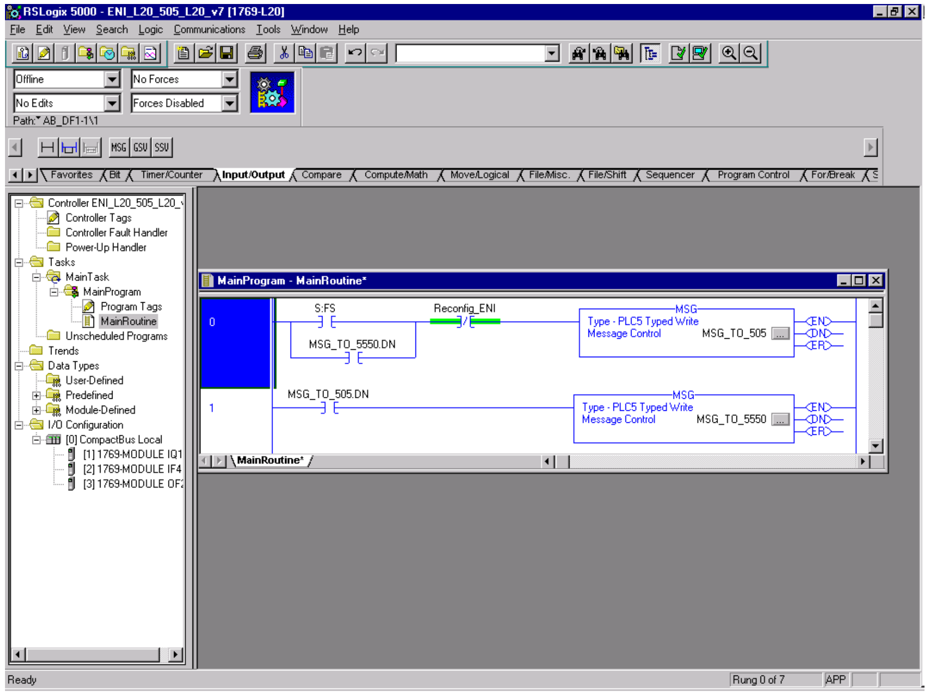
Example 2: Staggering Multiple Messages

In a CompactLogix controller, it is recommended to stagger multiple MSG instructions. For example, if there are two MSG instruction, let the first instruction complete before initiating the second instruction.

Keep in mind that over the serial port, the CompactLogix controller supports 12 message buffers. At most, you can have 4 simultaneous connected messages and 8 simultaneous unconnected messages. Or if all the messages are unconnected, you can have 12 simultaneous messages. If a message is greater than 250 bytes, it is divided across enough buffers to carry the message, which reduces the number of buffers remaining for other simultaneous messages.

Staggering messages keeps the amount of user memory needed for incoming and outgoing messages to a minimum. Each message requires approximately 1.1K bytes of user memory, allocated when the message is to be sent or received. If two messages were enabled at the same time, 2.2K bytes of user memory would be used.

This example shows how to program staggered MSG instructions.



Notes:

CompactLogix System Specifications

Using This Appendix

For information about:	See page
CompactLogix controller specifications	A-1
1747-BA battery specifications	A-2
Dimensions	A-3

CompactLogix Controller

Description	1769-L20	1769-L30
Communication ports	(1) RS-232	(2) RS-232
User memory	64K bytes	256K bytes
Maximum number of I/O modules supported	8 I/O modules	16 I/O modules
Maximum number of I/O banks supported	2 banks	3 banks
Backplane current	600 mA at +5V dc 0 mA at +24V dc 3W	800 mA at +5V dc 0 mA at +24V dc 4W
Operating temperature	0° to +60°C (+32° to +140°F)	
Storage temperature	-40° to +85°C (-40° to +185°F)	
Relative humidity	5% to 95% non-condensing	
Vibration	Operating: 10 to 500 Hz, 5G, 0.030 in. peak-to-peak	
Shock	Operating: 30G, 11 ms, panel mounted (20G, 11 ms, DIN rail mounted) Non-operating: 40G, panel mounted (30G, DIN rail mounted)	
Power supply distance rating	4 (The controller must be within 4 slot positions of the power supply.)	
Shipping weight	325 g (0.715 lbs)	362 g (0.796 lbs)
Battery	1747-BA	
Programming cable	1747-CP3 or 1756-CP3	
Agency certification	• C-UL certified (under CSA C22.2 No.142) • UL 508 listed • CE and C-Tick compliant for all applicable directives	
Hazardous environment class	Class I, Division 2, Hazardous Location, Groups A, B, C, D (UL 1604, C-UL under CSA C22.2 No. 213)	
Radiated and conducted emissions	EN50081-2 Class A	

Description	1769-L20	1769-L30
Electrical /EMC:	*The unit has passed testing at the following levels:*	
ESD Immunity (IEC61000-4-2)	4 kV contact, 8 kV air, 4 kV indirect	
Radiated Immunity (IEC61000-4-3)	10 V/m, 80 to 1000 MHz, 80% amplitude modulation, +900 MHz keyed carrier	
Fast Transient Burst (IEC61000-4-4)	2 kV, 5 kHz	
Surge Immunity (IEC61000-4-5)	1 kV galvanic gun	
Conducted Immunity (IEC61000-4-6)	10V, 0.15 to 80 MHz(1)	

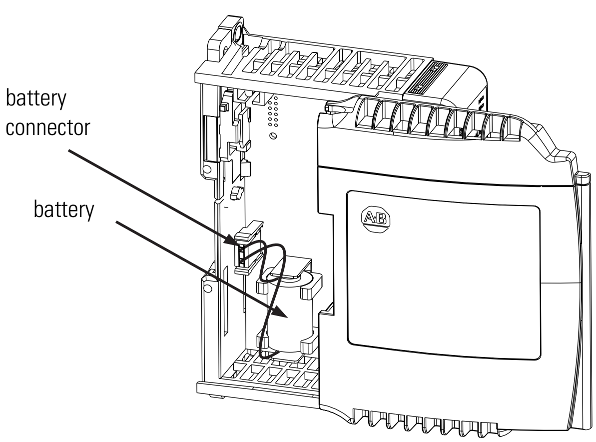
(1) Conducted Immunity frequency range may be 150 kHz to 30 MHz if the Radiated Immunity frequency range is 30 MHz to 1000 MHz.

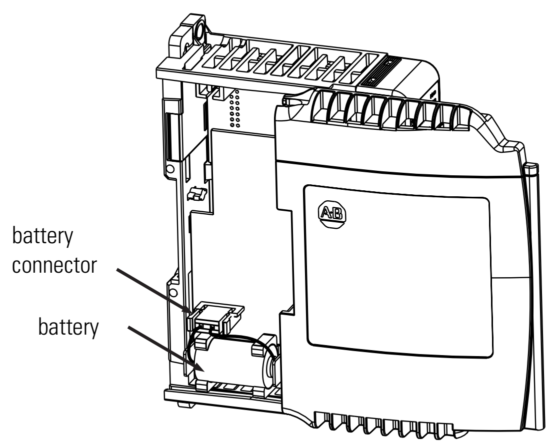
IMPORTANT The amount of memory that the software displays includes both the user available memory and the memory reserved for overhead. Certain operations dynamically allocate and de-allocate user-available memory.

1747-BA Battery

The CompactLogix controller uses the 1747-BA battery:

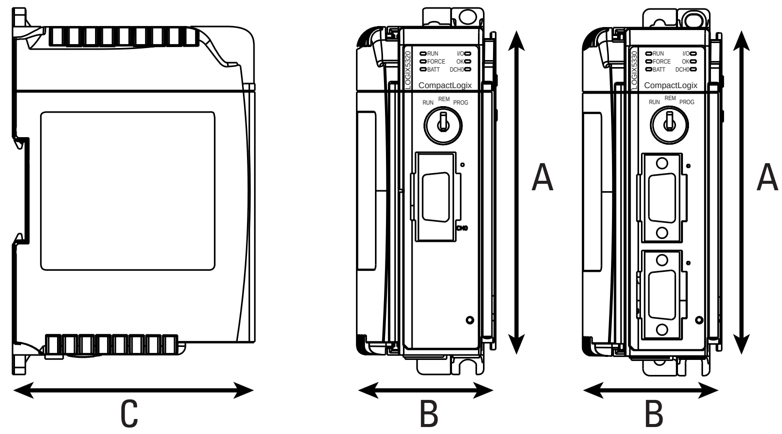
Battery	1747-BA containing 0.59g lithium





Dimensions

CompactLogix Modular Controller

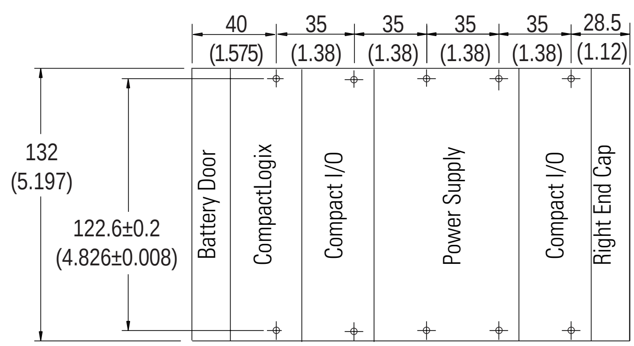


CompactLogix5320 CompactLogix5330

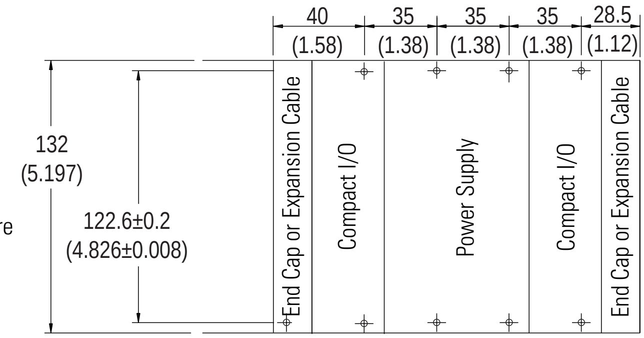
Dimension	Value
Height (A)	118 mm (4.649 in.)
Width (B)	50 mm (1.97 in.)
Depth (C)	87 mm (3.43 in.)

CompactLogix System

NOTE: All dimensions are in mm (inches). Hole spacing tolerance: ±0.4 mm (0.016 in.)



Compact I/O Expansion Power Supply and End Caps



NOTE: All dimensions are in mm (inches). Hole spacing tolerance: ±0.4 mm (0.016 in.)

TIP

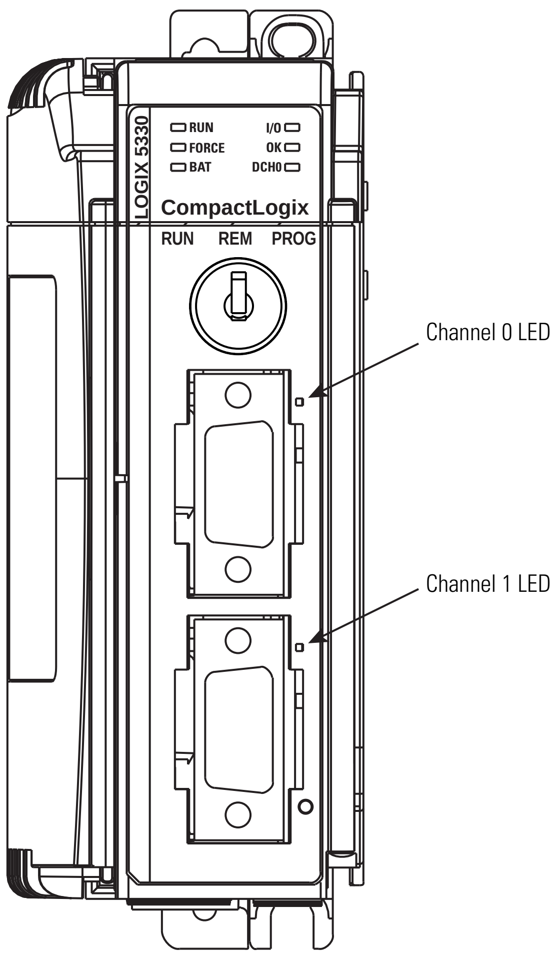
Compact I/O expansion cables have the same dimensions as the end caps.

CompactLogix Troubleshooting

Using This Appendix

For information about:	See page
CompactLogix controller LED descriptions	B-1
Identifying Controller Fault Messages	B-2
Calling Rockwell Automation for Assistance	B-3

CompactLogix Controller LEDs



The controller status LEDs provide a mechanism to determine the current status of the controller if a programming device is not present or available.

Indicator	Color/Status	Description
RUN	Off	no task(s) running; controller in Program mode
	Green	one or more tasks are running; controller is in the Run mode
FORCE	Off	no forces enabled
	Amber	forces enabled
	Amber Flashing	one or more input or output addresses have been forced to an On or Off state, but the forces have not been enabled
BAT	Off	battery supports memory
	Red	battery may not support memory, replace battery
OK	Off	no power applied
	Green	controller OK
	Red flashing	recoverable controller fault
	Red	Non-recoverable controller fault: Cycle power. The OK LED should change to flashing red. If LED remains solid red, replace the controller.
I/O	Off(1)	no activity; no I/O or communications configured
	Green	communicating to all devices
	Green flashing	one or more devices not responding
	Red flashing	not communicating to any devices controller faulted
DCH0	Off	user-configured communications active
	Green	default communications active
Channel 0	Off	no activity
	Green flickering	data is being received or transmitted
Channel 1	Off	no activity
	Green flickering	data is being received or transmitted

(1) If the controller does not contain an application (controller memory is empty), the I/O indicator will be off.

Identifying Controller Faults

Refer to the *Logix5000 Controllers Common Procedures Manual,* publication number 1756-PM001C-EN-P, for a list of controller fault messages that can occur during operation of the CompactLogix controller. Each description includes the error code, the probable cause, and the recommended corrective action.

The *Logix5000 Controllers Common Procedures Manual*, publication 1756-PM001, also contains procedures for monitoring faults and developing fault routines.

Calling Rockwell Automation for Assistance

If you need to contact Rockwell Automation or local distributor for assistance, it is helpful to obtain the following (prior to calling):

- controller type, series letter, and revision letter of the unit
- series letter, revision letter, and firmware (FRN) number of the controller (as reported by the software)
- controller LED status
- controller error codes

Notes:

Maintaining the Battery

Using this Appendix

For information about:	See page
Storing replacement batteries	C-1
Estimating battery life	C-1
Replacing batteries	C-3

Storing Replacement Batteries

Because a battery may leak potentially dangerous chemicals if stored improperly, store batteries as follows:

ATTENTION

Store batteries in a cool, dry environment. We recommend 25°C with 40% to 60% relative humidity. You may store batteries for up to 30 days between -45° to 85°C, such as during transportation. To avoid possible leakage, *do not* store batteries above 60°C for more than 30 days.

Estimating Battery Life

When the battery is about 95 percent discharged, the controller provides the following warnings:

- On the front of the controller, the BATTERY LED turns on (solid red).
- A minor fault occurs (type 10, code 10).

To estimate how long the battery will support the memory of the controller:

1. Determine the temperature (°C) 1 inch below the CompactLogix controller.

2. Determine the percentage of time that the controller is powered on per week.

EXAMPLE

If a controller is on:

- 8 hr/day during a 5-day work week
- all day Saturday and Sunday

Then the controller is on 52% of the time:

1. total hours per week = 7 x 24 = 168 hours
2. total on hours per week = (5 days x 8 hrs/day) + Saturday + Sunday = 88 hours
3. percentage on time = 88/168 = 52%

Use the on-time percentage you calculated with the following tables to determine battery life:

CompactLogix5320 - Typical Minimum Battery Life

Time ON/OFF	at 25°C (77°F)	at 40°C (104°F)	at 60°C (140°F)
Always OFF	12 months	10 months	7 months
ON 8 hours per day 5 days per week	16 months	13 months	10 months
ON 16 hours per day 5 days per week	23 months	19 months	14 months
Always ON	Not applicable(1)		

(1) There is almost no drain on the battery when the controller is always ON.

CompactLogix5330 - Typical Minimum Battery Life

Time ON/OFF	at 25°C (77°F)	at 40°C (104°F)	at 60°C (140°F)
Always OFF	67 months	29 months	11 months
ON 8 hours per day 5 days per week	87 months	38 months	14 months
ON 16 hours per day 5 days per week	127 months	56 months	20 months
Always ON	Not applicable(1)		

(1) There is almost no drain on the battery when the controller is always ON.

Battery Duration After the LED Turns On[1]

Temperature	CompactLogix5320	CompactLogix5330
60°C	9 days	8 days
25°C	14 days	20 days

(1) The battery indicator (BATTERY) warns you when the battery is low. These durations are the amounts of time the battery will retain processor memory from the time the controller is powered down after the LED first turns on.

IMPORTANT If the BATTERY LED turns on when you apply power to the controller, the battery life may be less than the tables above indicate. Some of the warning time may have been used while the controller was off and unable to turn on the BATTERY LED.

Replacing a Battery

Because the controller uses a lithium battery, you must follow specific precautions when handling or disposing of a battery.

ATTENTION The controller uses a lithium battery, which contains potentially dangerous chemicals. Before handling or disposing of a battery, review *Guidelines for Handling Lithium Batteries*, publication AG-5.4.

Follow the procedure below to replace the battery.

ATTENTION The user program will be lost when the battery is removed. Make a copy of your user program before removing and replacing the battery.

1. Save your user program.

2. Make sure the new 1747-BA battery is available and ready for installation.

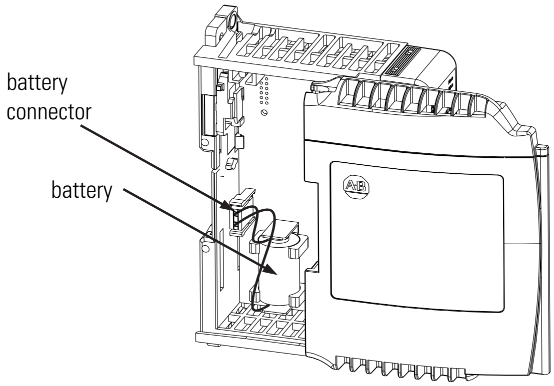
3. Turn off power to the CompactLogix controller.

4. Does the existing battery show signs of leakage or damage?

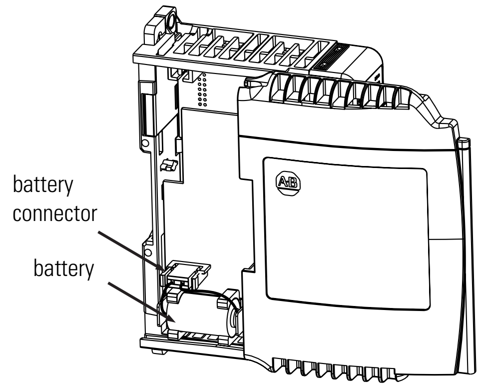
If:	Then:
Yes	Before handling the battery, review *Guidelines for Handling Lithium Batteries*, publication AG-5.4.
No	Go to the next step.

5. Remove the old battery.

CompactLogix5320



CompactLogix5330



6. Install a new 1747-BA battery. The battery connector is keyed to install only with the correct polarity.

ATTENTION

Only install a 1747-BA battery. If you install a different battery, you may damage the controller.

7. Write battery date on door of controller.

8. Apply power to the controller. On the front of the controller, is the BATTERY LED off?

If:	Then:
Yes	Go to the next step.
No	Remove power, then: A. Check that the battery is correctly connected to the controller. B. If the BATTERY LED remains on, install another 1747-BA battery. C. If the BATTERY LED remains on after you complete Step B., contact your Rockwell Automation representative or local distributor.

9. Download your user program.

10. Dispose the old battery according to state and local regulations.

ATTENTION

Do not incinerate or dispose of lithium batteries in general trash collection. They may explode or rupture violently. Follow state and local regulations for disposal of these materials. You are legally responsible for hazards created while your battery is being disposed.

Notes:

Dynamic Memory Allocation in CompactLogix Controllers

Certain operations cause the controller to dynamically allocate and de-allocate user-available memory, affecting the space available for program logic. As these functions become active, memory is allocated. Memory is then de-allocated when these functions become inactive.

The CompactLogix controller dynamically allocates memory for the following:

- Trend Objects
- Trend Drivers
- Connections

Operations that dynamically allocate memory are:

- Messages
- Connection to a Processor with RSLogix 5000
- RSLinx Tag Optimization
- Trends
- DDE/OPC Topics

Although messages are the most likely to cause dynamic memory allocation on a CompactLogix system, all the above operations are discussed in the following sections, along with general guidelines for estimating the amount of memory allocated.

Messages

Messages can come in and go out of the backplane, or come in and go out of the serial port(s), causing memory allocation as described in the table below. One simple method to reduce the affect that message instructions have on user-available memory is to prevent messages from being sent simultaneously. In general, interlocking messages in this fashion is good practice for peer-to-peer communications on networks such as DH-485 and DeviceNet.

Type		Connection Established	Dynamic Memory Allocated
Backplane	Incoming	The message is connected (connection established)	1200 bytes
		The message is unconnected (no connection established)	1200 bytes
	Outgoing	All outgoing messages whether connected or unconnnected	1200 bytes
Serial Port	Incoming	All incoming messages whether connected or unconnected	1200 bytes
	Outgoing	All outgoing messages whether connected or unconnected	1200 bytes

RSLinx Tag Optimization

Tag optimization creates three items which allocate memory, a trend object, a trend driver, and a connection.

Item	Description	Memory Allocated
Trend Object	Created in the controller to group the requested tags. One trend object can handle approximately 100 tags (connection points)	80 bytes
Trend Driver	Created to communicate to the trend object	36 bytes/single point (some economy for multiple points in a driver)
Connection	Created between the controller and RSLinx	1200 bytes

EXAMPLE

To monitor 100 points:

100 points x 36 bytes = 3600 bytes (Trend Driver)

3600 (Trend Driver) + 80 (Trend Object) + 1200 (Connection)

= approximately 4000 bytes(1)

(1) In general, we estimate that one tag takes about 40 bytes of memory.

Trends

Each trend created in a controller creates a trend object and allocates a buffer for logging as shown below.

Item	Memory Allocated
Trend Object	80 bytes
Log Buffer	4000 bytes

DDE/OPC Topics

A DDE/OPC Topic uses connections based on the following three variables:

- the number of “Maximum Messaging Connections per PLC” configured in RSLinx
- whether the “Use Connections for Writes to ControlLogix processor” is checked
- the number of connections needed to optimize throughput

IMPORTANT These variables are per path. For example, if you set up two different DDE/OPC topics, with different paths to the same controller, the variables limit the connections for each path. Therefore, if you have a limit of 5 connections, it is possible to have 10 connections, with 5 over each path.

Maximum Messaging Connections per PLC

This variable is configured in RSLinx under the “Communications” menu item “Configure CIP Options”. This number limits the number of read connections made to Logix controllers from a particular workstation.

Checking "Use Connections for Writes to ControlLogix Controller"

This variable is configured in RSLinx under the "Communications" menu item "Configure CIP Options". This check box indicates whether you want RSLinx to open up additional connections for writing data to a Logix controller.

TIP There is no way to limit the number of write connections, once this box is checked.

Number of Connections Needed to Optimize Throughput

RSLinx only opens the number of connections required to optimize throughput. For example, if you have 1 tag on scan, but have configured RSLinx to allow five connections as the maximum number of connections, RSLinx only opens one connection for the tag. Conversely, if you have thousands of tags on scan and limit the maximum number of CIP connections to five, that is the maximum number of connections that RSLinx establishes to the CompactLogix controller. RSLinx then funnels all of the tags through those five available connections.

Viewing the Number of Open Connections

You can see how many connections are made from your workstation to the CompactLogix controller in RSLinx by selecting "CIP Diagnostics" from the "Connections" menu. The Dispatching tab contains various CIP information, including the number of connections open to the CompactLogix controller.

Numerics

1756-BA1 C-1
1761-NET-DNI 6-13
1761-NET-ENI 7-3
1769-ADN 6-3
1769-SDN 6-5, 6-9

A

adding local input modules 1-7
alias
- defining 3-13
- getting started 1-12

ASCII protocol 4-16

B

backplane memory use 3-4
battery A-2, C-1

C

cables
- 1769 expansion 3-1
- connecting ASCII devices 4-17
- connecting serial devices 4-6
- connecting to 1761-NET-AIC 5-3
- DH-485 link cable length 5-1, 5-7
- multiple DH-485 connection 5-8
- programming cable recommendations A-1
- selecting serial cable 4-5
- serial cable length 4-3
- single DH-485 connection 5-8

changing
- module properties 1-9
- project properties 1-4

Channel 0 Default Communication push button 4-2
communicating
- DH-485 5-1
- serial 4-1

communication driver
- serial 4-10

communication format 3-8
CompactBus
- configuring 3-5
- inhibiting 3-6
- RPI 3-6

configuring
- 1769-ADN 6-3
- 1769-scanlist 6-5
- alias 3-13
- ASCII protocol 4-16
- communication format 3-8
- CompactBus 3-5
- DeviceNet system 6-1
- DF1 master 4-14
- DF1 point-to-point 4-11
- DF1 slave 4-14
- DH-485 system 5-2
- Ethernet system 7-1
- generic profile 3-17
- inhibit I/O module 3-9
- local I/O 3-7
- response to connection failure 3-11
- serial system 4-3
- system overhead 2-6

connection
- I/O module 3-13
- response to failure 3-11

controller
- LED status B-1
- ownership 3-8

creating
- project 1-3
- tags 1-11

current consumption 3-3

D

data 3-11
DDE/OPC topics D-3
developing programs 2-2
DeviceNet
- 1769-SDN scanlist 6-5
- commissioning 1761-NET-DNI modulles 6-13
- configure hardware 6-13
- configuring 1769-ADN 6-3
- configuring the system 6-1
- downloading to 1769-SDN 6-9
- example controlling devices 6-2
- module command array 6-9
- sending messages 6-13
- transferring data 6-8

DF1 protocol
master 4-9, 4-14
master/slave methods 4-12
point-to-point 4-9, 4-11
slave 4-9, 4-14
DH-485
cables 5-1, 5-7
configuring the system 5-2
connecting 1761-NET-AIC 5-3
hardware 5-2
installing 5-7
MSG example 5-9
network initialization 5-6
nodes 5-6
overview 5-1
token rotation 5-5
dimensions A-3
documenting I/O 1-12
downloading
project 1-15

E

end cap 3-16
entering logic 1-13
Ethernet
configuring 1761-NET-ENI 7-3
configuring system 7-1
driver 7-8
example sending messages 7-9
example staggering multiple messages 7-12
hardware 7-1
IP addreses 7-2
mapping data table files 7-7
expansion cables
configuration 3-1
dimensions A-4

F

fault data 3-14

G

generic profile 3-17
getting started
adding a local input module 1-7
changing module properties 1-9
changing project properties 1-4
creating a project 1-3
creating tags 1-11
documenting I/O with alias tags 1-12
downloading a project 1-15
entering logic 1-13
overview 1-1
steps 1-2
viewing controller memory usage 1-17
viewing I/O tags 1-10
viewing scan time 1-16
grounding
DH-485 network 5-9
serial network 4-6

H

hardware
DeviceNet 6-13
DH-485 5-2
Ethernet 7-1
serial 4-4

I

I/O module
alias 3-13
backplane memory use 3-4
communication format 3-8
CompactBus 3-5
configuring local 3-7
connection 3-13
end cap detection 3-16
fault data 3-14
generic profile 3-17
local overview 3-1
monitoring 3-14
power consumption 3-3
inhibit operation 3-9
CompactBus 3-6
input module
adding 1-7
IP addresses 7-2

L

LED status B-1
local I/O
  backplane memory use 3-4
  CompactBus 3-5
  configuring 3-7
  generic profile 3-17
  overview 3-1
  placing 3-1
  power consumption 3-3
logic
  entering 1-13
Logix environment 2-1
low battery C-1

M

mapping data table files 7-7
master/slave communication 4-12
memory
  backplane 3-4
  controller usage 1-17
memory allocation D-1
messages D-2
monitoring
  I/O module 3-14

P

placing
  local I/O 3-1
power budgeting 3-3
power supply
  current capacity 3-3
priority 2-3
program
  defining 2-5
  developing 2-2
programming
  inhibiting a module 3-10
project
  creating 1-3
  developing 2-2
  downloading 1-15
  program 2-5
  properties 1-4
  routine 2-5
  task 2-3

R

replacing the battery C-1
routine
  defining 2-5
RSLinx tag optimization D-2

S

scan list 6-5
scan time 1-16
serial
  ASCII protocol 4-16
  cable pinouts 4-5
  cables 4-3
  Channel 0 Default Communication push button 4-2
  communication driver 4-10
  configuring the system 4-3
  connecting ASCII devices 4-17
  connecting devices 4-6
  defaul configuration 4-1
  DF1 protocol 4-9
  hardware 4-4
  master 4-14
  overview 4-1
  point-to-point 4-11
  slave 4-14
slave/master communication 4-12
specifications A-1
staggering multiple messages 7-12
storing batteries C-1
synchronize I/O 3-5
system overhead 2-6

T

tag
  alias 3-13
  creating 1-11
  names 3-11
  sample alias 1-12
  viewing 1-10
task
  defining 2-3
  priority 2-3
trends D-3
troubleshooting B-1

V

verifying I/O layout 3-4
viewing
controller memory usage 1-17
I/O tags 1-10
scan time 1-16

How Are We Doing?

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

You can complete this form and mail it back to us, visit us online at www.ab.com/manuals, or email us at RADocumentComments@ra.rockwell.com

Pub. Title/Type CompactLogix System User Manual

Cat. No. 1769-L20, -L30 Pub. No. 1769-UM007D-EN-P Pub. Date August 2002 Part No. 957689-84

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

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

Your Name ____________ Location/Phone ____________

Your Title/Function ____________ Would you like us to contact you regarding your comments?

___No, there is no need to contact me

___Yes, please call me

___Yes, please email me at ____________

___Yes, please contact me via ____________

Return this form to: Allen-Bradley Marketing Communications, 1 Allen-Bradley Dr., Mayfield Hts., OH 44124-9705

Phone: 440-646-3176 Fax: 440-646-3525 Email: RADocumentComments@ra.rockwell.com

PLEASE FASTEN HERE (DO NOT STAPLE)

Other Comments

PLEASE FOLD HERE

PLEASE REMOVE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 18235 CLEVELAND OH

POSTAGE WILL BE PAID BY THE ADDRESSEE

Allen-Bradley RELIANCE ELECTRIC DODGE ROCKWELL SOFTWARE

Rockwell Automation

1 ALLEN-BRADLEY DR
MAYFIELD HEIGHTS OH 44124-9705

Reach us now at www.rockwellautomation.com

Wherever you need us, Rockwell Automation brings together leading brands in industrial automation including Allen-Bradley controls, Reliance Electric power transmission products, Dodge mechanical power transmission components, and Rockwell Software. Rockwell Automation's unique, flexible approach to helping customers achieve a competitive advantage is supported by thousands of authorized partners, distributors and system integrators around the world.

Allen-Bradley
RELIANCE ELECTRIC
DODGE
ROCKWELL SOFTWARE
Rockwell Automation

Americas Headquarters, 1201 South Second Street, Milwaukee, WI 53204, USA, Tel: (1) 414 382-2000, Fax: (1) 414 382-4444
European Headquarters SA/NV, avenue Herrmann Debroux, 46, 1160 Brussels, Belgium, Tel: (32) 2 663 06 00, Fax: (32) 2 663 06 40
Asia Pacific Headquarters, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication 1769-UM007D-EN-P - August 2002 PN 957689-84
Supersedes Publication 1769-UM007C-EN-P - June 2001

Allen-Bradley

CompactLogix™ System

User Manual