Allen-Bradley

EtherNet/IP Modules in Logix5000 Control Systems

1756-ENBT, 1788-ENBT, 1769-L32E, 1768-ENBT, 1794-AENT, 1734-AENT

User Manual

Rockwell Automation

Important User Information

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

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

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

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

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

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

WARNING

Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.

IMPORTANT

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

ATTENTION

Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you:

- identify a hazard
- avoid a hazard
- recognize the consequence

SHOCK HAZARD

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

BURN HAZARD

Labels may be located on or inside the drive to alert people that surfaces may be dangerous temperatures.

This document describes how to use EtherNet/IP modules in Logix5000 control systems. Revision bars in the margin identify updated information. This version of the document adds the 1768-ENBT CompactLogix EtherNet/IP Communication Module.

For This Information	See Page
1768-ENBT Overview	1-3
Module LED Indicators	A-1
EtherNet/IP Network Connections	B-1

Notes:

Notes:

About the Logix5000 Communication Modules for EtherNet/IP Networks

Use This Chapter

The Logix5000 family offers several EtherNet/IP communication modules. Select the module you need based on the EtherNet/IP functions you need.

EtherNet/IP Module	Works With a Controller to Originate Communication (Scanner/bridge)	Interfaces With Distributed I/O Modules (Adapter)
1756-ENBT	X	X
1756-EWEB[1]	X	
1769-L32E, 1769-L35E	X	
1768-ENBT	X	
1788-ENBT	X	
1794-AENT		X
1734-AENT		X

[1] For more information about the 1756-EWEB module, see the EtherNet/IP Web Server Module User Manual, publication ENET-UM527.

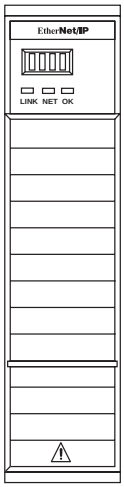
The EtherNet/IP communication modules:

- Support messaging, produced/consumed tags, and distributed I/O.
- Encapsulate messages within standard TCP/UDP/IP protocol.
- Share a common application layer with ControlNet and DeviceNet protocols.
- Interface via RJ45, category 5, unshielded, twisted-pair cable.
- Support half/full duplex 10 Mbps or 100 Mbps operation.
- Require no network scheduling.
- Require no routing tables.

This chapter introduces the modules listed above and describes how you can use these modules in a control system:

The remaining chapters in this publication describe how to configure and program the EtherNet/IP communication modules. A listing of catalog numbers at the beginning of each chapter identifies the modules that support the feature described in that chapter.

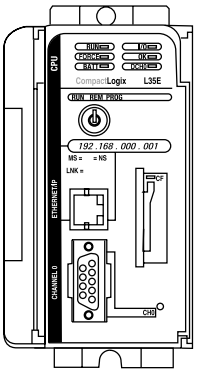
1756-ENBT Overview



The 1756-ENBT module operates either as an interface for a ControlLogix controller to communicate with other devices over an EtherNet/IP network or as an adapter for 1756 I/O modules on an EtherNet/IP network. This module supports:

- Control of I/O
- Communication via produced/consumed tags and MSG instructions
- Communication with HMI
- Configuration and programming, such as upload, download
- Adapter functionality for 1756 I/O modules
- A web server to provide diagnostic and status information

1769-L32E, 1769-L35E Overview

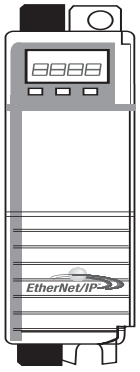


The 1769-L32E and 1768-L35E CompactLogix controllers have an integrated EtherNet/IP port. Through this port, the controller supports:

- Control of I/O
- Communication via produced/consumed tags and MSG instructions
- Communication with HMI
- Configuration and programming, such as upload, download
- A web server to provide diagnostic and status information

For more information, see the CompactLogix System Manual, publication 1769-UM011.

1768-ENBT Overview



The 1768-ENBT module is an interface that lets a CompactLogix controller (1768-L43 or 1768-L45) communicate with devices over an EtherNet/IP network. The module supports:

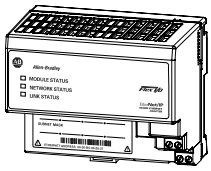
- Control of I/O
- Communication via produced/consumed tags and MSG instructions
- Communication with HMI
- Configuration and programming, such as upload, download
- A web server to provide diagnostic and status information

1788-ENBT Overview

The 1788-ENBT module operates as an interface for a FlexLogix and DriveLogix controller to communicate with other devices over an EtherNet/IP network. This module supports:

- Control of I/O
- Communication via produced/consumed tags and MSG instructions
- Communication with HMI
- Configuration and programming, such as upload, download
- A web server to provide diagnostic and status information

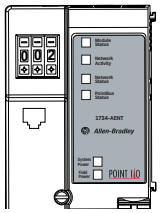
1794-AENT Overview



The 1794-AENT module operates as an adapter for FLEX I/O modules on an EtherNet/IP network. This module supports:

- Control of I/O
- Configuration
- A web server to provide diagnostic and status information

1734-AENT Overview



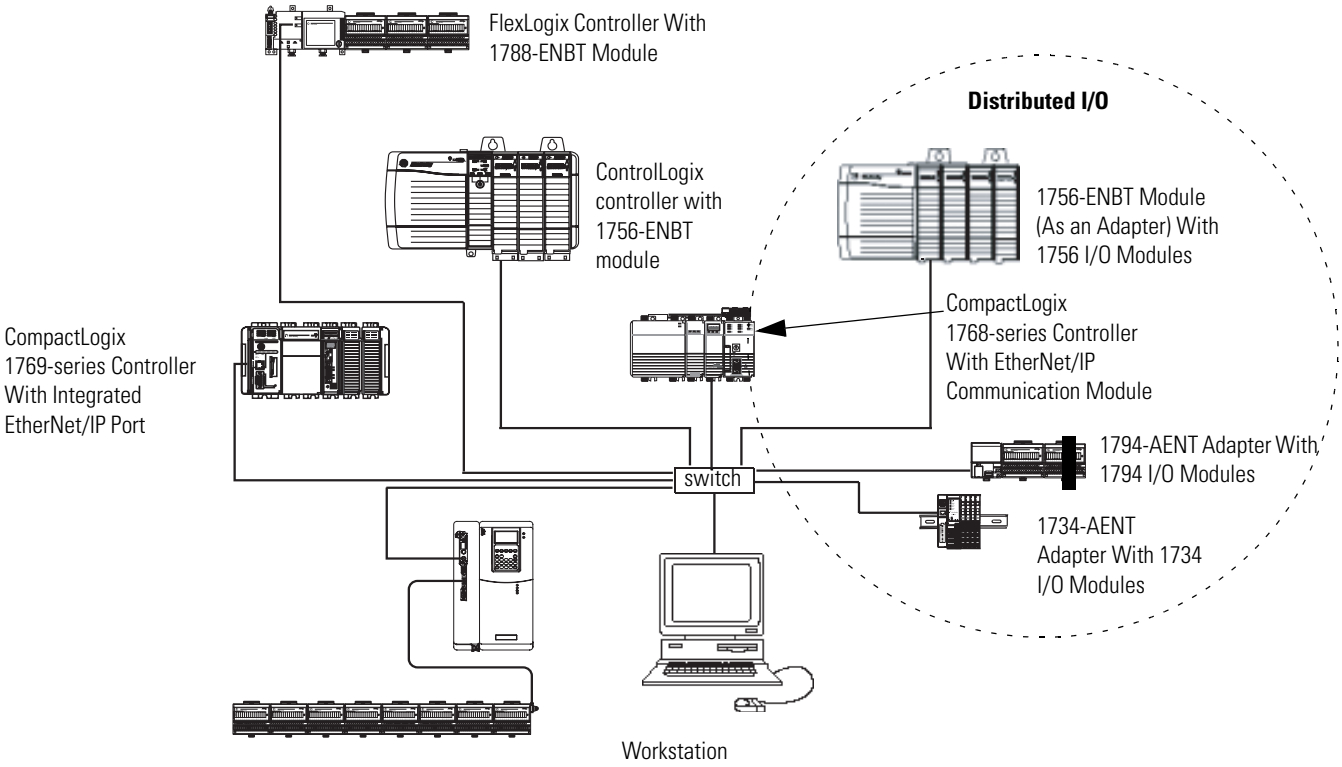
The 1734-AENT module operates as an adapter for POINT I/O modules on an EtherNet/IP network. This module supports:

- Control of I/O
- Configuration
- A web server to provide diagnostic and status information

This module's configuration process varies quite a bit from the other modules described in this publication. For more information, see the 1734-AENT User Manual, publication 1734-UM011.

Use the EtherNet/IP Communication Modules in a Control System

This diagram shows how EtherNet/IP modules can fit into a control system.



In this example:

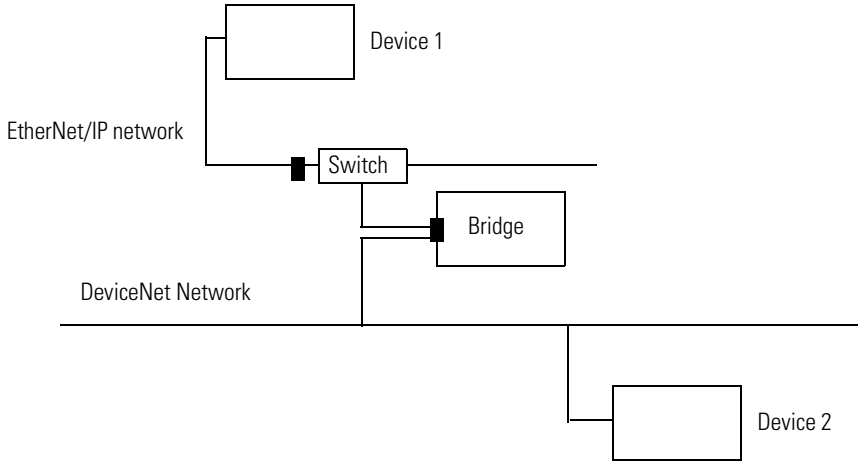
- The controllers can produce and consume tags with each other.
- The controllers can initiate MSG instructions that send/receive data or configure devices.
- The personal computer can upload/download projects to the controllers.
- The personal computer can configure devices on the EtherNet/IP network.

Bridge Across Networks

Some EtherNet/IP modules support the ability to bridge or route communication through devices, depending on the capabilities of the platform and communication devices.

IMPORTANT The update time of local I/O modules may increase when bridging messages.

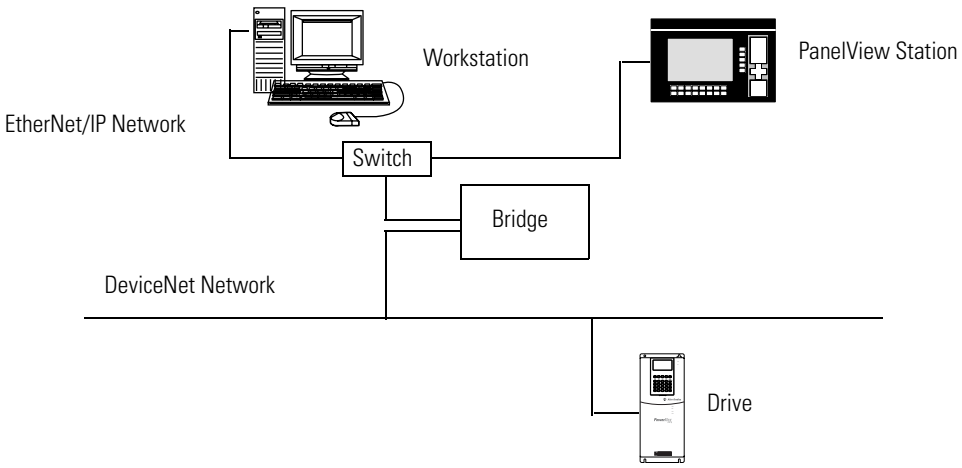
You have a bridge when you have a connection between communication devices on two separate networks. For example, the bridge device shown below has both EtherNet/IP and DeviceNet connections so that Device 1 on the EtherNet/IP network can communicate with Device 2 on DeviceNet through the bridge.



CIP messages can bridge these networks:.

CIP Messages That Originate on This Network	Can Bridge to This Network			
	EtherNet/IP	ControlNet	DeviceNet	RS-232 serial
EtherNet/IP	yes	yes	yes	yes
ControlNet	yes	yes	yes	yes
RS-232	yes	yes	yes	yes

In this example, a workstation configures a drive on a DeviceNet network. The workstation bridges EtherNet/IP networks to reach the drive.



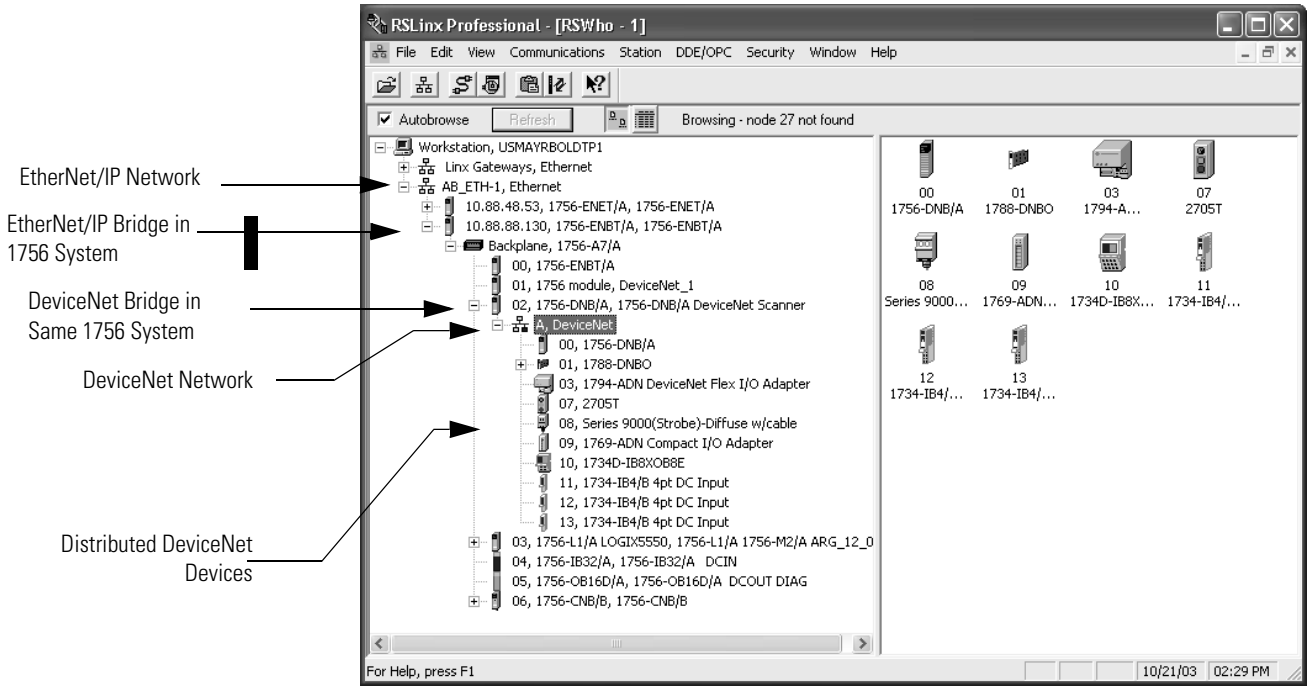
In this example, the bridge can be an EtherNet/IP to DeviceNet bridging device or a Logix5000 system with an EtherNet/IP communication module and a DeviceNet communication module. The bridge can be a:

- ControlLogix chassis with a 1756-ENBT module and a 1756-DNB module. The controller is not required.
- 1769-L32E, 1769-L35E, and 1768-L43 CompactLogix controller with a 1769-SDN module.
- FlexLogix controller with 1788-ENBT and 1788-DNBO modules.
- 1788-EN2DN linking device.

In the example above, status data can also be transferred from DeviceNet through the Logix5000 controller to a RSView32 operator interface. For a CompactLogix or FlexLogix controller, map the data into the DeviceNet I/O image and then use RSLinx OPC from the PC to the Logix5000 controller over the EtherNet/IP network. This avoids using the limited bridging resources of the CompactLogix or FlexLogix controller.

You cannot bridge EtherNet/IP I/O across networks. I/O modules must be configured in either a local chassis or a remote chassis. You cannot go through a gateway chassis to control I/O, even though in some circumstances, RSLogix 5000 software accepts such a configuration in the I/O Configuration folder.

This example RSLinx software screen shows how the DeviceNet bridge links to the EtherNet/IP network.



Configure a Personal Computer to Operate on an EtherNet/IP Network

Use This Chapter

Read this chapter for:

- *1756-ENBT module*
- *1769-L32E, -L35E controller*
- *1768-ENBT module*
- *1788-ENBT card*
- *1794-AENT adapter*

This chapter describes how to configure a personal computer to operate on an EtherNet/IP network.

You need to load an Ethernet communication driver for all Rockwell Software applications to communicate with devices on an EtherNet/IP network. A personal computer needs this driver if you use the personal computer to perform such tasks as:

- Upload and download controller projects over the EtherNet/IP network via RSLogix 5000 programming software
- Configure EtherNet/IP network parameters for devices on the network via RSNetWorx for EtherNet/IP software
- Collect controller data for PanelView terminals and RSView applications

Before you load a communication driver, make sure the

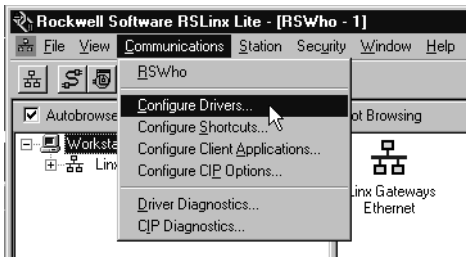
- Ethernet communication card is already installed in the personal computer.
- IP address and other network parameters are correctly configured for the personal computer.
- Personal computer is properly connected to the EtherNet/IP network.

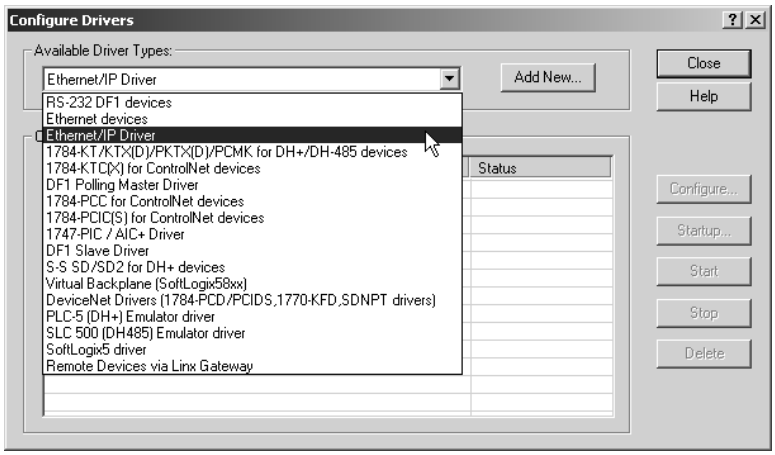
See the documentation for the appropriate Ethernet communication card for information on installing and configuring the card.

Configure the Ethernet Communication Driver in RSLinx Software

To configure the Ethernet communication driver for the personal computer (programming workstation):

1. In RSLinx software, select Configure Drivers. Select EtherNet/IP Driver or Ethernet devices.

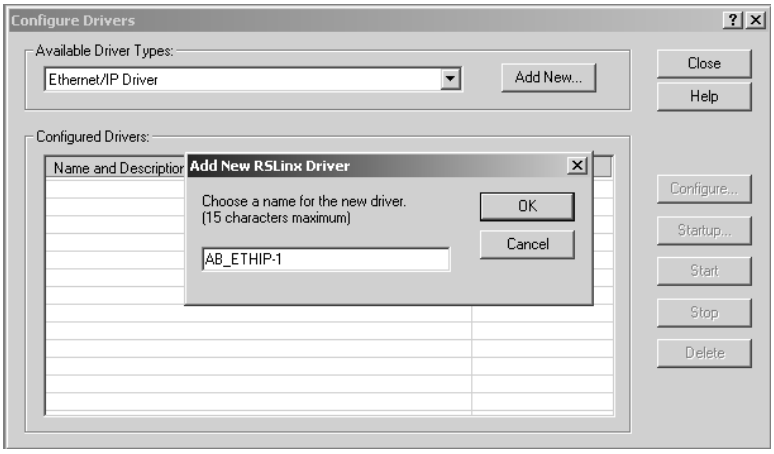




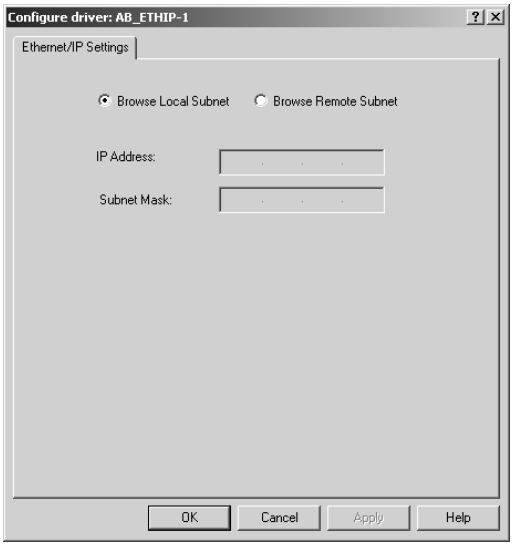
This example shows the EtherNet/IP Driver selection because it lets you autobrowse to select the appropriate device. If you select the Ethernet Devices selection, you have to enter the IP address of the device. See RSLinx online help for more information.

2. Click Add New to add the driver.

Enter a name for the driver. →

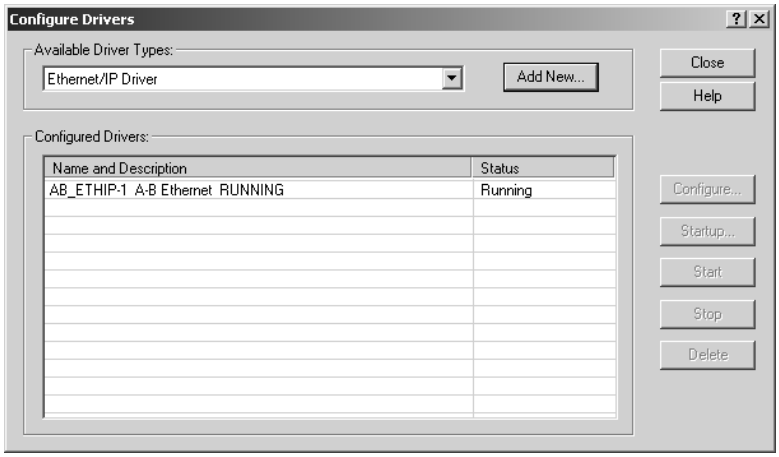


3. Select Browse Local Subnet. This displays the devices on the local network so you can navigate to the EtherNet/IP communication module for the controller you want to program.



After you navigate to the appropriate EtherNet/IP communication module, click OK.

4. The driver is now available and you can select the Ethernet port from Who Active in RSLogix 5000 programming software.



Notes:

Configure an EtherNet/IP Module to Operate on the Network

Use This Chapter

Read this chapter for:

- *1756-ENBT module*
- *1769-L32E, -L35E controller*
- *1768-ENBT module*
- *1788-ENBT card*
- *1794-AENT adapter*

This chapter describes how to configure an EtherNet/IP communication module to operate on an EtherNet/IP network.

When you first install a Rockwell Automation EtherNet/IP module (right out of the box), the module is BOOTP/DHCP enabled.

Determine Required Network Parameters

To operate on an EtherNet/IP network, you must define these parameters:

EtherNet/IP Network Parameter	Description
IP address	The IP address uniquely identifies the module. The IP address is in the form xxx.xxx.xxx.xxx where each xxx is a number between 0-255. These are reserved values you cannot use: • 127.0.0.1 • 0.0.0.0 • 255.255.255.255
Subnet mask	Subnet addressing is an extension of the IP address scheme that allows a site to use a single network ID for multiple physical networks. Routing outside of the site continues by dividing the IP address into a net ID and a host ID via the class. Inside a site, the subnet mask is used to redivide the IP address into a custom network ID portion and host ID portion. This field is set to 0.0.0.0 by default. If you change the subnet mask of an already-configured module, you must cycle power to the module for the change to take effect.
Gateway	A gateway connects individual physical networks into a system of networks. When a node needs to communicate with a node on another network, a gateway transfers the data between the two networks. This field is set to 0.0.0.0 by default.

If you use DNS addressing, or reference the module via host name in MSG instructions, define these parameters:

EtherNet/IP Network Parameter	Description
Host name	A host name is part of a text address that identifies the host for a module. The full text address of a module is *host_name.domain_name*.
Domain name	A domain name is part of a text address that identifies the domain in which the module resides. The full text address of a module is *host_name.domain_name*. The domain name has a 48-character limit. If you specify a DNS server, you must enter a domain name. Also, if you send email from the module, some mail relay servers require a domain name be provided during the initial handshake of the SMTP session.
Primary DNS server address	This identifies the DNS server(s), if used in the network. You must have a DNS server configured if you specified a domain name or a host name in the module's configuration. The DNS server converts the domain name or host name to an IP address that can be used by the network. The 1756-ENBT requires a DNS server address. For more information on DNS addressing, see page 3-10.
Secondary DNS server address	

Check with your Ethernet network administrator to determine if you need to specify all of the above parameters.

To configure these network parameters, the recommended method is to use the Rockwell Automation BOOTP/DHCP utility (see page 3-3). If this utility is not available, there are other methods you can use (see page 3-5).

Assign Network Parameters via the BOOTP/DHCP Utility

By default, the EtherNet/IP module is BOOTP enabled. The BOOTP/DHCP utility is a standalone program that is located in the:

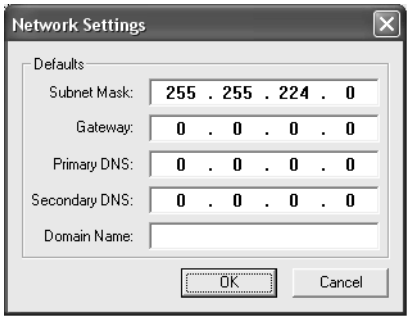
- BOOTP-DHCP Server folder in the Rockwell Software program folder on the Start menu (the utility is automatically installed when you install RSLinx software).
- Tools directory on the RSLogix 5000 installation CD.

IMPORTANT Before you start the BOOTP/DHCP utility, make sure you have the hardware (MAC) address of the module. The hardware address is on a sticker located on the side of the EtherNet/IP module. The hardware address in a format similar to: 00-0b-db-14-55-35.

This utility recognizes BOOTP-enabled devices and provides an interface to configure a static IP address for each device.

To use the BOOTP/DHCP utility:

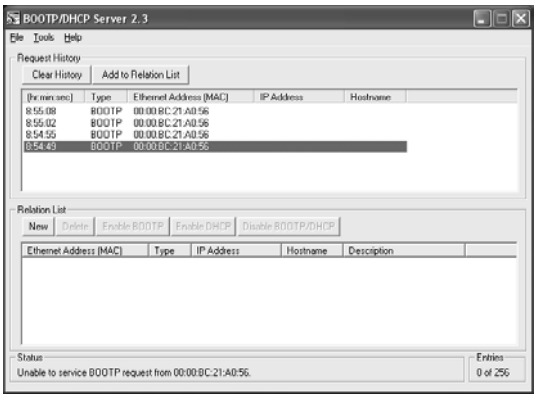
1. Start the BOOTP/DHCP software.
2. Select Tool → Network Settings.



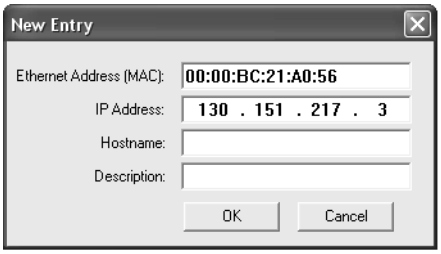
If appropriate for your network, enter the subnet mask, gateway address, primary/secondary server addresses, and domain name. Click OK.

3. In the Request History panel you see the hardware addresses of modules issuing BOOTP requests. Double-click on the hardware (MAC) address of the module you want to configure.

   The hardware address is on a sticker located on the side of the EtherNet/IP module. The hardware address will be in a format similar to 00-0b-db-14-55-35.



4. The New Entry window appears with the module's Ethernet Address (MAC).



   Enter the IP address or the host name. You can also enter a description of the module. Click OK

5. To permanently assign this configuration to the module, highlight the module and click on the Disable BOOTP/DHCP button. When power is recycled, the module uses the configuration you assigned and not issue a BOOTP request.

   If you do not select the Disable BOOTP/DHCP button, on a power cycle, the host controller clears the current IP configuration and will again begin sending BOOTP requests.

Use Other Methods to Assign Network Parameters

Other methods to assign network parameters include:

If	Use This Method For Assigning Network Parameter	See Page
• A BOOTP server is not available • the EtherNet/IP module is connected to another NetLinx network	RSLinx software	3-5
• The RSLogix 5000 project is online with the controller that communicates to or through the EtherNet/IP module	RSLogix 5000 software	3-7
• DHCP is enabled (not BOOTP) for the EtherNet/IP module	DHCP software	3-7

Other considerations that might affect your choice of method include:

- Whether the network is isolated from or integrated into the plant/enterprise network
- Size of the network

  For large networks, even isolated networks, it might be more convenient and safer to use a BOOTP/DHCP server rather than RSLogix 5000 or RSLinx software. It might also offer fewer opportunities for assigning duplicate IP addresses.

- Company policies and procedures dealing with plant floor network installation and maintenance
- Level of involvement by IT personnel in plant floor network installation and maintenance
- Type of training offered to control engineers and maintenance personnel

If you use the Rockwell Automation BOOTP or DHCP server in an uplinked subnet where an enterprise DHCP server exists, a module may get an address from the enterprise server before the Rockwell Automation utility even sees the module. You might have to disconnect from the uplink to set the address and have the module remember its static address before reconnecting to the uplink. This is not a problem if you have node names configured in the module and leave DHCP enabled.

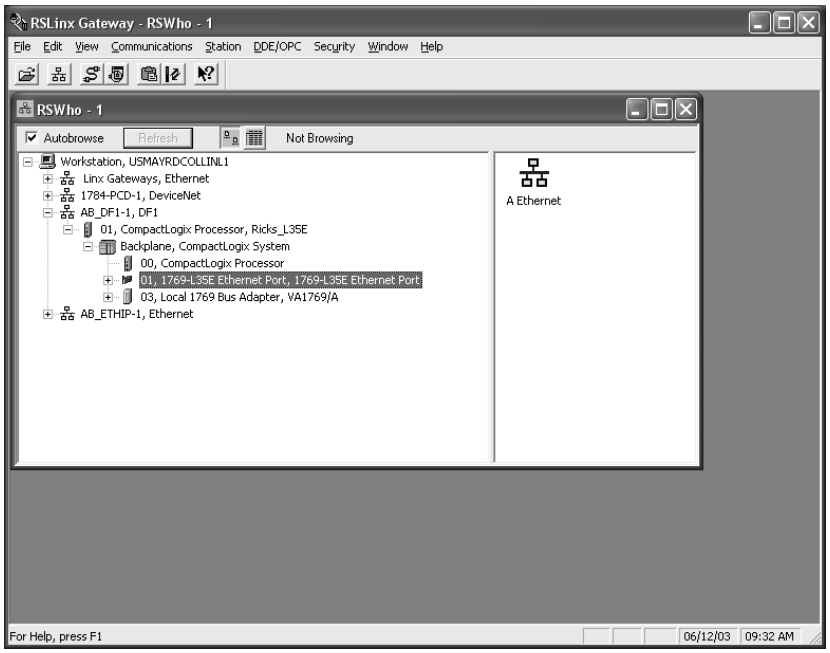
Using RSLinx software to set the IP address

To use RSLinx to configure the EtherNet/IP module:

1. Make sure the module is installed and powered up.

2. Start RSLinx. The RSWho window opens. Navigate in RSWho to the Ethernet network.

3. Right-click on the EtherNet/IP module (not the controller, if there is one) and select Module Configuration.



4. Select the Port Configuration tab, choose Status Network Configuration type, and enter the IP address and the other network parameters, if needed.

   Also, select the Static radio button to permanently assign this configuration to the port. If you select Dynamic, on a power cycle, the controller clears the current IP configuration and will again begin sending BOOTP requests.

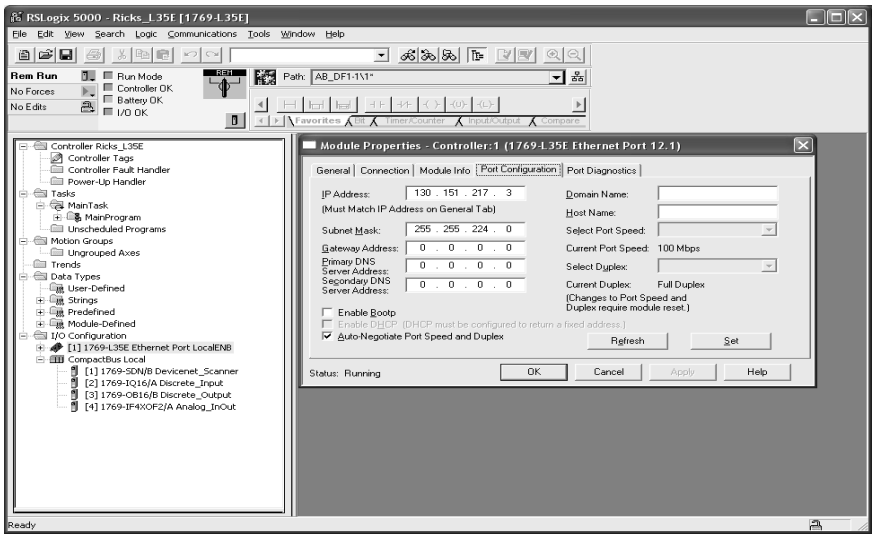
Use RSLogix 5000 software to set the IP address

To use RSLogix 5000 software to configure the EtherNet/IP module:

1. Make sure the module is installed and has power.

2. Connect to the controller via a serial, or other network, connection.

3. Start RSLogix 5000 software. In the Controller Organizer, select properties for the EtherNet/IP module.



4. Select the Port Configuration tab and specify the IP address and the other network parameters, if needed. Click Apply and then click OK.

   This sets the IP address in the hardware. This IP address should be the same IP address you assigned under the General tab.

   On this screen, you can also specify port speed (10 Mbps or 100 Mbps) and duplex mode (autonegotiate, half duplex, or full duplex). All modules on the same subnet must be configured for the same port speed and duplex mode.

Use DHCP software to set the IP address

Dynamic Host Configuration Protocol (DHCP) software automatically assigns IP addresses to client stations logging onto a TCP/IP network. DHCP is based on BOOTP and maintains some backward compatibility. The main difference is that BOOTP allows for manual configuration (static), while DHCP allows for both static and dynamic

allocation of network addresses and configurations to newly attached modules.

Be cautious when using DHCP software to configure your module. A BOOTP client, such as the EtherNet/IP modules, can boot from a DHCP server only if the DHCP server is specifically written to also handle BOOTP queries. This is specific to the DHCP software package you use. Check with your system administrator to see if your DHCP package supports BOOTP commands and manual IP allocation.

ATTENTION

The EtherNet/IP module must be assigned a fixed network address. The IP address of this module must not be dynamically provided.

Failure to observe this precaution may result in unintended machine motion or loss of process control.

Duplicate IP Address Detection

These EtherNet/IP modules (and their future revisions) support duplicate IP address detection:

- 1756-ENBT, firmware revision 3.2 and greater
- 1768-ENBT, firmware revision 1.3.3 or greater
- 1769-L32E and 1769-L35E, firmware revision 15.01 and greater (For more information, see the CompactLogix User Manual, publication 1769-UM011.)
- 1788-ENBT, firmware revision 2.1 and greater
- 1756-EWEB, firmware revision 2.2 and greater (For more information, see the EtherNet/IP Web Server Module User Manual, publication ENET-UM527.)

When you change the IP address or connect one of these modules to an EtherNet/IP network, the module checks to make sure that the IP address assigned to this module is not the same as that for any other device already on the network. If the module determines that there is a conflict (some other device on the network already has the IP address), the EtherNet/IP port of the module goes into conflict mode, where the module's:

- OK LED blinks red.
- Network (NET) LED is solid red.
- Front display indicates the conflict (1756-ENBT only).

  The display scrolls:OK <IP_address_of_this_module> Duplicate IP <Mac_address_of_duplicate_node_detected>

  For example: OK 10.88.60.196 Duplicate IP - 00:00:BC:02:34:B4

To correct this conflict, use the instructions in this chapter to change the IP address of the module. Then cycle power to the module or reset the module (such as disconnecting the EtherNet/IP cable and reconnecting the cable).

There is also the possibility that two modules can detect a conflict simultaneously. If this occurs, remove the module that has the incorrect IP address or correct its conflict. To get the second module out of conflict mode, cycle power to the module or disconnect its EtherNet/IP cable and reconnect the cable.

Duplicate detection scenarios

The behavior of devices that are in conflict over an IP address varies depending on whether connections have been established to either of the modules and whether both modules support duplicate IP address detection.

- If both modules support duplicate IP address detection, the module that powers up first and uses the IP address, keeps the IP address. The other module will detect a conflict, give up the IP address, and enter conflict mode.
- If both modules support duplicate IP address detection and both modules power up at roughly the same time, both modules give up the IP address and enter conflict mode.
- If one module supports duplicate IP address detection and a second module does not, the second module generally keeps it IP address, regardless of which module obtains the IP address first. The module that supports duplicate IP address detection will detect the conflict and give up the IP address.

IP Address Swapping

These EtherNet/IP modules (and their future revisions) support IP address swapping in ControlLogix redundancy systems:

- 1756-ENBT, firmware revision 3.1 and greater
- 1756-EWEB, firmware revision 2.2 and greater

During a switchover in ControlLogix redundancy systems, these modules swap their IP addresses with their partner modules in the other redundant chassis.

For more information about IP address swapping, see the ControlLogix Redundancy User Manual, publication 1756-UM523.

DNS Addressing

To further qualify an address of a module, you can use DNS addressing to specify a host name for a module, which also includes specifying a domain name and DNS servers. DNS addressing lets you set up similar network structures and IP address sequences under different domains.

DNS addressing is only necessary if you refer to the module by host name, such as in path descriptions in MSG instructions.

To use DNS addressing, you must:

1. Assign a host name to the module.

   Your network administrator should be able to assign a host name. Valid host names should be IEC-1131-3 compliant.

2. Configure the module's parameters.

   In addition to the IP address, subnet mask, and gateway address, you must also configure a host name for the module, domain name, and primary/secondary DNS server addresses. In the DNS server, the host name must match the IP address of the module.

IMPORTANT Make sure the DNS enable bit is set

If you configure your module using RSLinx 2.41.00, the enable bit is cleared and DNS addressing will not work. If you configure your module using the Port Configuration tab in RSLogix 5000 software, the enable bit is set, so DNS addressing should work.

3. In RSLogix 5000 software, add the module to the I/O configuration tree and enter the host name in the General tab of the module.

   If a child module resides in the same domain as its parent module, just enter the host name. If the child module is in a different domain that its parent module, you must enter the host name and the domain name (host.domain)

You can also use DNS addressing in a module profile in the I/O controller tree or in a message path. If the domain name of the destination module is different from the source module, use a fully-qualified DNS name (hostname.domainname). For example, to send a message from ENBT1.location1.companyA to ENTB1.location2.companyA, the host names are the same, but the domains are different. If you do not enter a fully-qualified DNS name, the module appends the default domain name to the specified host name.

Use the EtherNet/IP Modules in a Logix5000 Controller Application

After you physically install an EtherNet/IP module and set its IP address, you must add the module to the Controller Organizer in an RSLogix 5000 project to establish I/O control.

You must download that project to the host controller before operation can begin. When the controller begins operation, it establishes a connection with the EtherNet/IP module. The module behaves as determined in its configuration.

For information on configuring and placing a personal computer (for developing an RSLogix 5000 project) on an EtherNet/IP network, see the chapter “Configure a Personal Computer to Operate on an EtherNet/IP Network”.

For information on controlling I/O, see the chapter “Configuring I/O”.

Notes:

Control I/O

Use This Chapter

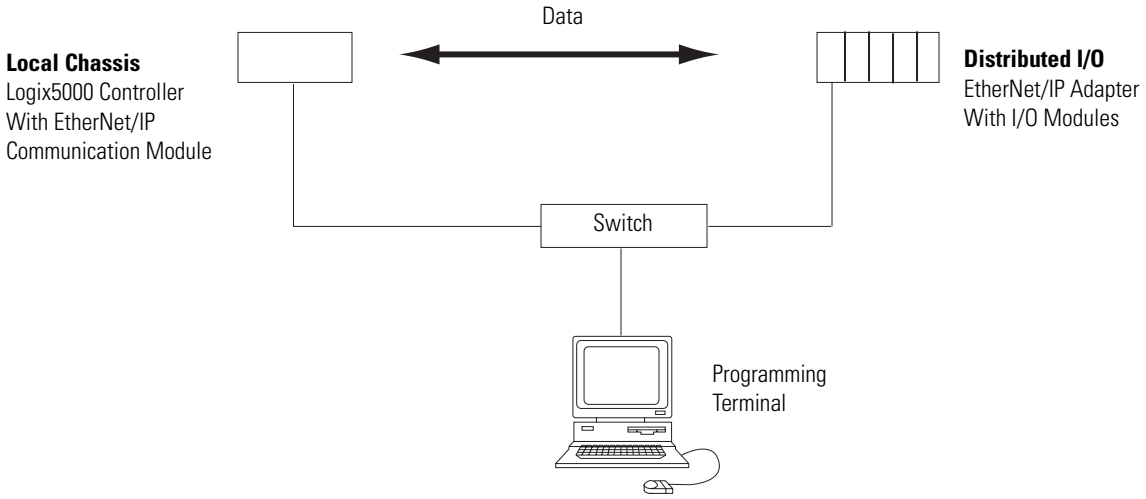
Read this chapter for:

- *1756-ENBT module*
- *1769-L32E, -L35E controller*
- *1768-ENBT module*
- *1788-ENBT card*
- *1794-AENT adapter*

This chapter describes how a controller controls distributed I/O over an EtherNet/IP network. The controller requires a communication module to connect to the network. Distributed I/O modules require an adapter to connect to the network.

Set Up the Hardware

In this example, the Logix5000 controller has an EtherNet/IP communication module to connect to the EtherNet/IP network. The distributed (remote) I/O has an EtherNet/IP adapter to connect it to the EtherNet/IP network.



The Logix5000 controller can communicate with each I/O module directly (direct connection). Or you can configure a rack-optimized connection to the EtherNet/IP adapter to send data to any digital I/O modules. Analog modules always require direct connections.

Make sure:

- The IP addresses are set for each EtherNet/IP module.
- All wiring and cabling is properly connected.
- The communication driver (such as AB-ETHIP-1) is configured for the programming workstation.

Set the Requested Packet Interval (RPI)

When you configure an I/O module, you define the requested packet interval (RPI) rate for the module.

The RPI specifies the period at which data updates over a connection. For example, an input module sends data to a controller at the RPI that you assign to the module. Configure the RPI in milliseconds.

RPIs are used only for modules that produce data. For example, a local EtherNet/IP communication module does not require an RPI because it is not a data-producing member of the system; it is used only as a bridge.

In Logix5000 controllers, I/O values update at a period that you configure via the I/O configuration folder of the project. The values update asynchronous to the execution of logic. At the specified interval, the controller updates a value independently from the execution of logic.

Set the RPI only as fast as needed by the application. The RPI also determines the number of packets per second that the module will produce on a connection. Each module has a limit of how many packets it can produce per second. If you exceed this limit, the module cannot open any more connections.

For information on RPI and how it affects the actual packet interval (API), see the EtherNet/IP Performance Application Solution, publication ENET-AP001.

Select a Communication Format

When you configure an I/O module, you select a communication format for the module. The communication format you choose determines the data structure for the tags that are associated with the module. Many I/O modules support different formats. Each format uses a different data structure. The communication format that you choose also determines:

- Direct or rack-optimized connection
- Ownership

The available communication formats depend on the type of I/O module. In general:

If You Have This Type of I/O Module	And Want	Select a Communication Format That Specifies
Digital module	A rack-optimized connection	Rack Optimization
	To use specialty features of the module, such as diagnostics, timestamps, or electronic fuses	Full Diagnostics... CST Timestamped...
Digital module	A direct connection	Scheduled Data... Input Data Output Data
Analog module	A direct connection (only direct connection is supported for analog modules)	Float Data... Integer Data CST Timestamped...

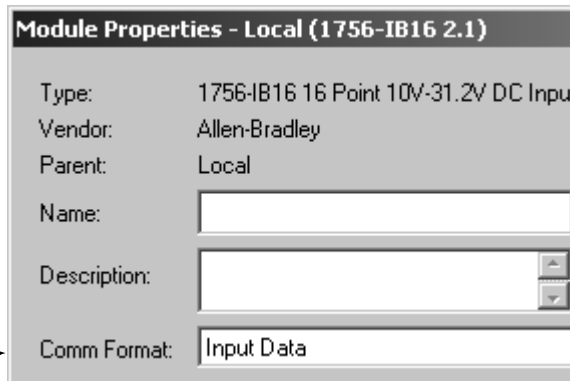
See online help in RSLogix 5000 programming software for specific communication formats per I/O module.

Choose direct or rack-optimized connection

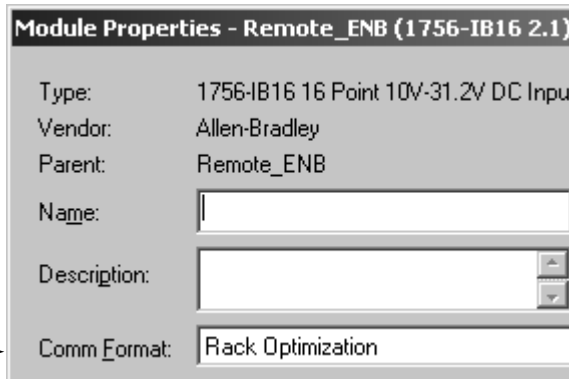
The Logix5000 controller uses connections to transmit I/O data. These connections can be direct connections or rack-optimized connections.

Term	Definition
Direct connection	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 with the I/O module. Any break in the connection, such as a module fault or the removal of a module while under power, sets fault bits in the data area associated with the module.

A direct connection is any connection that does not use the Rack Optimization Comm Format.

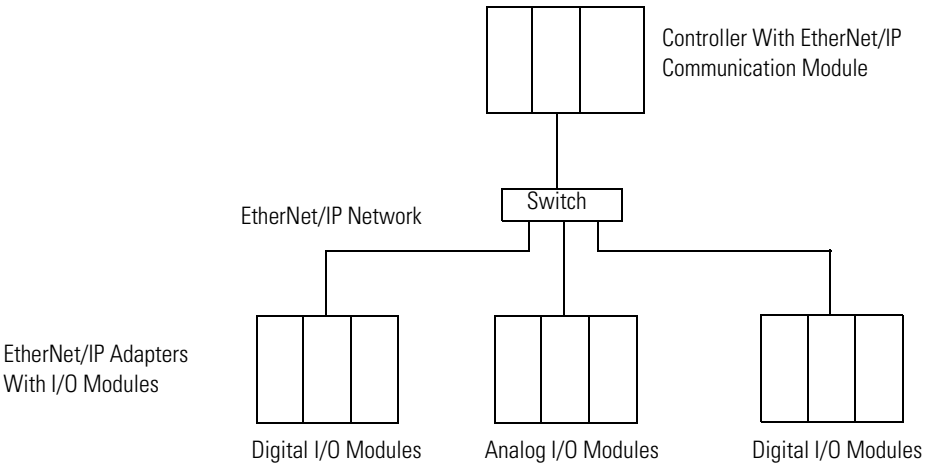


Term	Definition
Rack-optimized connection	For digital I/O modules, you can select rack-optimized communication. A rack-optimized connection consolidates connection usage between the controller and all the digital I/O modules in the chassis (or DIN rail). Rather than having individual, direct connections for each I/O module, there is one connection for the entire chassis (or DIN rail).



Direct Connections For I/O Modules

In this example, assume that each distributed I/O module is configured for a direct connection to the controller.



The following table calculates the connections in this example.

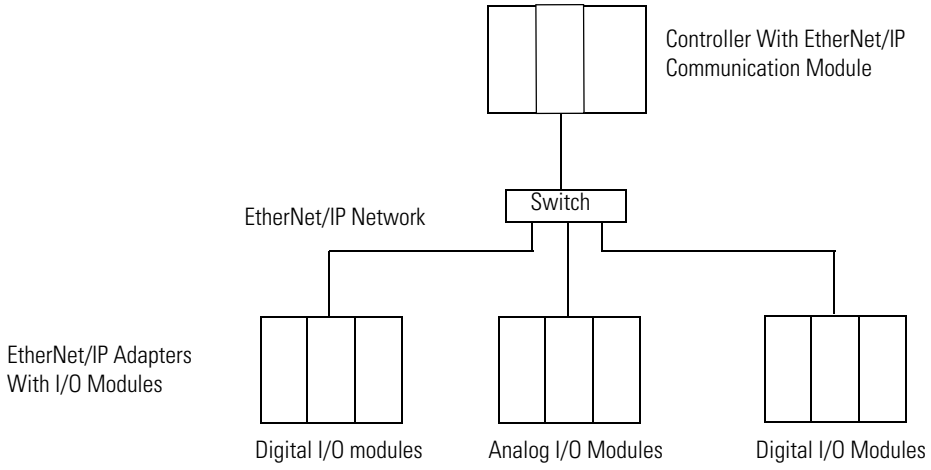
System Connections	Amount
Controller to local EtherNet/IP communication module	0
Controller to EtherNet/IP adapter	
Direct connection for digital I/O module	4
Direct connection for analog I/O module	2
Total connections used	6

If you have many modules, direct connections to each module may not be feasible because you could use up the number of connections and packets per second supported by the module.

Refer to Rack-optimized Connections For I/O Modules on page 4-6 to conserve connection use and network traffic.

Rack-optimized Connections For I/O Modules

In this example, assume that each digital I/O module is configured for a rack-optimized connection to the controller. Analog modules must be configured for direct connections.



The following table calculates the connections in this example.

System Connections	Amount
Controller to local EtherNet/IP communication module	0
Controller to EtherNet/IP adapter with digital modules (rack-optimized connection to each adapter)	2
Controller to EtherNet/IP adapter with analog modules (direct connection for each analog I/O module)	2
Total Connections used	4

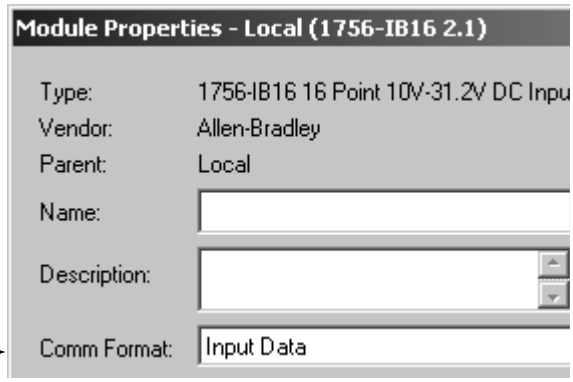
The rack-optimized connection conserves connections, but can limit the status and diagnostic information that is available from the I/O modules.

To optimize the number of available connections, use a rack-optimized connection between any digital I/O that allow it and the remote adapter that connects the distributed I/O to the controller via the communication module.

Ownership

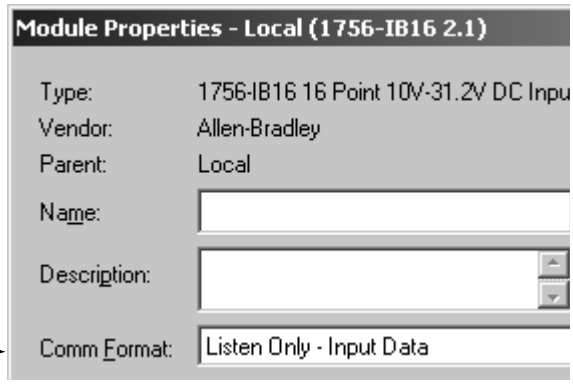
In a Logix5000 system, modules multicast data. This means that multiple modules can receive the same data at the same time from a single module. When you choose a communication format, you have to choose whether to establish an owner or listen-only relationship with the module.

Owner controller	The controller that creates the primary configuration and communication connection to a module. The owner controller writes configuration data and can establish a connection to the module.



An owner connection is any connection that does not include Listen-Only in its Comm Format.

Listen-only connection	An I/O connection where another controller owns/provides the configuration data for the I/O module. A controller using a listen-only connection only monitors the module. It does not write configuration data and can only maintain a connection to the I/O module when the owner controller is actively controlling the I/O module.



Listen-only connection

Use the following table to choose the type of ownership for a module:

If the Module Is an	And Another Controller	And You Want to	Then Use This Type of Connection
Input module	Does not own the module	→	Owner (i.e., *not* listen-only)
	Owns the module	Maintain communication with the module if it loses communication with the other controller	Owner (i.e., *not* listen-only) Use the same configuration as the other owner controller.
		Stop communication with the module if it loses communication with the other controller	Listen-only
Output module	Does not own the module	→	Owner (i.e., *not* listen-only)
	Owns the module	→	Listen-only

There is a noted difference in controlling input modules versus controlling output modules.

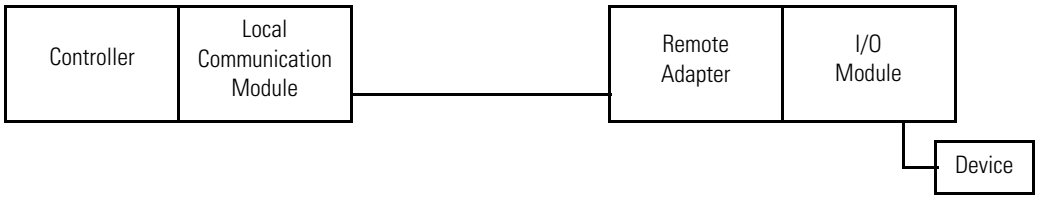
Control	This Ownership	Description
Input modules	Owner	An input module is configured by a controller that establishes a connection as an owner. This configuring controller is the first controller to establish an owner connection. Once an input module has been configured (and owned by a controller), other controllers can establish owner connections to that module. This allows additional owners to continue to receive multicast data if the original owner controller breaks its connection to the module. All other additional owners must have the identical configuration data and identical communications format that the original owner controller has, otherwise the connection attempt is rejected.
	Listen-only	Once an input module has been configured (and owned by a controller), other controllers can establish a listen-only connection to that module. These controllers can receive multicast data while another controller owns the module. If all owner controllers break their connections to the input module, all controllers with listen-only connections no longer receive multicast data.
Output modules	Owner	An output module is configured by a controller that establishes a connection as an owner. Only one owner connection is allowed for an output module. If another controller attempts to establish an owner connection, the connection attempt is rejected.
	Listen-only	Once an output module has been configured (and owned by one controller), other controllers can establish listen-only connections to that module. These controllers can receive multicast data while another controller owns the module. If the owner controller breaks its connection to the output module, all controllers with listen-only connections no longer receive multicast data.

If the module is also in the I/O configuration of another controller, select the Listen Only version of the Comm Format (for example, Listen Only - Input Data).

Add Distributed I/O

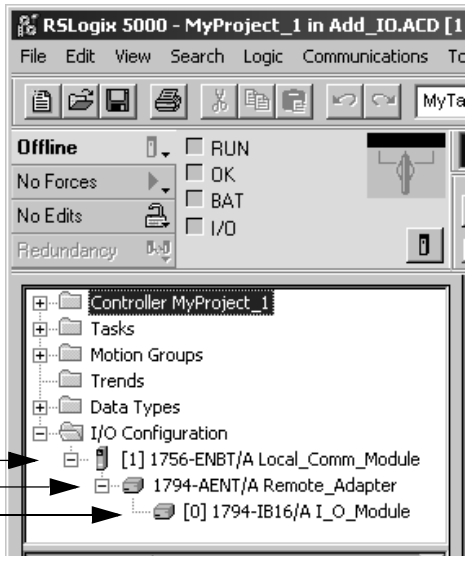
To communicate with the I/O modules in your system, you add bridge, adapter, and I/O modules to the I/O Configuration folder of the controller. Within the I/O Configuration folder, you organize the modules into a hierarchy (tree/branch, parent/child).

For a typical distributed I/O network…



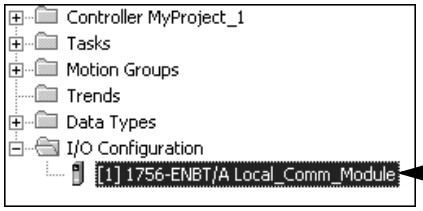
…you build the I/O configuration in this order.

1. Add the local communication module (bridge).
2. Add the remote adapter for the distributed I/O chassis or DIN rail.
3. Add the I/O module.

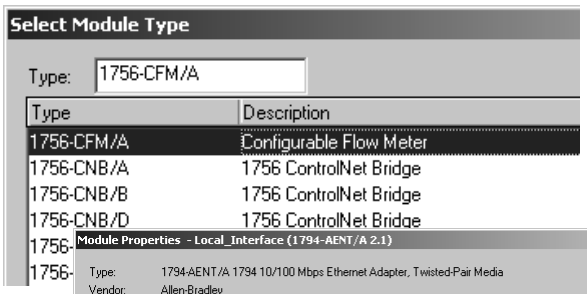


Add a Module

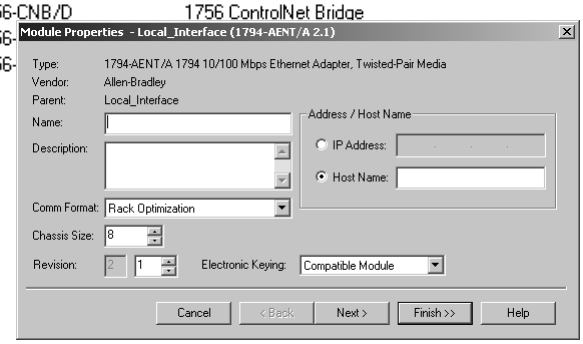
To add a module to the I/O Configuration folder:



1. Right-click the level (branch) to which you want to add the module and choose New Module.



2. Choose the module.



3. Configure the module.

To	Do This
Use the default configuration	Specify the general information about the module (name, comm format, etc.) and click Finish.
Customize the configuration	Specify the general information about the module (name, comm format, etc.). Then use the Next buttons to step through subsequent screens.

The Comm Format selection you make when you add a communication module and its I/O modules is based on whether you want rack-optimized or direct connections to each distributed I/O module. In general:

If the Distributed I/O Is	Select This Format For the Remote Adapter	Select This Format For the Distributed I/O Module
digital	Rack Optimization	Rack Optimization
analog	None	an appropriate direct-connection format

Select a Remote Adapter

The remote adapter you use depends on the distributed I/O you use

If the Distributed I/O Is	Select This Remote Adapter	Which You Configure Via
1756 ControlLogix I/O	1756-ENBT	BOOTP utility
1794 FLEX I/O	1794-AENT	BOOTP utility
1734 POINT I/O	1734-AENT	thumbwheel switches on the module or DHCP utility

Access Distributed I/O

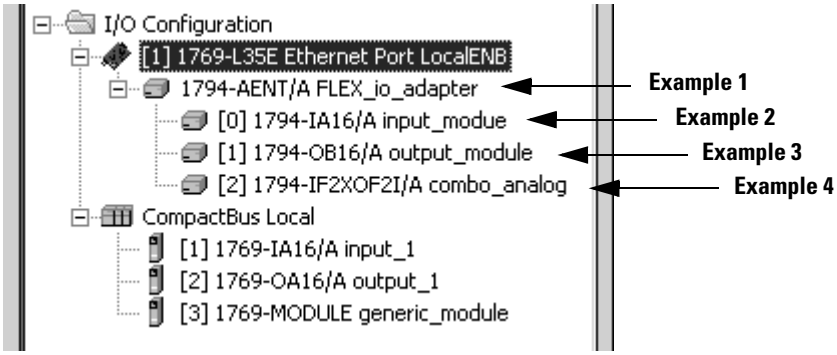
I/O information is presented as a structure of multiple fields, which depend on the specific features of the I/O module. The name of the structure is based on the location of the I/O module in the system. Each I/O tag is automatically created when you configure the I/O module through the programming software. Each tag name follows this format:

Location:SlotNumber:Type.MemberName.SubMemberName.Bit

where:

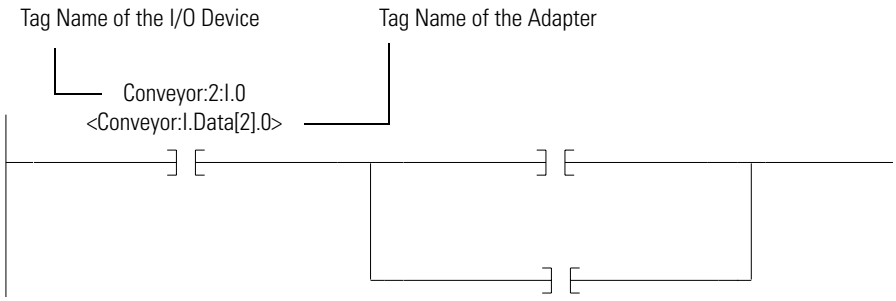
This Address Variable	Is
Location	Identifies network location LOCAL = local DIN rail or chassis ADAPTER_NAME = identifies remote adapter or bridge
SlotNumber	Slot number of I/O module in its chassis
Type	Type of data I = input O = output C = configuration S = status
MemberName	Specific data from the 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 the 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)

EXAMPLE



Example	Module	Example Tag Names (Automatically Created By the Software)
Example 1	remote 1794-AENT adapter "FLEX_io_adapter"	FLEX_io_adapter:I FLEX_io_adapter:I.SlotStatusBits FLEX_io_adapter:I.Data FLEX_io_adapter:O FLEX_io_adapter:O.Data
Example 2	remote 1794-IA16 "input_module" in slot 0 rack-optimized connection	FLEX_io_adapter:0:C FLEX_io_adapter:0:C.Config FLEX_io_adapter:0:C.DelayTime_0 FLEX_io_adapter:0:C.DelayTime_1 FLEX_io_adapter:0:C.DelayTime_2 FLEX_io_adapter:0:C.DelayTime_3 FLEX_io_adapter:0:C.DelayTime_4 FLEX_io_adapter:0:C.DelayTime_5 FLEX_io_adapter:0:I
Example 3	remote 1794-OB16 "output_module" in slot 1 rack-optimized connection	FLEX_io_adapter:1:C FLEX_io_adapter:1:C.SSData FLEX_io_adapter:1:O FLEX_io_adapter:1:O.Data
Example 4	remote 1794-IF2XOF2I "combo_analog" in slot 2 direct connection	FLEX_io_adapter:2:C FLEX_io_adapter:2:C.InputFIlter FLEX_io_adapter:2:C.InputConfiguration FLEX_io_adapter:2:C.OutputConfiguration FLEX_io_adapter:2:C.RTSInterval FLEX_io_adapter:2:C.SSCh0OuputData FLEX_io_adapter:2:C.SSCH1OutputData FLEX_io_adapter:2:I

When you choose rack optimization for an I/O module, its tags are created as aliases for the tags of the adapter module. In your logic, you see the tag of the device as aliases for a tag of the adapter module. (The tag name of the adapter is in angle brackets.)



Notes:

Interlocking and Data Transfer Between Controllers

Use This Chapter

Read this chapter for:

- *1756-ENBT module*
- *1769-L32E, -L35E controller*
- *1768-ENBT module*
- *1788-ENBT card*

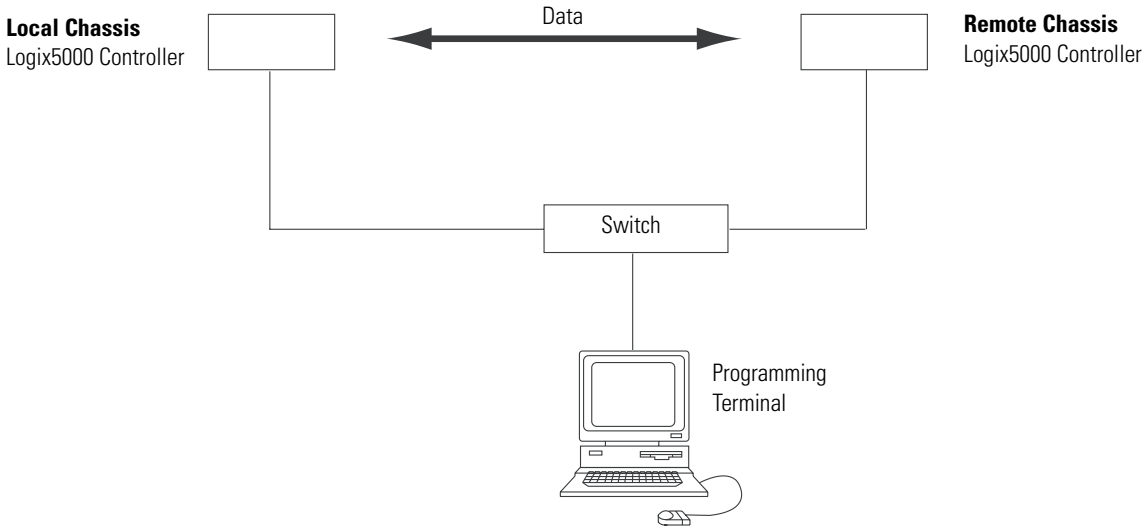
This chapter describes how to share data. You can interlock controllers (produce and consume tags). You can also send messages between controllers via an EtherNet/IP network.

There are different methods of communicating with other controllers.

If You Want To	And the Data	Then	See Page
Interlock operations	Resides on Logix5000 controllers	Produce and consume a tag	5-3
Transfer data	Needs regular delivery at an interval that you specify	Produce and consume a tag	5-3
	Is sent when a specific condition occurs in your application	Execute a message (MSG) instruction	5-9

Set Up the Hardware

In this example, the controller in the local chassis can produces a tag that is consumed by the controller in the remote chassis. The local controller can also send a MSG instruction to the remote controller.



The Logix5000 controller in the local chassis and in the remote chassis can be any of the following, with their EtherNet/IP communication modules:

- 1756 ControlLogix controller with a 1756-ENBT communication module in the chassis.
- 1768 CompactLogix controller with a 1768-ENBT communication module in the chassis.
- 1769-L35E CompactLogix controller with a built-in EtherNet/IP port.
- 1794 FlexLogix controller with a 1788-ENBT EtherNet/IP communication card.
- PowerFlex 700S with DriveLogix controller and a 1788-ENBT EtherNet/IP communication card.

Make sure that:

- the IP addresses, and other network parameters if necessary, are set for each EtherNet/IP communication module.
- all wiring and cabling is properly connected.
- the communication driver (such as AB-ETHIP-1) is configured for the programming workstation.

TIP If you are sharing tags between ControlLogix controllers and the controllers are only sharing tags (not sending messages), set the communication format of the 1756-ENBT module to None.

Organize Tags for Produced or Consumed Data

As you organize your tags for produced or consumed data (shared data), follow these guidelines.

Guideline	Details
Create the tags at the controller scope.	You can share only controller-scoped tags.
Use one of these data types: • DINT • REAL • array of DINTs or REALs • user-defined	• To share other data types, create a user-defined data type that contains the required data. • Use the same data type for the produced tag and corresponding consumed tag or tags.
Limit the size of the tag to ≤ 500 bytes.	If you must transfer more than 500 bytes, create logic to transfer the data in packets. A DINT size of < 125 bytes will keep total bytes within 500. This helps reduce the total number of packets for transactions.
Combine data that goes to the same controller.	If you are producing several tags for the same controller: • Group the data into one or more user-defined data types. (This uses less connections than producing each tag separately.) • Group the data according to similar update intervals. (To conserve network bandwidth, use a greater RPI for less critical data.) For example, you could create one tag for data that is critical and another tag for data that is not as critical.

Terminology

A Logix5000 controller lets you produce (broadcast) and consume (receive) system-shared tags.

Term	Definition
Produced tag	A tag that a controller makes available for use by other controllers. Multiple controllers can simultaneously consume (receive) the data. A produced tag sends its data to one or more consumed tags (consumers) without using logic. The produced tag sends its data at the RPI of the consuming tag.
Consumed tag	A tag that receives the data of a produced tag. The data type of the consumed tag must match the data type (including any array dimensions) of the produced tag. The RPI of the consumed tag determines the period at which the data updates.

For two controllers to share produced or consumed tags, both controllers must be attached to the same EtherNet/IP subnet. You cannot bridge produced and consumed tags over two subnets.

Determine Connections for Produced and Consumed Tags

Logix controllers can produce (broadcast) and consume (receive) system-shared tags that are sent and received via the EtherNet/IP communication module. Produced and consumed tags each require connections.

This Type of Tag	Requires These Connections
Produced	The local controller (producing) must have one connection for the produced tag and the first consumer and one more connection for each additional consumer (heartbeat). The produced tag requires two connections. As you increase the number of controllers that can consume a produced tag, you also reduce the number of connections the controller has available for other operations, like communications and I/O.
Consumed	Each consumed tag requires one connection for the controller that is consuming the tag.

All EtherNet/IP modules support as many as 32 produced connections. Additionally, the total number of tags that can be produced or consumed is limited by the number of available connections. If the communication module uses all of its connections for I/O and other communication modules, no connections are left for produced and consumed tags.

Each produced or consumed tag uses the following number of connections:

This Controller	And This Type of Tag	Use This Many Connections
ControlLogix SoftLogix5800	Produced tag	Number_of_consumers
	Consumed tag	1
CompactLogix DriveLogix FlexLogix	Produced tag	Number_of_consumers
	Consumed tag	1

TCP and CIP connection capacities vary for all EtherNet/IP modules.

Module	Packets/Second	TCP Connections	CIP Connections
1756-ENBT	5000	64	128
1769-L32E	4000	32	32
1769-L35E	4000	32	32
1768-ENBT	5000	32	64
1788-ENBT	4000	64	32
1794-AENT	9500	64	63

Produce a Tag

To produce a tag, configure the produced tag in the RSLogix 5000 project for the local (producer) controller. You do not have to configure the consumer controller(s) in the I/O Configuration folder of the producer controller.

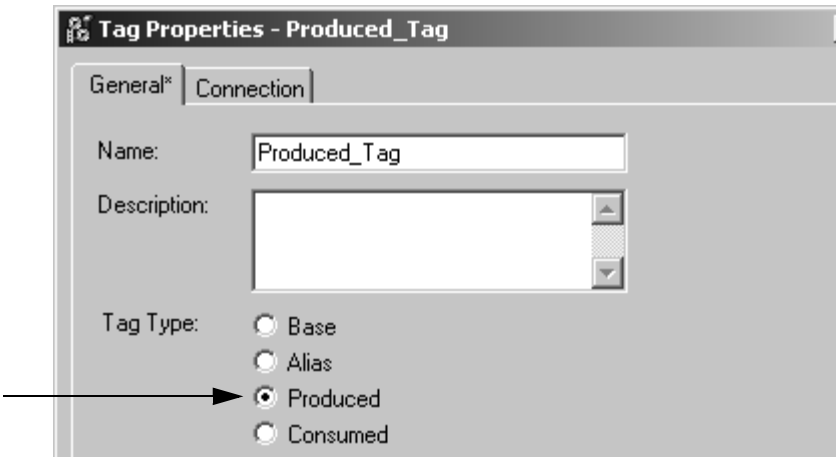
Configure the produced tag

1. In the producer's controller organizer, right-click the Controller Tags folder and select Edit Tags.

   You can produce only controller-scoped tags.

2. In the Controller Tags window, right-click the tag that you want to produce and choose Edit Tag Properties.

3. Select the Produced option button.



4. Select the Connection tab.

5. Type or select the number of controllers that will consume (receive) the tag.

6. Click OK.

Consume Data Produced by Another Controller

To consume a produced tag, you specify both the producer controller and the produced tag in the RSLogix 5000 project for the remote (consumer) Logix5000 controller.

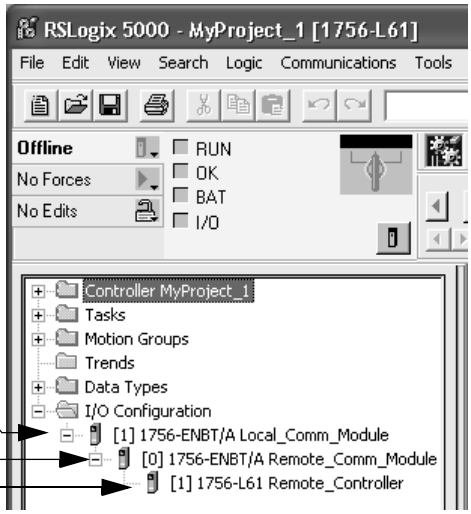
Add the Producer Controller to the Consumer's I/O Configuration

You add the producer controller to the I/O Configuration folder of the remote (consumer) controller. Within the I/O Configuration folder, you organize the controllers and communication modules into a hierarchy (tree/branch, parent/child).

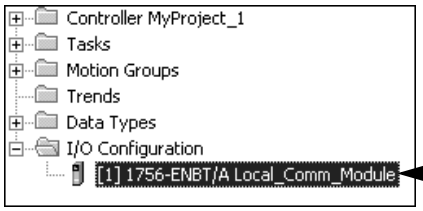
For a typical producer/consumer structure…

…you build the I/O configuration in this order:

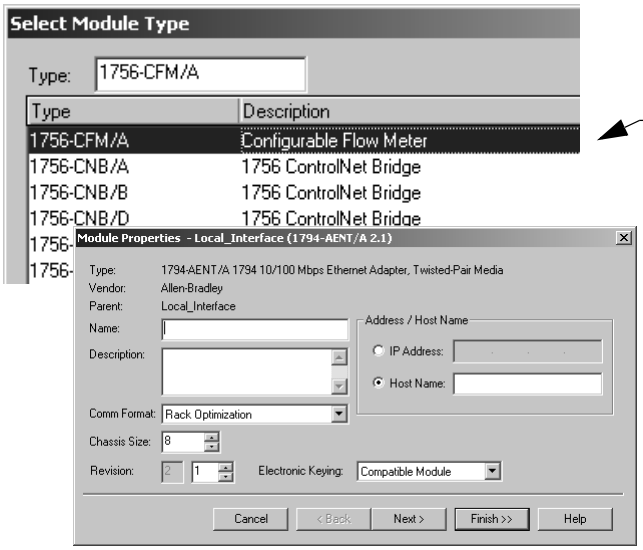
1. Add the local communication module for the consumer controller.
2. Add the remote communication module for the producer controller.
3. Add the producer controller.



To add a module to the I/O Configuration folder:



1. Right-click the level (branch) to which you want to add the module and choose New Module.



2. Choose the module.

3. Configure the module.

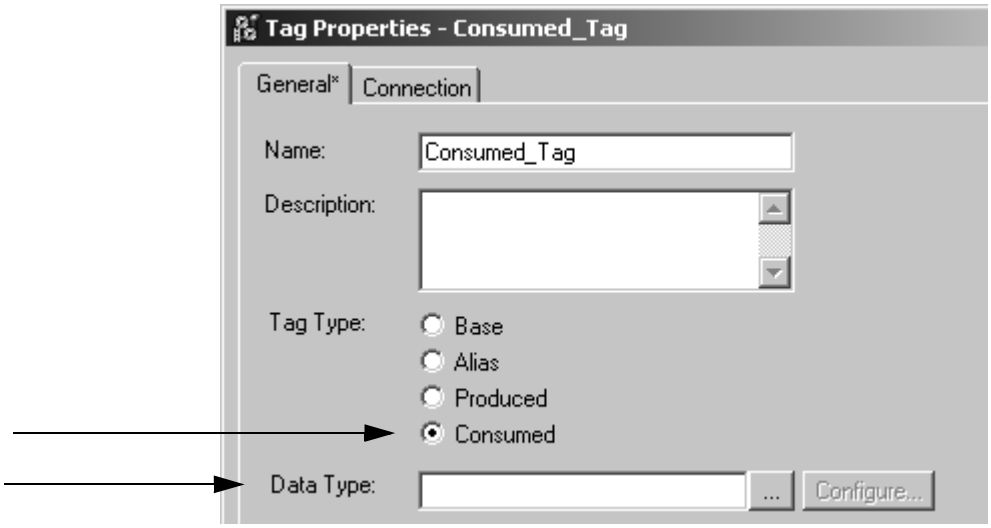
To	Do This
Use the default configuration	Specify the general information about the module (name, comm format) and click Finish.
Customize the configuration	Specify the general information about the module (name, comm format). Then use the Next buttons to step through subsequent screens.

Create the Consumed Tag

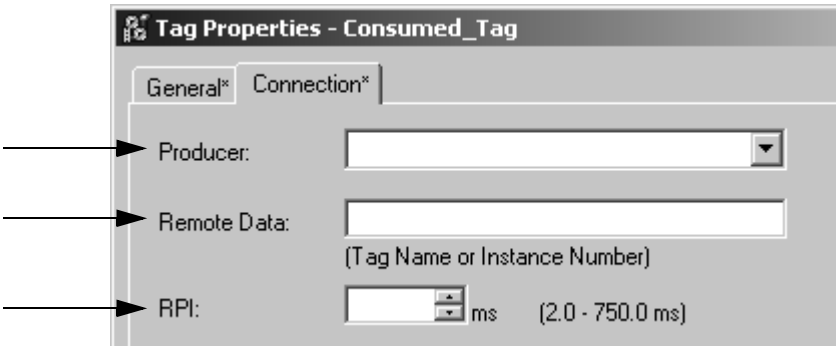
1. In the consumer controller's project, right-click the Controller Tags folder and choose Edit Tags. (Only controller-scoped tags can consume data.)

2. In the Controller Tags window, right-click the tag that will consume the data and choose Edit Tag Properties.

3. Select the Consumed option button and make sure the data type is the same as the produced tag.



4. Click the Connection tab and specify the producer controller.



   a. Select the controller that produces the data.
   b. Type the name of the produced data.
   c. Type or select the requested packet interval (RPI) for the connection.

   Set the RPI only as fast as needed by the application. The RPI also determines the number of packets per second that the module will produce on a connection. Each module has a limit of how many packets it can produce per second. If you exceed this limit, the module cannot open any more connections.

   For information on RPI and how it affects the actual packet interval (API), see the EtherNet/IP Performance Application Solution, ENET-AP001.

5. Click OK.

Guidelines for MSG Instructions

Follow these guidelines.

Guideline	Details
1. For each MSG instruction, create a control tag.	Each MSG instruction requires its own control tag. • Data type = MESSAGE • Scope = controller • The tag cannot be part of an array or a user-defined data type.
2. Keep the source and/or destination data at the controller scope.	A MSG instruction can access only tags that are in the Controller Tags folder (controller scope).
3. If your MSG is to a module that uses 16-bit integers, use a buffer of INTs in the MSG and DINTs throughout the project.	If your message is to a module that uses 16-bit integers, such as a PLC-5 or SLC 500 controller, and it transfers integers (not REALs), use a buffer of INTs in the message and DINTs throughout the project. This increases the efficiency of your project because Logix5000 controllers execute more efficiently and use less memory when working with 32-bit integers (DINTs).
4. Cache the connected MSGs that execute most frequently.	Cache the connection for those MSG instructions that execute most frequently, up to the maximum number permissible for your controller revision. This optimizes execution time because the controller does not have to open a connection each time the message executes.
5. If you want to enable more than 16 MSGs at one time, use some type of management strategy.	If you enable more than 16 MSGs at one time, some MSG instructions may experience delays in entering the queue. To guarantee the execution of each message, use one of these options: • Enable each message in sequence • Enable the messages in groups • Program a message to communicate with multiple modules • Program logic to coordinate the execution of messages
6. Keep the number of unconnected and uncached MSGs less than the number of unconnected buffers.	The controller can have 10 - 40 unconnected buffers. The default number is 10. • If all the unconnected buffers are in use when an instruction leaves the message queue, the instruction errors and does not transfer the data. • You can increase the number of unconnected buffers (40 max.).

For more information on programming MSG instructions, see the Logix5000 Controller General Instructions Reference Manual, publication 1756-RM003.

The individual system user manuals for Logix5000 controllers also provide MSG examples unique to specific controller platforms.

Determine Connections for Messages

Messages transfer data to other modules, such as other controllers or operator interfaces. Each message uses one connection, regardless of how many modules are in the message path. To conserve connections, you can configure one message to read from or write to multiple modules.

These connected messages can leave the connection open (cache) or close the connection when the message is done transmitting. The following table shows which messages use a connection and whether or not you can cache the connection.

This Type of Message	Using This Communication Method	Uses a Connection
CIP data table read or write	CIP	yes
PLC2, PLC3, PLC5, or SLC (all types)	CIP	no
	CIP with Source ID	no
	DH+	yes
CIP generic	CIP	your choice(1)
block-transfer read or write	na	yes

(1) You can connect CIP generic messages, but for most applications we recommend you leave CIP generic messages unconnected.

Guidelines For Caching Message Connections

- If a message executes repeatedly, cache the connection. This keeps the connection open and optimizes execution time. Opening a connection each time the message executes increases execution time.
- If a message executes infrequently, do not cache the connection. This closes the connection upon completion of the message, which frees up that connection for other uses.

Enter Message Logic

To send or receive data from an EtherNet/IP module via a message, you must program a MSG instruction in the local controller's logic. If the target module is configured in the I/O Configuration folder of the controller, you can browse to select the module. Otherwise, you can manually enter the message path in the MSG instruction.

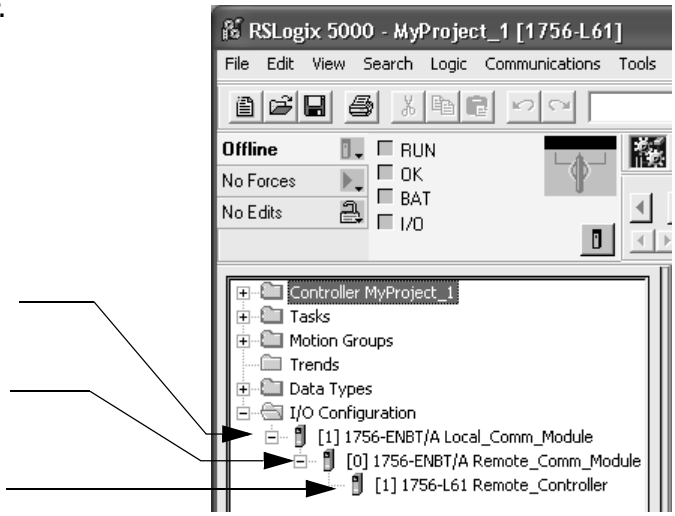
Add the EtherNet/IP Module To The Local Controller's I/O Configuration

To use the Browse button to select the target device of a MSG instruction, you add that remote device to the I/O Configuration folder of the local controller. Within the I/O Configuration folder, you organize the local and remote devices into a hierarchy (tree/branch, parent/child).

For a typical local/remote MSG structure…

…you build the I/O configuration in this order.

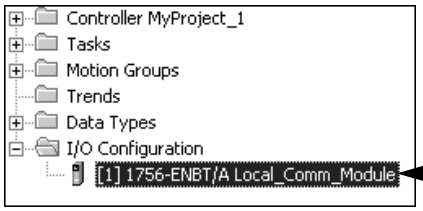
1. Add the local communication module for the local controller.
2. Add the remote communication module for the remote controller.
3. Add the remote controller.



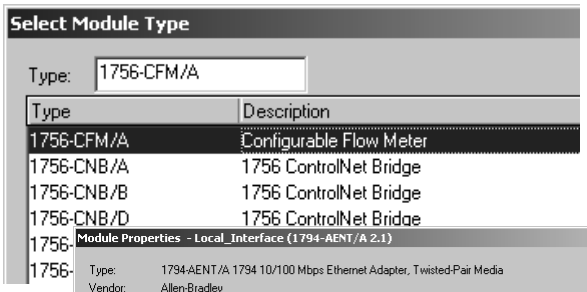
You also select a communication format for a communication module based on the modules in its remote chassis.

If	Select a Communication Format That Specifies
The remote chassis contains only analog modules, diagnostic digital modules, fused output modules, or communication modules	None
The remote chassis only contains standard, digital input and output modules (no diagnostic modules or fused output modules)	Rack Optimization
You want to receive I/O module and chassis slot information from a rack-optimized remote chassis owned by another controller	Listen-Only Rack Optimization

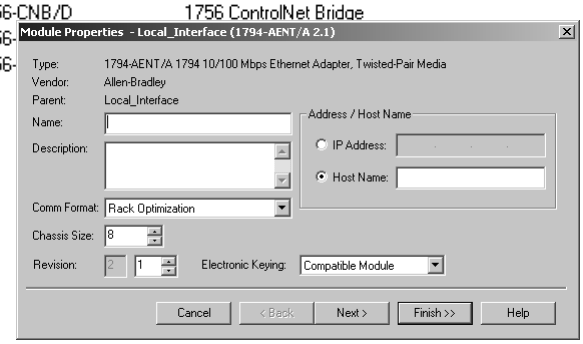
To add a module to the I/O Configuration folder:



1. Right-click the level (branch) to which you want to add the module and choose New Module.



2. Choose the module.



3. Configure the module.

To	Do This
Use the default configuration	Specify the general information about the module (name, comm format) and click Finish.
Customize the configuration	Specify the general information about the module (name, comm format). Then use the Next buttons to step through subsequent screens.

Enter a Message

Use relay ladder logic to enter a MSG instruction. Click the button ... to configure the MSG instruction.

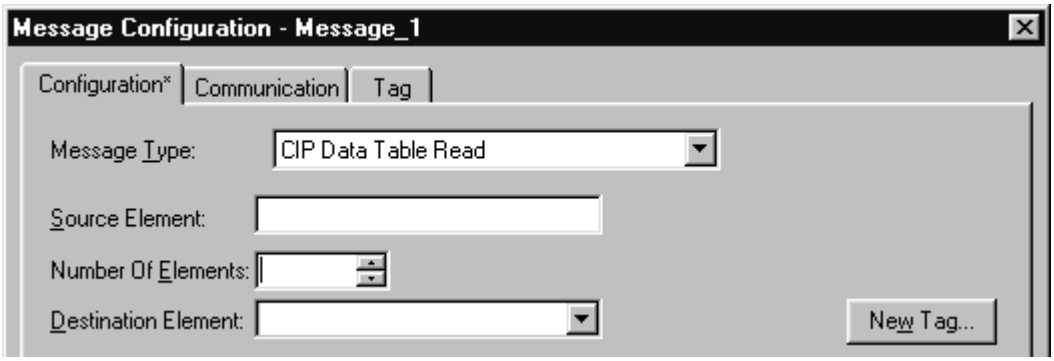
EXAMPLE Enter a MSG instruction

If *count_send* = 1 and *count_msg.EN* = 0 (MSG instruction is not already enabled), then execute a MSG instruction that sends data to another controller.

Configure a MSG Instruction

To configure a MSG instruction, click ... in the MSG box.

On the Configuration tab, specify the type of MSG instruction:



Configure a MSG to Logix5000 Controller

If You Want To	For This Item	Type Or Select
Read (receive) the data	Message Type	CIP Data Table Read
	Source Element	First element of the tag that contains data in the other controller
	Number of Elements	Number of elements to transfer
	Destination Tag	First element of the tag (controller-scoped) in this controller for the data
Write (send) the data	Message Type	CIP Data Table Write
	Source Tag	First element of the tag (controller-scoped) in this controller that contains the data
	Number of Elements	Number of elements to transfer
	Destination Element	First element of the tag for the data in the other controller

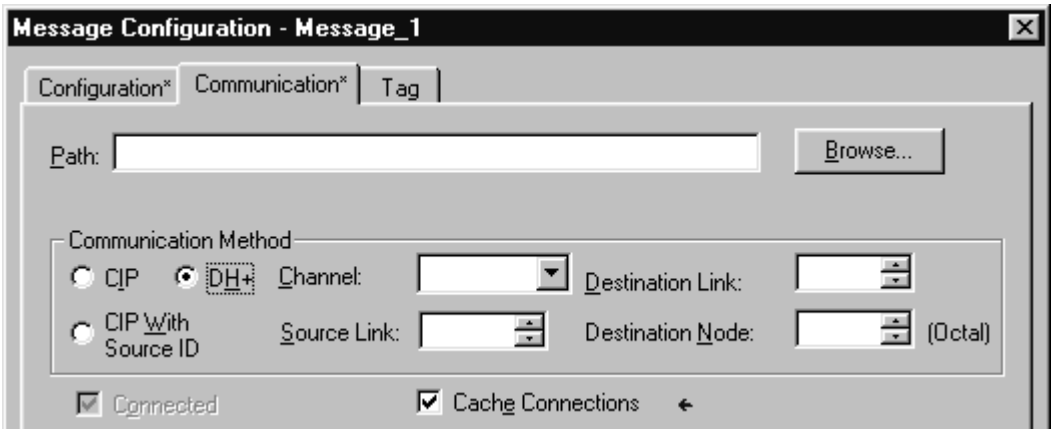
Configure a MSG To an SLC 500 Processor

If the data is	And you want to	For this item	Type or select
Integer	Read (receive) data	Message Type	SLC Typed Read
		Source Element	Data table address in the SLC 500 controller (e.g., N7:10)
		Number of Elements	Number of integers to transfer
		Destination Tag	First element of int_buffer
	Write (send) data	Message Type	SLC Typed Write
		Source Tag	First Element of int_buffer
		Number of Elements	Number of integers to transfer
		Destination Element	Data table address in the SLC 500 controller (for example, N7:10)
Floating-point (REAL)	Read (receive) data	Message Type	SLC Typed Read
		Source Element	Data table address in the SLC 500 controller (for example, F8:0)
		Number of Elements	Number of values to transfer
		Destination Tag	First element of the tag (controller-scoped) in this controller for the data
	Write (send) data	Message Type	SLC Typed Write
		Source Tag	First element of the tag (controller-scoped) in this controller that contains the data
		Number of Elements	Number of values to transfer
		Destination Element	Data table address in the SLC 500 controller (for example, F8:0)

Configure a MSG To a PLC-5 Processor

If the Data Is	And You Want To	For This Item	Type or Select
Integer	Read (receive) data	Message Type	PLC5 Typed Read
		Source Element	Data table address in the PLC-5 controller (e.g., N7:10)
		Number of Elements	Number of integers to transfer
		Destination Tag	First element of int_buffer
	Write (send) data	Message Type	PLC5 Typed Write
		Source Tag	First element of int_buffer
		Number of Elements	Number of integers to transfer
		Destination Element	Data Table address in the PLC-5 controller (e.g., N7:10)
Floating-point (REAL)	Read (receive) data	Message Type	PLC5 Typed Read
		Source Element	Data table address in the PLC-5 controller (e.g., F8:0)
		Number of Elements	Number of values to transfer
		Destination Tag	First element of the tag (controller-scoped) in this controller for the data
	Write (send) data	Message Type	PLC5 Typed Write
		Source Tag	First element of the tag (controller-scoped) in this controller that contains the data
		Number of Elements	Number of values to transfer
		Destination Element	Data table address in the PLC-5 controller (e.g., F8:0)

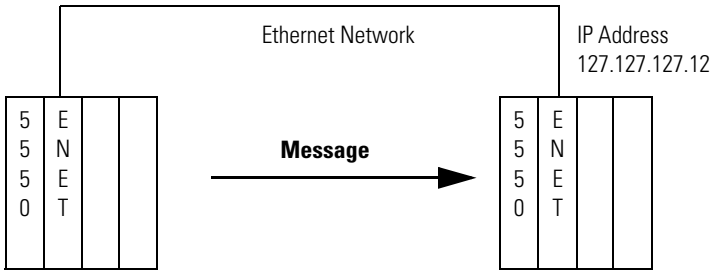
On the Communication tab, specify the communications details.



If the target module is configured in the I/O Configuration folder of the originating controller, use the Browse button to select the module. Otherwise, manually enter the path to the target module.

A manually entered path starts with the name of the local EtherNet/IP module, the port the message exits (2 for EtherNet/IP), and the IP address of the next module in the path (which could be the target module). For example:

EXAMPLE Communication path from a Logix5000 controller to a Logix5000 controller over an EtherNet/IP network



washer, 2, 127.127.127.12, 1, 0

Where	Indicates
washer	Name of the ENB or ENET module
2	Ethernet port of the ENB or ENET module
127.127.127.12	IP address of the ENB or ENET module in the destination chassis
1	Backplane port of the ENB or ENET module in the destination chassis
0	Slot number of the destination controller

Communicate with PLC-5 or SLC Processors

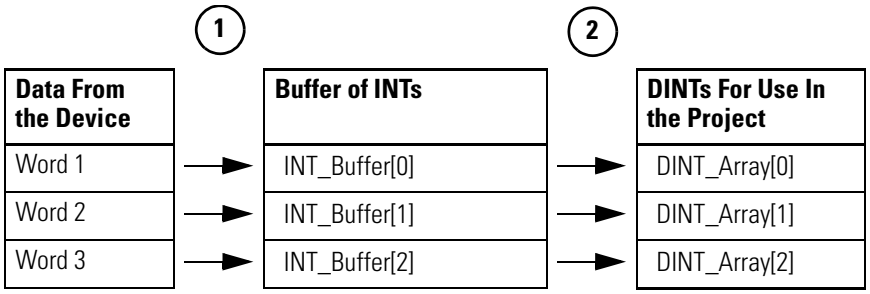
If the message is to a PLC-5 or SLC 500 processor and it reads or writes integers (not REALs), use a buffer of INTs in the message.

- Logix5000 controllers execute more efficiently and use less memory when working with 32-bit integers (DINTs).
- PLC-5 and SLC 500 processors require 16-bit integers.
- Use an INT buffer in the message and move the data into or out of the buffer as needed.

Convert between INTs and DINTs

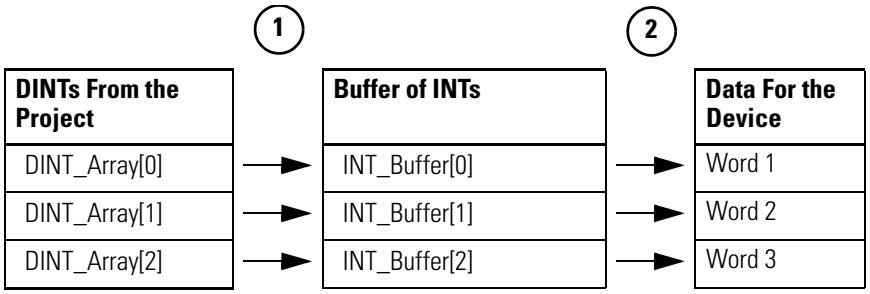
If your message is to a device that uses 16-bit integers, such as a PLC-5 or SLC 500 controller, and it transfers integers (not REALs), use a buffer of INTs in the message and DINTs throughout the project. This increases the efficiency of your project.

Read 16-Bit Integers



1. The Message (MSG) instruction reads 16-bit integers (INTs) from the device and stores them in a temporary array of INTs.
2. An File Arith/Logical (FAL) instruction converts the INTs to DINTs for use by other instructions in your project.

Write 16-Bit Integers

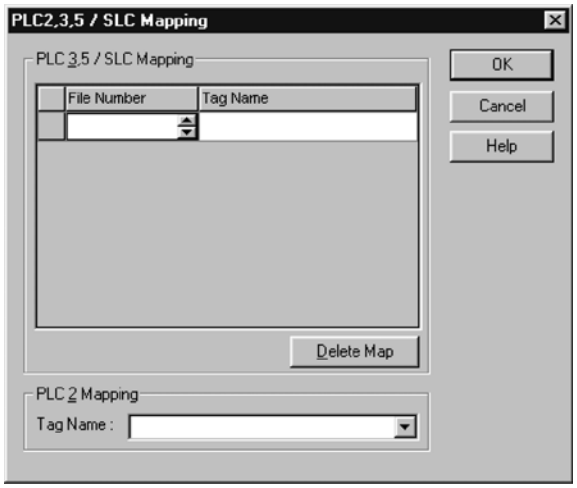


1. An FAL instruction converts the DINTs from the Logix5000 controller to INTs.
2. The MSG instruction writes the INTs from the temporary array to the device.

Map Tags

A Logix5000 controller stores tag names on the controller so that other devices can read or write data without having to know physical memory locations. Many products only understand PLC/SLC data tables, so the Logix5000 controller offers a PLC/SLC mapping function that lets you map Logix tag names to memory locations.

- You only have to map the file numbers that are used in messages; the other file numbers do not need to be mapped.
- The mapping table is loaded into the controller and is used whenever a logical address accesses data.
- You can only access controller-scoped tags (global data).



- For each file that is referenced in a PLC-5 or SLC command, make a map entry:
  - Type the PLC/SLC file number of the logical address.
  - Type or select the Logix5000 controller-scoped (global) tag that supplies or receives data for the file number. (You can map multiple files to the same tag.)
- For PLC-2 commands, specify the tag that supplies or receives the data.

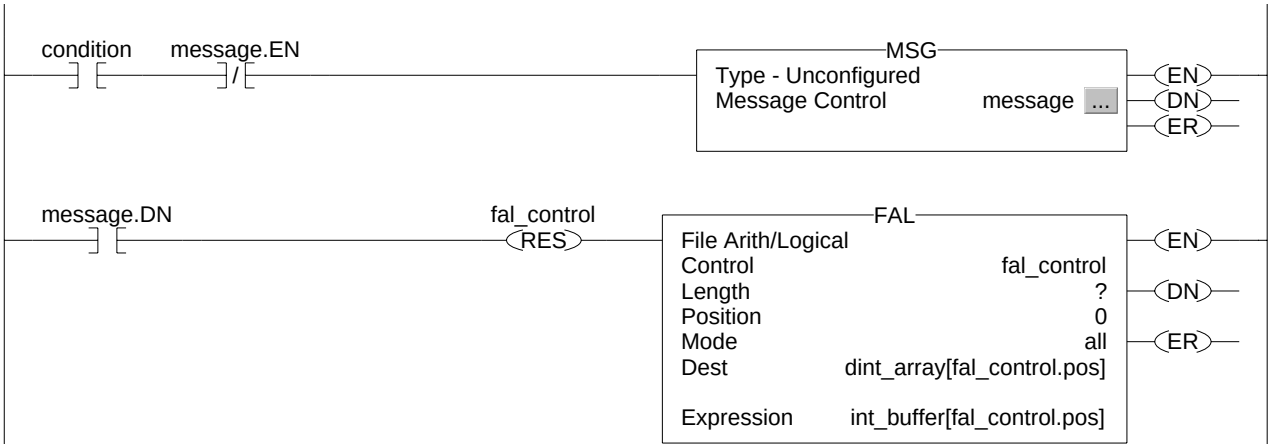
When mapping tags:

- Do not use file numbers 0, 1, and 2. These files are reserved for Output, Input, and Status files in a PLC-5 processor.
- Use PLC-5 mapping only for tag arrays of data type INT, DINT, or REAL. Attempting to map elements of system structures may produce undesirable effects.
- Use the PLC file identifier of N or B when accessing elements in an INT tag array.

This example shows how to use a buffer of INTs.

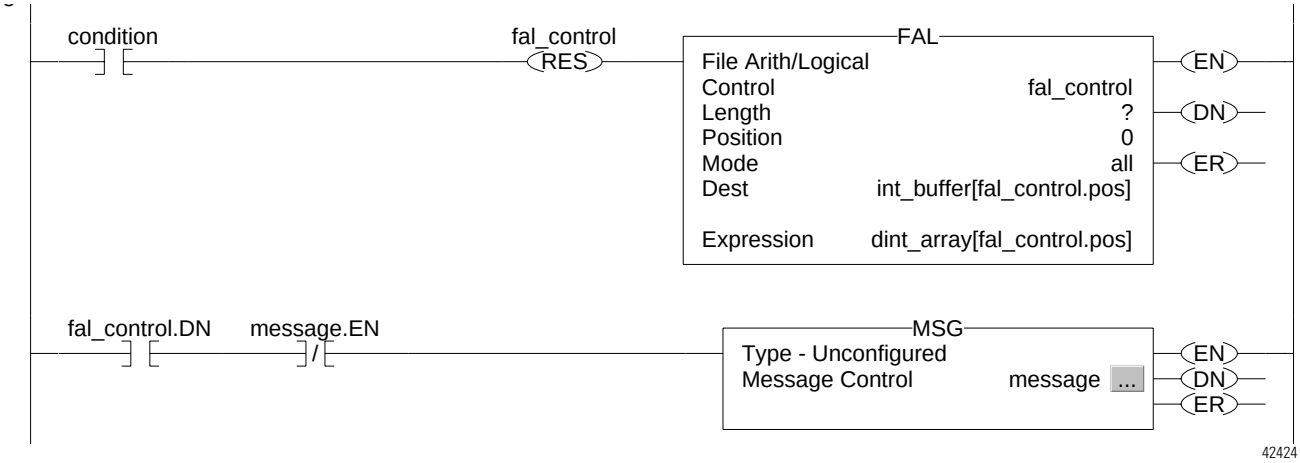
EXAMPLE Read integers from a PLC-5 controller

When condition turns on, reads 16-bit integer values (INTs) and stores them in int_buffer. Then the FAL instruction moves the values to dint_array. This converts the values to 32-bit integers (DINTs), for use by other instructions in the ControlLogix controller.



EXAMPLE Write integers to a PLC-5 controller

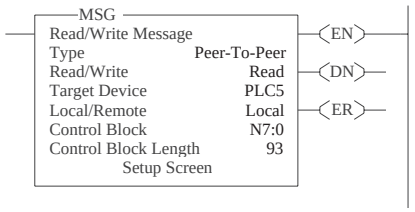
When condition turns on, moves the values in dint_array to int_buffer. This converts the values to 16-bit integers (INTs). Then the message instruction sends int_buffer to the other controller.



Where	Is an
dint_array	Array of DINTs that are used in the ControlLogix controller
int_buffer	Array of INTs with the same number of elements as dint_array

Receive MSGs from PLC-5 or SLC 500 processors

If the originating controller is a PLC-5 or SLC 500 processor, in the MSG instruction, select PLC5.



If the Controller Is a	For This Section	And This Item	Specify
PLC-5	This PLC-5	Communication Command	PLC-5 Typed Read or PLC-5 Typed Write
		Data Table Address	Starting address of the data in the PLC-5 controller
		Size in Elements	Number of elements to read or write
		Port Number	2
	Target Device	Data Table Address	Type, in quotation marks [" "], the name of the tag in the ControlLogix controller (for example, "count").
		MultiHop	Select Yes.
SLC 500	This Controller	Communication Command	PLC5 Read or PLC5 Write
		Data Table Address	Starting address of the data in the SLC 500 controller
		Size in Elements	Number of elements to read or write
		Channel	1
	Target Device	Data Table Address	Type, in quotation marks [" "], the name of the tag in the ControlLogix controller (for example, "count").
		MultiHop	Select Yes

On the MultiHop tab, specify:

- The IP address of the EtherNet/IP communication module that is local to the Logix5000 controller.
- The slot number of the Logix5000 controller.

Send Email

Use This Chapter

Read this chapter for:

- *1756-ENBT module*
- *1769-L32E, -L35E controller*
- *1768-ENBT module*
- *1788-ENBT card*

This chapter describes how to send an email through an EtherNet/IP module.

For email, the EtherNet/IP module can be remote or local to the controller.

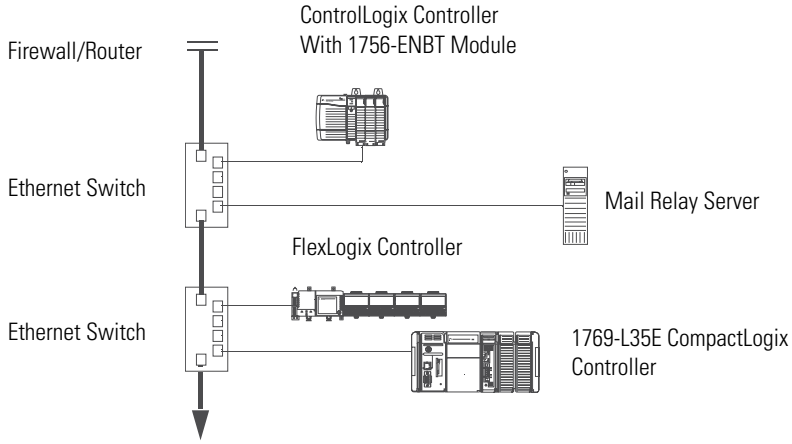
Overview

The EtherNet/IP module is an email client that uses a mail relay server to send email.

If You Want To	Then
Send an email to specific personnel when a controller application generates an alarm or reaches a certain condition	Program the controller to send a MSG instruction to the EtherNet/IP module The MSG instruction then instructs the EtherNet/IP module to send the email text (contained within the MSG instruction) to the mail relay server. Multiple controllers can use the same EtherNet/IP module to initiate email.
Send controller or application status information on a regular basis to a project manager	

The EtherNet/IP module sends only the content of a MSG instruction as an email to a mail relay server. Delivery of the email depends on the mail relay server. The EtherNet/IP module does not receive email.

For example, in this sample system:



This Device	Can
ControlLogix controller	Send a MSG instruction to the 1756-ENBT module to initiate sending an email to the mail relay server Use the path of the MSG instruction to identify the 1756-ENBT module as the target of the MSG instruction.
FlexLogix controller	
CompactLogix controller	
1756-ENBT module	Send an email to the mail relay server from the email interface on the Send an Email link. Each time you use this interface, you must enter all the email information.
mail relay server	Send email to specified recipients The mail relay server determines the delivery of any email send through an EtherNet/IP module, whether via a MSG instruction or from its built-in interface.

Send an Email Via a Controller-initiated Message Instruction

A Logix controller can send a generic CIP message instruction to the EtherNet/IP module that instructs the module to send an email message to a SMTP mail relay server using the standard SMTP protocol. This is useful to automatically communicate controller data and/or application conditions to appropriate personnel.

IMPORTANT Be careful to write the ladder logic to ensure the MSG instructions are not continuously triggered to send email messages.

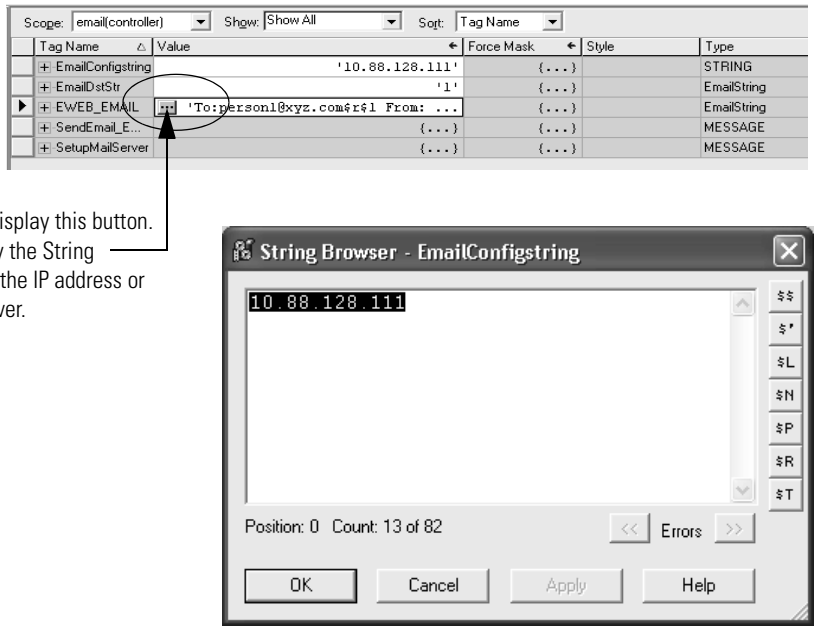
Some mail relay servers require a domain name be provided during the initial handshake of the SMTP session. For these mail relay servers, make sure you specify a domain name when you configure the network settings for the EtherNet/IP module.

Step 1: Create String Tags

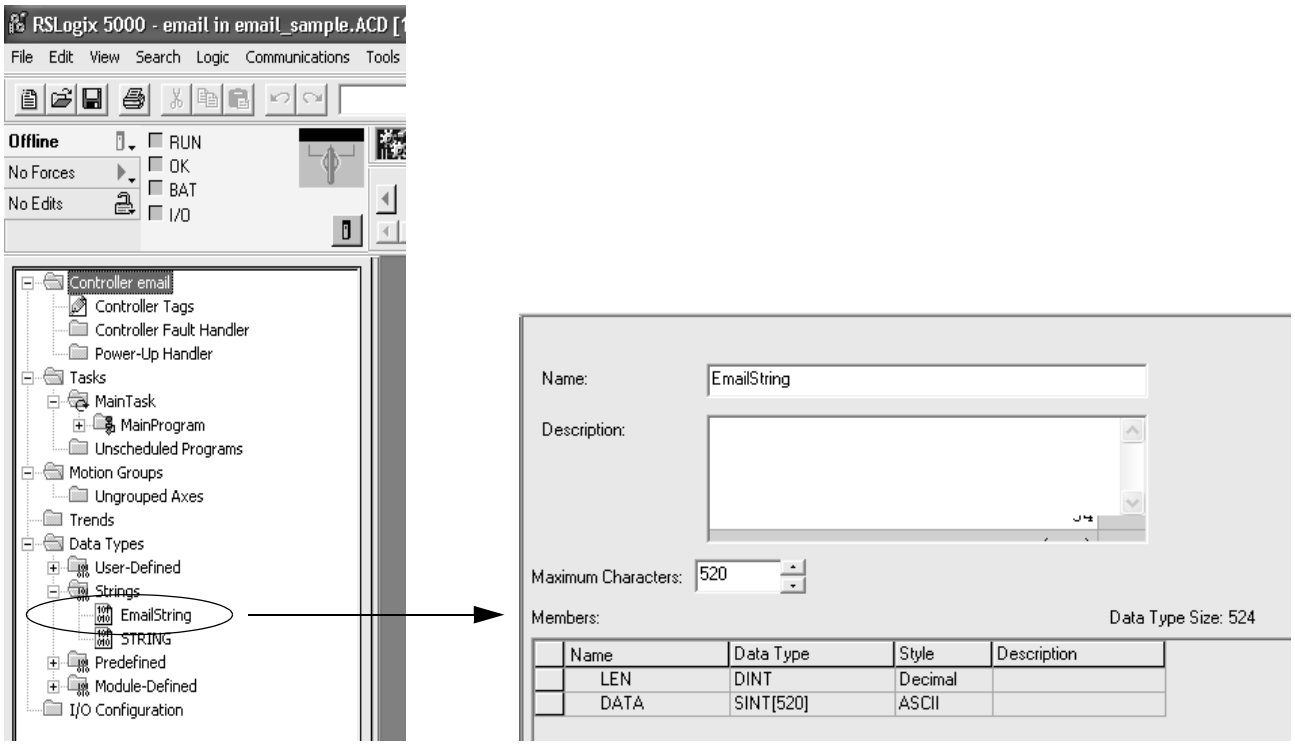
You need three controller-scoped, string tags. Each tag performs one of these functions.

- Identify the mail server
- Contain the email text
- Contain the status of the email transmission

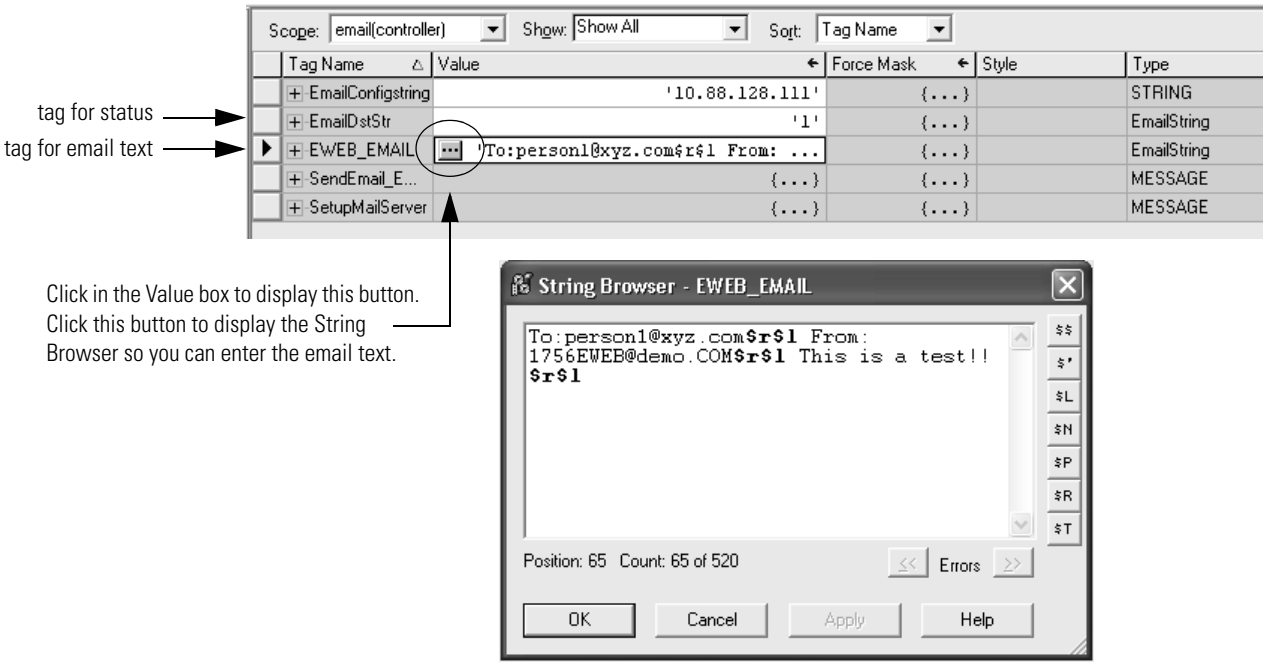
The default STRING data type supports as many as 82 characters. In most cases, this is sufficient to contain the address of the mail server. For example, create tag EmailConfigstring of type STRING:



The tags for the email text and transmission status can contain as many as 474 characters. For these tags, you must create a user-defined STRING data type (the default STRING data type in RSLogix 5000 software is not large enough for most email text). For example, create a STRING data type named EmailString.



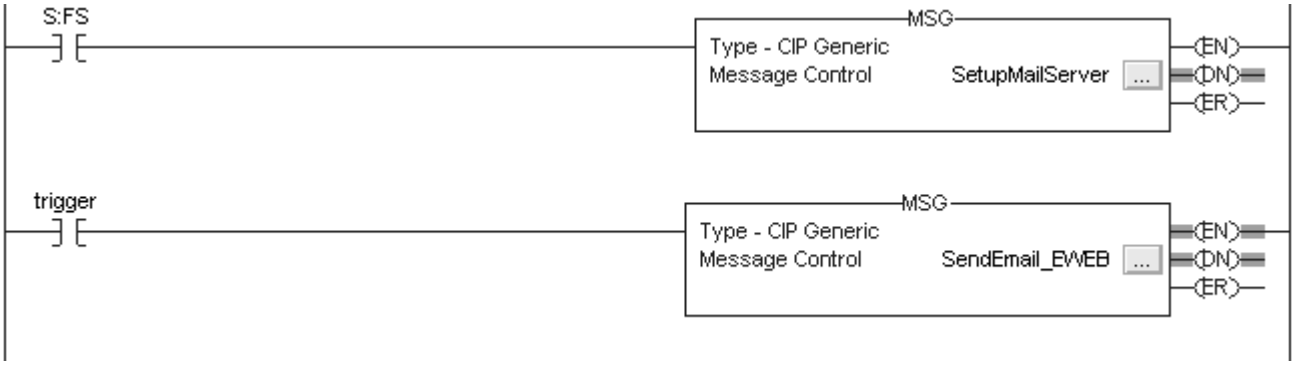
Create one controller-scoped tag of this new data type to contain the email text. Create a second controller-scoped tag of this new data type to contain the transmission status. For example, create tag EWEB_EMAIL (to contain the email text) and EmailDstStr (to contain the transmission status). Both of these tags are of type EmailString.



The text of the email does not have to be static. You can program a controller project to collect specific data to be sent in an email. For more information on using ladder logic to manipulate string data, see the Logix5000 Controllers Common Procedures Programming Manual, publication 1756-PM001.

Step 2: Enter the Ladder Logic

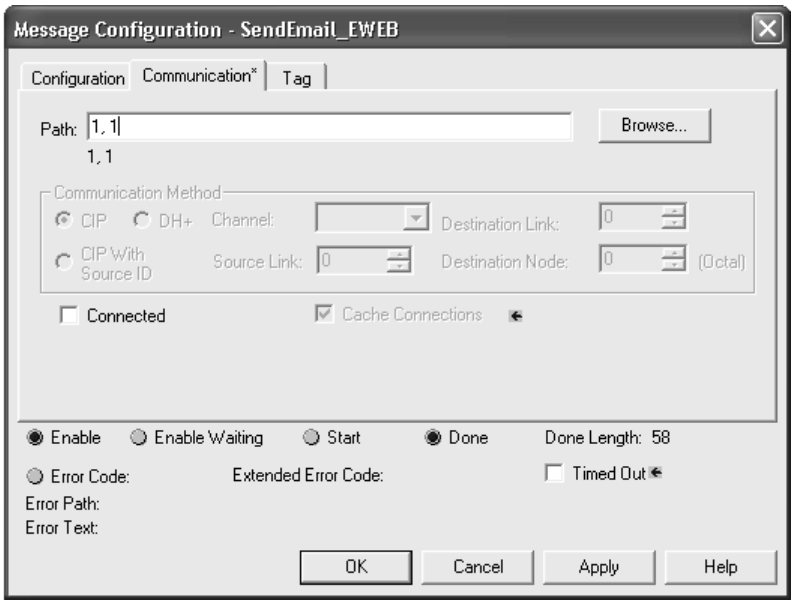
You need two MSG instructions. One MSG instruction configures the mail server. This only needs to be executed once. The next MSG instruction triggers the email. Execute this email MSG instruction as often as needed.



The first rung configures the mail server. The second rung sends the email text.

Step 3: Configure the MSG Instruction That Identifies the Mail Relay Server

On the Communication tab of the MSG instruction, configure the path for the MSG instruction.



The path starts with the controller initiating the MSG instruction. Then enter the port the message exits and the address of the next module in the path. For example, if the EtherNet/IP module is in the same chassis as the controller and is in slot 2, the path is: 1, 2.

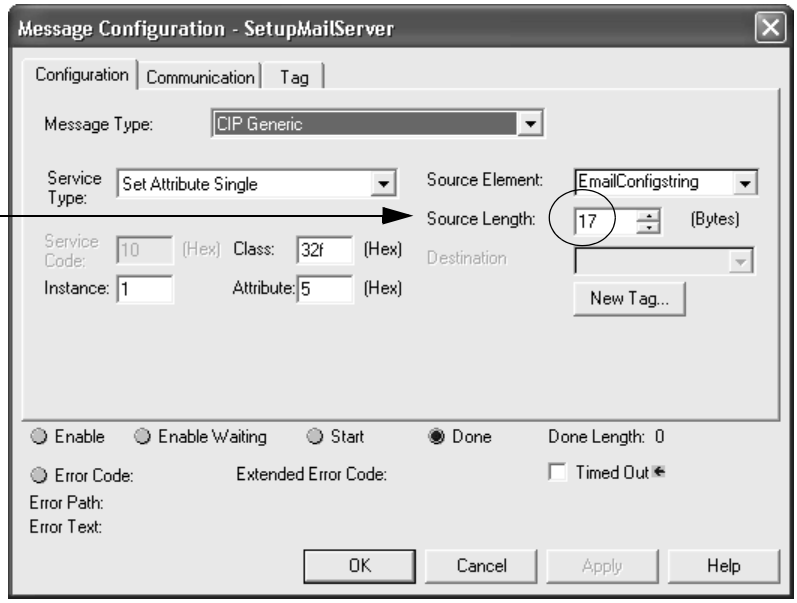
For more information on configuring the path of a MSG instruction, see the Logix5000 Controllers General Instructions Reference Manual, publication 1756-RM003.

On the Communication tab of the MSG instruction, configure the MSG parameters for identifying the mail relay server.

Some mail relay servers require a domain name be provided during the initial handshake of the SMTP session. For these mail relay servers, make sure you specify a domain name when you configure the network settings for the EtherNet/IP module.

The Source Length is the number of characters in the STRING tag that identifies the mail relay server plus 4 characters.

In this example, the tag contains 13 characters.



where:

In This Field	Enter
Service Type	Set Attribute Single
Instance	1
Class	32f
Attribute	5
Source Element	The STRING tag that contains the IP address or host name of the mail relay server In this example, enter EmailConfigstring
Source Length	The number of characters in the IP address or host name of the mail server plus 4 In this example, enter 17 (13 characters in the IP address 10.88.128.111 + 4)

After the MSG instruction that configures the mail relay server executes successfully, the controller stores the mail relay server information in non-volatile memory. The controller retains this information, even through power cycles, until another MSG instruction changes the information.

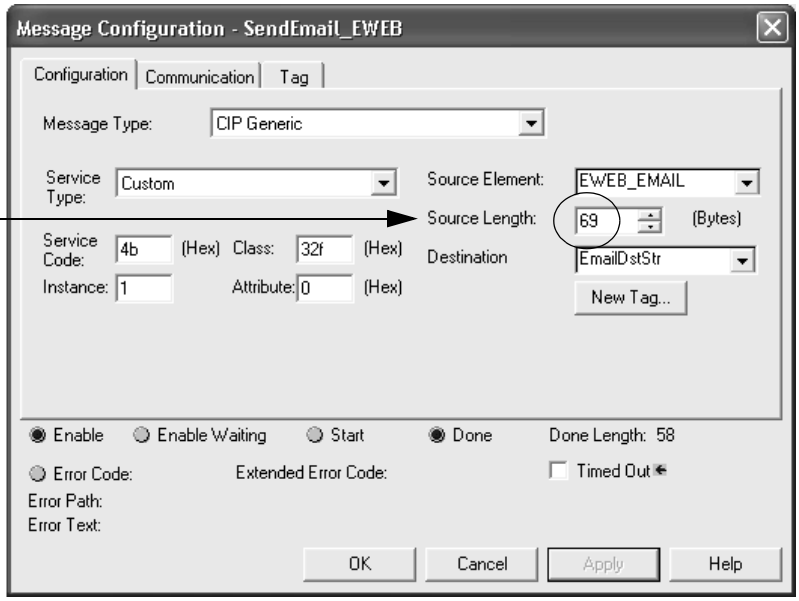
Step 4: Configure the MSG Instruction That Contains the Email Text

On the Communication tab of the MSG instruction, configure the path for the MSG instruction. This is the same as for the MSG instruction that identifies the mail relay server (see page 6-6).

On the Configuration tab of the MSG instruction, configure the MSG parameters for sending an email.

The Source Length is the number of characters in the email tag plus 4 characters.

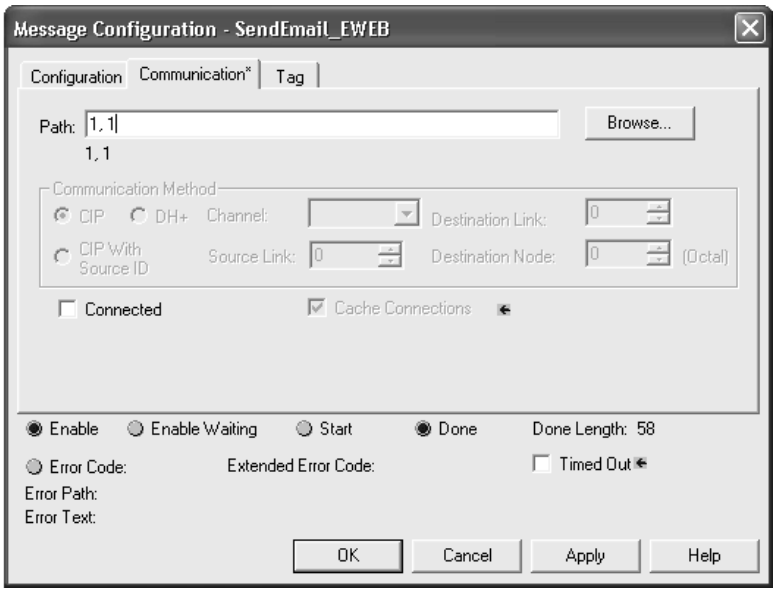
In this example, the email text contains 65 characters.



where:

In This Field	Enter
Service Type	Custom
Service Code	4b
Instance	1
Class	32f
Attribute	0
Source Element	The tag that contains the email text This tag is of the STRING data type you created to contain the email text. In this example, enter EWEB_EMAIL which is of type EmailString
Source Length	The number of characters in the email text plus 4 In this example, enter 69 (65 characters in the email + 4)
Destination	A tag to contain the status of the email transmission This tag is also of the STRING data type you created to contain the email text. In this example, enter EmailDstStr which is of type EmailString

On the Communication tab of the MSG instruction, configure the path from the controller to the EtherNet/IP module.



The path starts with the controller initiating the MSG instruction. Then enter the port the message exits and the address of the next module in the path. For example, if the EtherNet/IP module is in the same chassis as the controller and is in slot 2, the path is: 1, 2.

If all the devices in the path are configured in the initiating controller's I/O Configuration tree, you can use the Browse button to select the target module and the software automatically fills in the path.

For more information on configuring the path of a MSG instruction, see the Logix5000 Controllers General Instructions Reference Manual, publication 1756-RM003.

Enter the Text of the Email

Use the string browser to enter the text of the email. In the example above, you enter the email text into the EWEB_EMAIL tag. To include "To:", "From:", and "Subject:" fields in the email, use <CR><LF> symbols to separate each of these fields. The "To:" and "From"" fields are required; the "Subject:" field is optional. Use a second set of <CR><LF> symbols after the last one of these fields you enter. For example:

To: email address of recipient rl
From: email address of sender$r$l
Subject: subject of message rlrl
body of email message

The maximum length of an email message is 474 characters. An additional 4-byte string-length value is added to the tag. As a result, the maximum source length is 478 characters.

Possible Email Status Codes

Examine the destination element of the email MSG to see whether the email was successfully delivered to the mail relay server. This indicates that the mail relay server placed the email message in a queue for delivery. It does not mean the intended recipient successfully received the email message. Possible codes that could be in this destination element are:

Error Code (Hex)	Extended-Error Code (Hex)	Description
0x00	None	Delivery successful to the mail relay server
0x02	None	Resource unavailable. The email object was unable to obtain memory resources to initiate the SMTP session
0x08	None	Unsupported Service Request. Make sure the service code is 0x4B and the Class is 0x32F.
0x11	None	Reply data too large. The Destination string must reserve space for the SMTP server reply message. The maximum reply can be 470 bytes.
0x13	None	Configuration data size too short. The Source Length is less than the Source Element string size plus the 4-byte length. The Source Length must equal the Source Element string size + 4.
0x15	None	Configuration data size too large. The Source Length is greater than the Source Element string size plus the 4-byte length. The Source Length must equal the Source Element string size + 4.
0x19	None	Data write failure. An error occurred when attempting to write the SMTP server address (attribute 4) to non-volatile memory.

Error Code (Hex)	Extended-Error Code (Hex)	Description
0xFF	0x0100	Error returned by email server; check the Destination string for reason. The email message was not queued for delivery.
	0x0101	SMTP mail server not configured. Attribute 5 was not set with a SMTP server address.
	0x0102	"To:" address not specified. Attribute 1 was not set with a "To:" address AND there is not a "To:" field header in the email body.
	0x0103	"From:" address not specified. Attribute 2 was not set with a "From:" address AND there is not a "From:" field header in the email body.
	0x0104	Unable to connect to SMTP mail server set in Attribute 5. If the mail server address is a hostname, make sure that the device supports DNS, and that a Name Server is configured. If the hostname is not fully qualified, i.e., "mailhost" and not "mailhost.xx.yy.com" then the domain must be configured as "xx.yy.com". Try "ping <mail server address>" to insure the mail server is reachable from your network. Also try "telnet <mail server address> 25" which attempts to initiate a SMTP session with the mail server via telnet over port 25. (If you connect then enter "QUIT").
	0x0105	Communication error with SMTP mail server. An error occurred after the initial connection with the SMTP mail server. See the ASCII text following the error code for more details as to the type of error.
	0x0106	SMTP mail server host name DNS query did not complete. A previous send service request with a host name as the SMTP mail server address did not yet complete. Note that a timeout for a DNS lookup with an invalid host name can take up to 3 minutes. Long timeouts can also occur if a domain name or name server is not configured correctly.

Notes:

Communicate With PanelView Terminals

Use This Chapter

Read this chapter for:

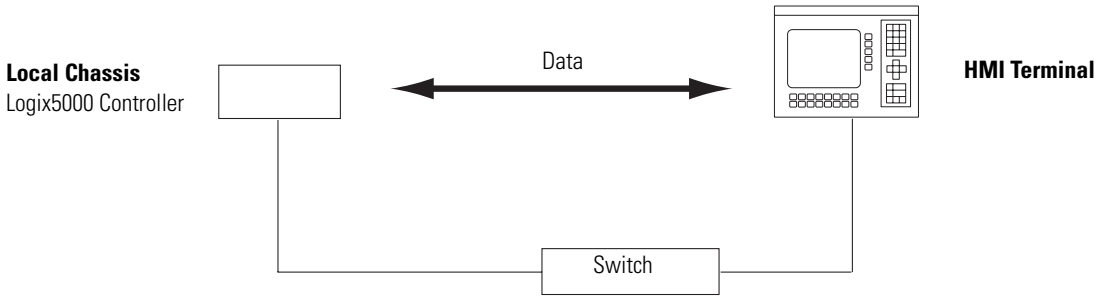
- *1756-ENBT module*
- *1769-L32E, -L35E controller*
- *1768-ENBT module*
- *1788-ENBT card*

This chapter describes how a controller uses an EtherNet/IP communication module to communicate with PanelView and PanelView Plus terminals over an EtherNet/IP network.

Set Up the Hardware

In this example, the controller in the local chassis shares data with an HMI application on the EtherNet/IP network. This application could be running any of the following:

- PanelView terminal
- PanelView Plus terminal
- Workstation running an RSView 32 software
- Workstation running an RSView Enterprise application, such as RSView Machine Edition or RSView Supervisory Edition



The Logix5000 controller in the local chassis can be any of the following, with its EtherNet/IP communication module:

- 1756 ControlLogix controller with a 1756-ENBT communication module in the chassis.
- 1768 CompactLogix controller with a 1768-ENBT communication module in the chassis.
- 1769-L35E CompactLogix controller with built-in EtherNet/IP port.
- 1794 FlexLogix controller with a 1788-ENBT EtherNet/IP communication card.
- PowerFlex 700S with DriveLogix controller and a 1788-ENBT EtherNet/IP communication card.

Make sure that:

- The IP addresses are set for the controller's EtherNet/IP communication module and the HMI terminal.
- All wiring and cabling is properly connected.

Determine Connections to PanelView Terminals

How you establish communication between a PanelView or PanelView Plus terminal depends on how you want to use controller connections.

Type of Communication	Terminal Type	
	PanelView	PanelView Plus
Implicit (connected) • Logix controller communicates to the PanelView terminal like an I/O module • You must add the PanelView terminal to the I/O configuration tree for the controller project	Supported	Not supported
Explicit (unconnected) • Communications are set up in PanelBuilder or RSView ME Software • All communications are initiated by the PanelView or PanelView Plus terminal	Supported	Supported

When communicating implicitly (PanelView terminals only), the controller uses one connection for each terminal. Make sure to account for these connections when designing the system. The Logix5000 controllers:

- Firmware revisions 11 and earlier support as many as 16 bidirectional implicit buffers (connections)
- Firmware revisions 12 or greater support as many as 32 bidirectional implicit buffers (connections)

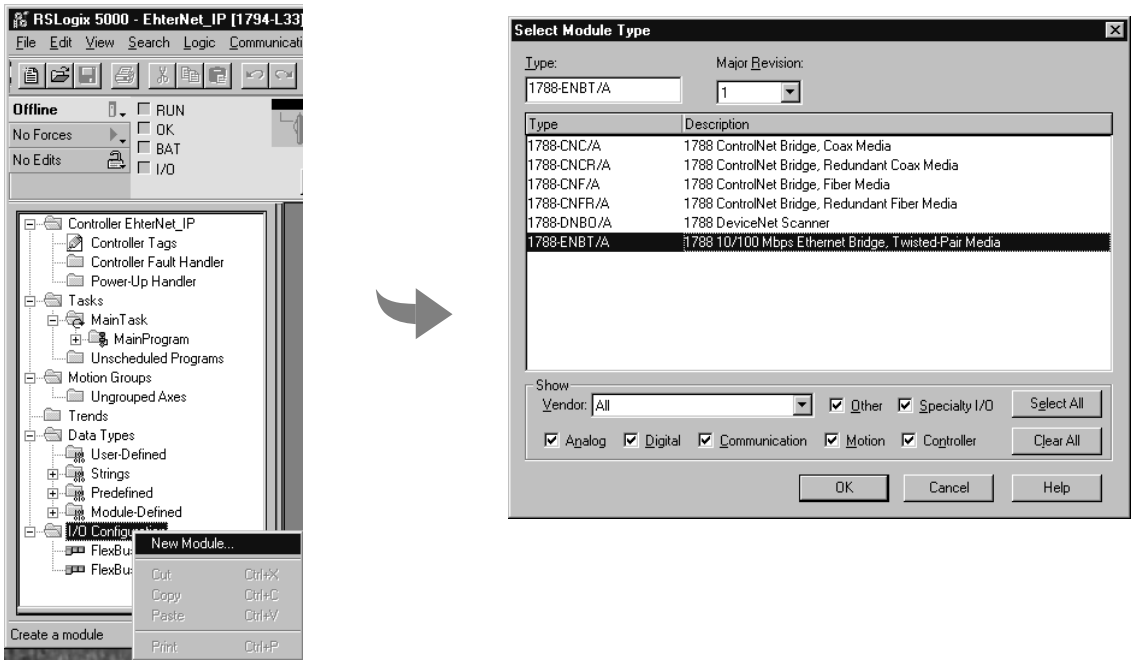
The larger number of implicit buffers allows significantly more PanelView terminals to simultaneously request data from the controller via implicit communications.

When communicating explicitly, the controller supports 40 outgoing and 3 incoming buffers. This number of incoming buffers limits how many terminals can simultaneously request data from a controller via explicit communications. In other words, while you can have multiple terminals in a system, only three terminals can explicitly request data from a Logix controller at the same time.

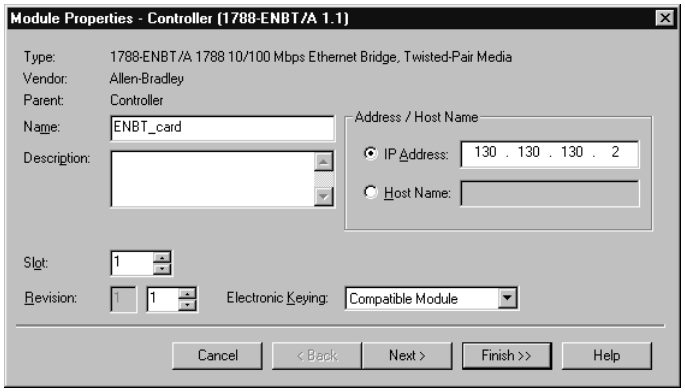
Add a PanelView Terminal

Adding a PanelView terminal is similar to adding distributed I/O. You add the local EtherNet/IP communication module and then you add the terminal to that module.

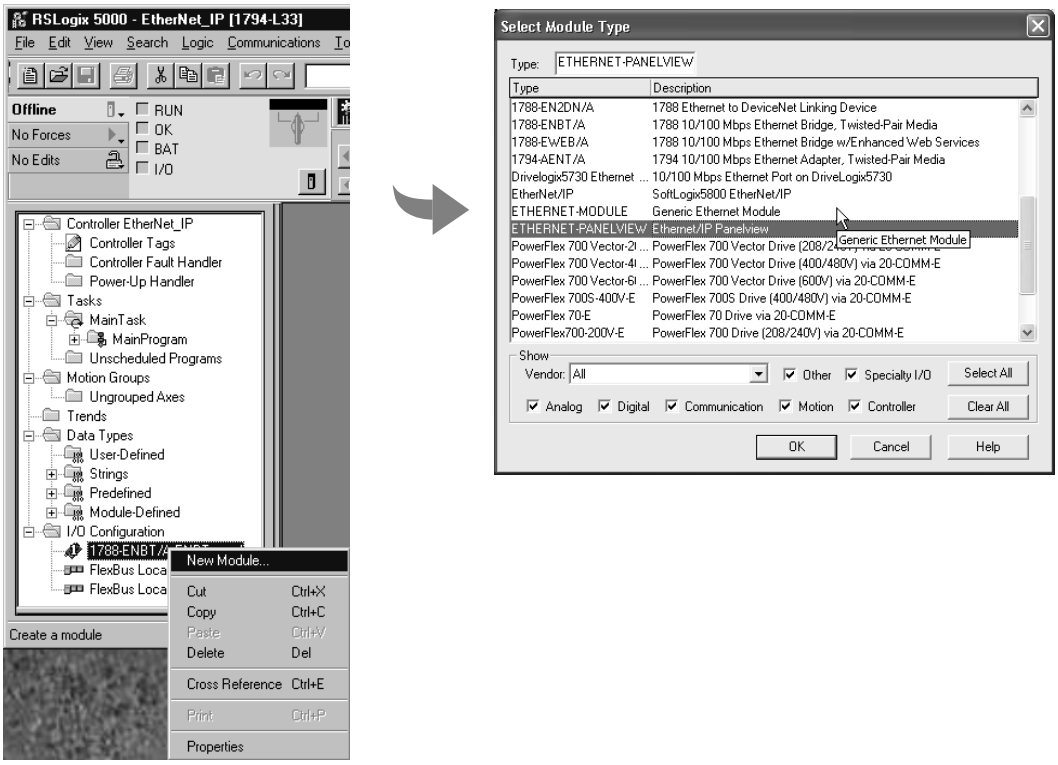
1. In RSLogix 5000 programming software, right-click to select New Module, and add the local EtherNet/IP communication module.



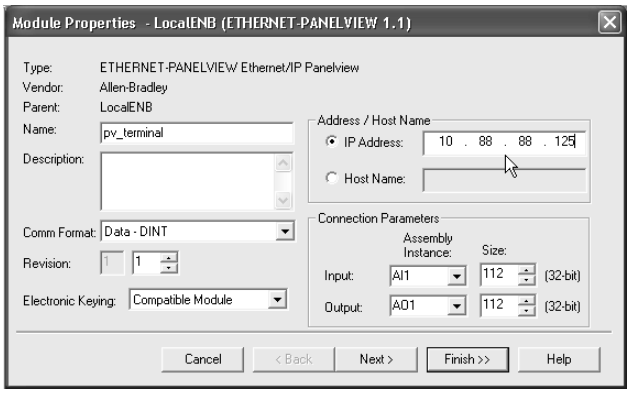
2. Configure the local EtherNet/IP communication module.



3. Select the local EtherNet/IP communication module, right-click to select New Module, and add an ETHERNET-PANELVIEW.



4. Configure the terminal.



In This Field	Do This
Comm Format	select Data - DINT
Connection Parameters	specify the input and output instances for this terminal You can establish up to eight different instances with each terminal. For example, one controller can use all eight instances. Or eight controllers can each use one instance.

Organize Controller Data for a PanelView Terminal

Organize data for a PanelView terminal based on how the data is used.

For Data That Is	Do This
Time-critical (for example, data that controls a machine)	Use the I/O tags of the terminal. The tags for this data were created when you added the terminal to the I/O configuration of the controller. They are similar to the tags of I/O modules.
Not time critical	Create arrays to store the data: 1. For each screen, create a BOOL array with enough elements for the bit-level objects on the screen. For example, the BOOL[32] array gives you 32 bits for push buttons, indicators, etc. 2. For each screen, create a DINT array with enough elements for the word-level objects on the screen. For example, the DINT[28] array, give you 28 values for numeric entry controls, numeric displays, etc.

To access the I/O tags of the PanelView or PanelView Plus terminal, use the following address format:

If the Terminal	Use This Address
Writes the data	name_of_terminal:I.Data[x].y
Reads the data	name_of_terminal:O.Data[x].y

where:

This Address Variable	Is
name_of_terminal	Name of the instance in the I/O configuration of the controller
x	Element of the input (I) or output (O) structure.
y	Bit number within the input or output element

Determine Connections to RSView Applications

How you establish communication to an RSView application depends on how you configure RSLinx software to collect tags from the controller. An RSView 32 or RSView Enterprise application use RSLinx software as a data server.

RSLinx Enterprise software defaults to 4 read connections and 1 write connection per configured controller. You can modify your RSLinx configuration as needed.

Notes:

Monitor Diagnostics

Use This Chapter

The EtherNet/IP communication modules provide several levels of diagnostics. There are user-oriented diagnostics, as well as more detailed diagnostics for technical support personnel. This chapter describes the diagnostics presented on the user-oriented diagnostic pages.

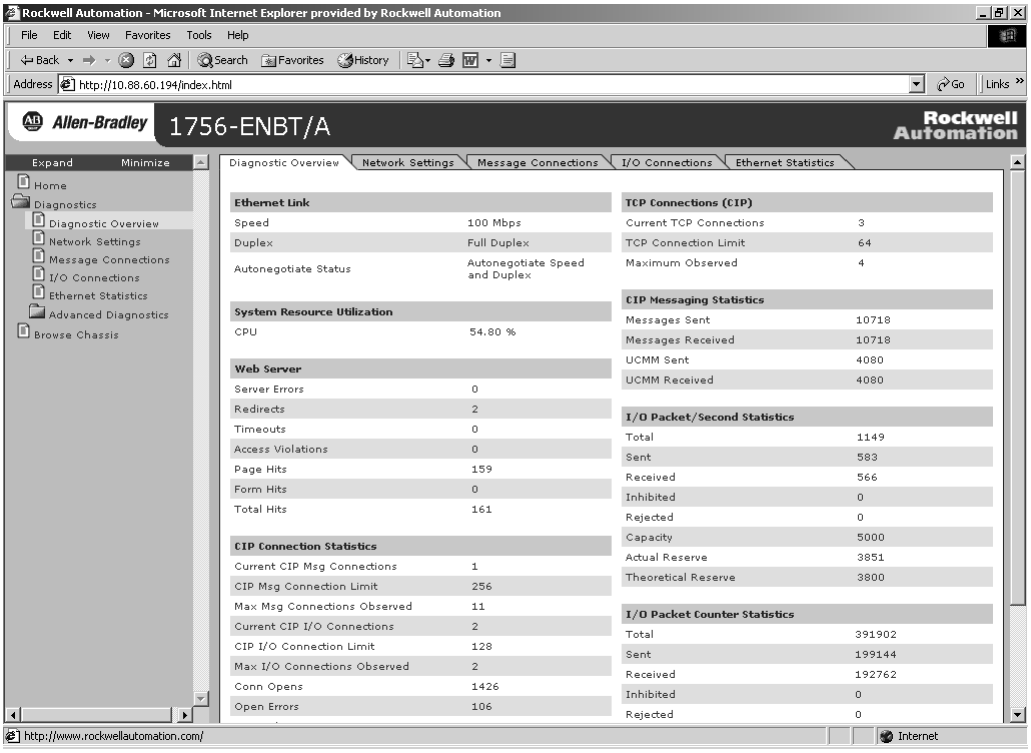
Module Diagnostics

The EtherNet/IP modules provide pages of user-oriented diagnostics. This information is organized into these web pages.

For This Information	Access This Web Page
Overview of the current configuration of the module	Diagnostics → Diagnostic Overview
Summary of the network settings configured for the module	Diagnostics → Network Settings
Statistics about messages initiated by the module and their associated connections	Diagnostics → Message Connections
Statistics about I/O modules associated with the module	Diagnostics → I/O Connections
Ethernet statistics	Diagnostics → Ethernet Statistics

Diagnostics Overview

The Diagnostics → Diagnostic Overview page presents a summary of the current configuration and overall status of the module.



This Field	Specifies
Ethernet Link	
Speed	Whether the Ethernet port is operating at 10 Mbps or 100 Mbps
Duplex	Whether the Ethernet port is operating at half duplex or full duplex
Autonegotiate Status	Whether the port speed and duplex mode were determined via autonegotiation or whether they were manually configured
System Resource Utilization	
CPU	Current percent CPU utilization for the module
Web Server	
Server Errors	Number of requests to the module with an invalid URL
Redirects	Number of requests for a web page that were redirected by the module (e.g., requesting "/" is redirected to "/index.html")
Timeouts	Number of times a connection timeout occurred while processing a web page
Access Violations	Number of times a page has been requested for which the user has insufficient privilege
Page Hits	Number of times a web page was successfully accessed
Form Hits	Number of times a web page form was accessed
Total Hits	Total number of web page access attempts
CIP Connection Statistics	
Current CIP MSG Connections	Current number of CIP connections for message
CIP MSG Connection Limit	Maximum number of CIP connections for messages allowed

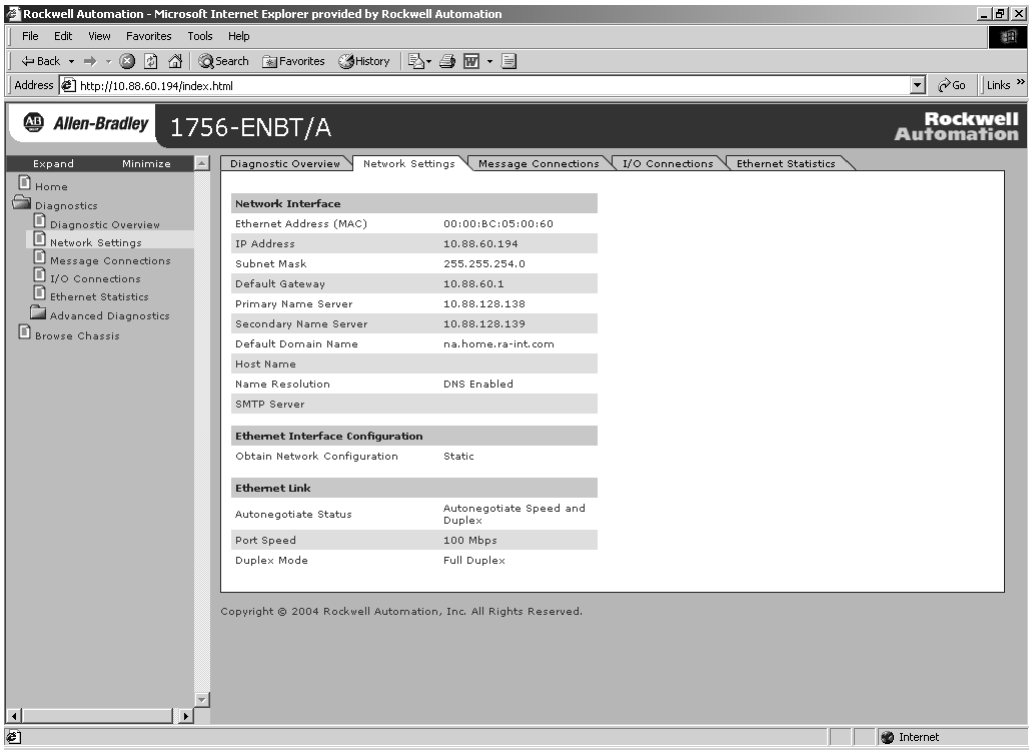
This Field	Specifies
Max MSG Connections Observed	Maximum observed number of CIP connections for messages
Current CIP I/O Connections	Current number of CIP connections for I/O
CIP I/O Connection Limit	Maximum number of CIP connections allowed for I/O
Max I/O Connections Observed	Maximum observed number of CIP connections for I/O
Conn Opens	Number of CIP connection open requests
Open Errors	Number of CIP connection open request errors
TCP Connections (CIP)	
Current TCP Connections	Current number of active TCP connections for CIP messaging
TCP Connection Limit	Maximum number of TCP connections for CIP messaging allowed
Maximum Observed	Maximum observed number of TCP connections for CIP messaging
CIP Messaging Statistics	
Messages Sent	Number of CIP connected messages (packets) sent
Messages Received	Number of CIP connected messages (packets) received
UCMM Sent	Number of CIP unconnected messages (packets) sent
UCMM Received	Number of CIP unconnected messages (packets) received
I/O Packet / Second Statistics	
Total	Total number of Class 1 UDP packets the module transmitted/received in the last 1 second snapshot The Total is the sum of the Sent, Received, Inhibited, and Rejected numbers.
Sent	Number of Class 1 UDP packets the module transmitted in the last 1 second snapshot
Received	Number of Class 1 UDP packets the module received in the last 1 second snapshot
Inhibited	Number of Class 1 UDP packets the module inhibited in the last 1 second snapshot Packets are inhibited if a COS module produces packets faster than 1/4 of the connection's RPI.
Rejected	Number of Class 1 UDP packets the module rejected in the last 1 second snapshot These packets were messages received and then rejected because the connection was closed or there was a duplicate multicast address.
Capacity	Number of Class 1 UDP packets the module can handle over the Ethernet network at any time
Actual Reserve	Actual Reserve = Capacity - Total This is based on the total of number packets the module has transmitted/received in the last 1 second snapshot.
Theoretical Reserve	Theoretical Reserve = Capacity - the sum of the theoretical packet/second of all connections based on the RPI
I/O Packet Counter Statistics	
Total	Cumulative number of Class 1 UDP packets the module transmitted/received The Total is the sum of the Sent, Received, Inhibited, and Rejected numbers
Sent	Cumulative number of Class 1 UDP packets the module transmitted

This Field	Specifies
Received	Cumulative number of Class 1 UDP packets the module received
Inhibited	Cumulative number of Class 1 UDP packets the module inhibited Packets are inhibited if a COS module produces packets faster than 1/4 of the connection's RPI.
Rejected	Cumulative number of Class 1 UDP packets the module rejected These packets were messages received and then rejected because the connection was closed or there was a duplicate multicast address.
Missed	Cumulative number packets that were not received in order Each UDP packet has a sequence number and if a packet is missing (corrupted or dropped), the module will recognize this void upon receipt of the next packet received. T Missed counter increments by the number of packets missed.

A CIP connection transfers data from one Logix application running on one end-node to a second Logix application running on another end-node. A CIP connection is established over a TCP connection.

Network Settings

The Diagnostics → Network Settings page presents a summary of the current Ethernet configuration for the module.



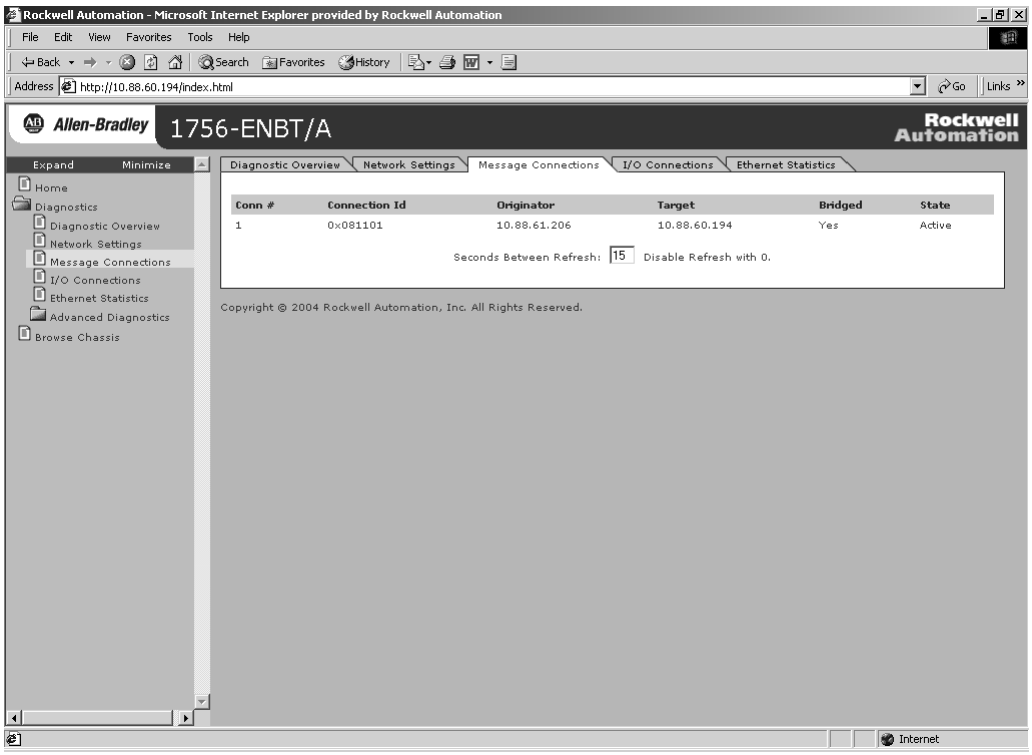
Any fields not configured remain blank.

This Field	Specifies
Network Interface	
Ethernet Address (MAC)	Ethernet (MAC) address of the module
IP Address	IP address for the module
Subnet Mask	Subnet mask for the module
Default Gateway	Gateway address for the module
Primary Name Server	Primary name server
Secondary Name Server	Secondary name server
Default Domain Name	Default domain name for the module
Host Name	Host name for the module
Name Resolution	Whether or not Domain Name System (DNS) resolution is enabled
SMTP Server	SMTP server address for the module (required for email)

This Field	Specifies
Ethernet Interface Configuration	
Obtain Network Configuration	Whether the module is configured to obtain its network parameters (IP address, etc.) via BOOTP, DHCP, or from static configuration
Ethernet Link	
Autonegotiate Status	Whether the Ethernet port is operating at 10 Mbps or 100 Mbps
Port Speed	Whether the Ethernet port is operating at half duplex or full duplex
Duplex Mode	Whether the port speed and duplex mode were determined via autonegotiation or whether they were manually configured

Explicit Message Connections

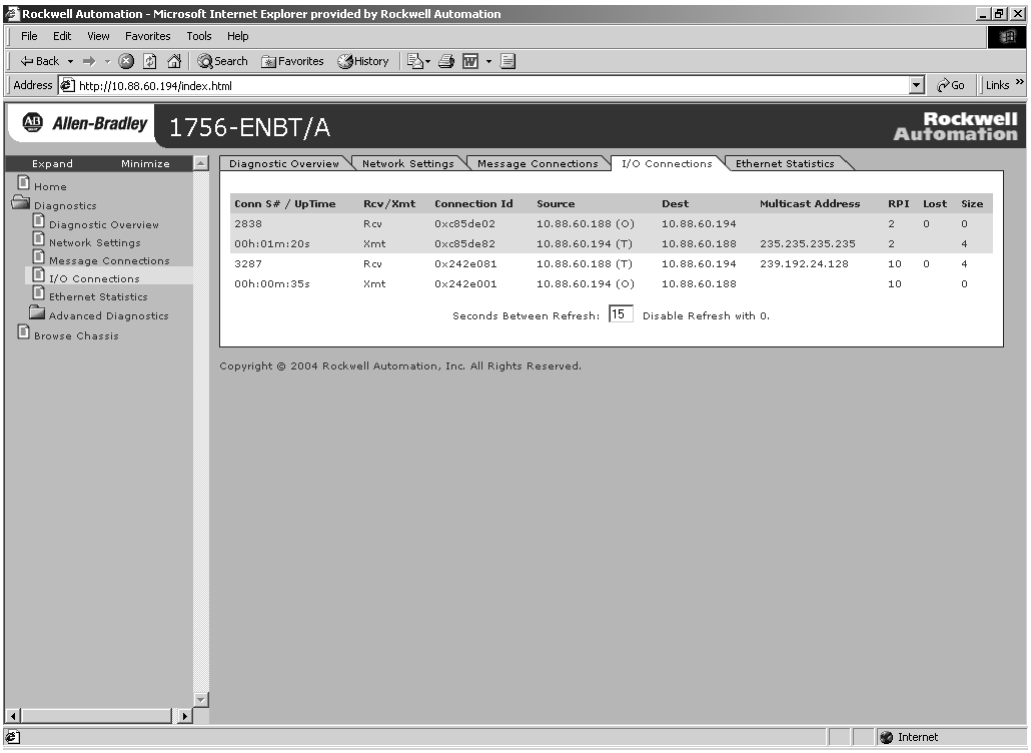
The Diagnostics → Message Connections page presents a summary of messages bridged through or initiated by the module.



This Field	Specifies
Conn #	The relative index of this connection (on the Message Connections page)
Connection ID	The unique identifier for each connection
Originator	The IP address of the device that originated the connection on Ethernet network
Target	The IP address of the device that is the target of the connection on Ethernet. This may not be the ultimate target of the connection for example, the target could be a Logix controller in a chassis).
Bridged	Whether the connection bridges through the module
State	The current state of the connection: • Active • Closing • Faulted • Reserved

I/O Connections

The Diagnostics → I/O Connections page presents a summary of I/O connections initiated by the module.



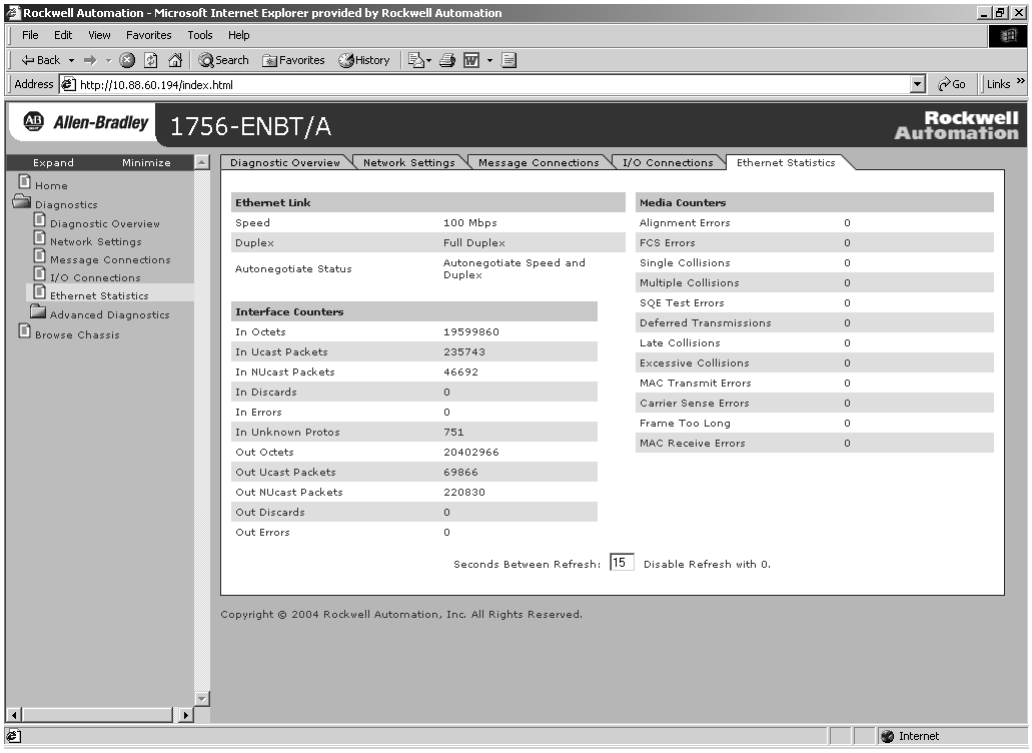
Each Class 1 UDP connection has a receive/transmit (Rcv/Xmt) pair of data and heartbeat. The originator of a connection listens on the multicast address to receive the data. The target of the connection receives the heartbeat.

In this example, the web page is for the module at address 10.88.60.194. This module (10.88.60.194) originated a connection to 10.88.60.188 with an RPI of 10.

This Field	Specifies
Conn S# / Up Time	Connection serial number and the elapsed time the connection has been maintained
Rcv / Xmt	Connection was received or transmitted from this source address
Connection ID	Connection identifier
Source	IP address of the Rcv/Xmt packet (T) = target; (O) = originator
Dest	Destination address
Multicast Address	Connection targets produce at this multicast address Connection originators listen on this multicast address
RPI	Programmed connection RPI
Lost	Total number of packets received where the Common Packet Encapsulation sequence number is less than the last received on this connection
Size	Size of class 1 UDP packet data (in bytes)

Ethernet Statistics

The Diagnostics → Ethernet Statistics page presents a summary of the status of communication activity on the Ethernet network.



This Field	Specifies
Ethernet Link	
Speed	Whether the Ethernet port is operating at 10 Mbps or 100 Mbps
Duplex	Whether the Ethernet port is operating at half duplex or full duplex
Autonegotiate Status	Whether the port speed and duplex mode were determined via autonegotiation or whether they were manually configured
Interface Counters	
In Octets	Octets received on the Ethernet interface
In Ucast Packets	Unicast packets received on the Ethernet interface
In NUcast Packets	Non-unicast packets received on the Ethernet interface
In Discards	Inbound packets received on the Ethernet interface but discarded
In Errors	Inbound packets that contain errors (does not include In Discards)
In Unknown Protos	Inbound packets with unknown protocol
Out Octets	Octets sent on the Ethernet interface
Out Ucast Packets	Unicast packets sent on the Ethernet interface
Out NUcast Packets	Non-unicast packets sent on the Ethernet interface
Out Discards	Outbound packets discarded
Out Errors	Outbound packets that contain errors

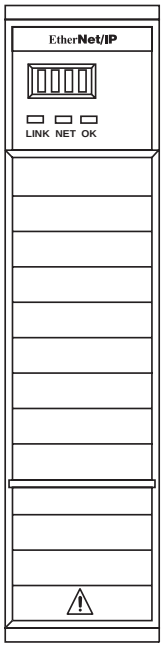
This Field	Specifies
Media Counters	
Alignment Errors	Frames received that are not an integral number of octets in length
FCS Errors	Frames received that do not pass the FCS check
Single Collisions	Successfully transmitted frames which experienced exactly one collision
Multiple Collisions	Successfully transmitted frames which experienced more than one collision
SQE Test Errors	Number of times SQE test error message is generated
Deferred Transmissions	Frames for which first transmission attempt is delayed because the medium is busy
Late Collisions	Number of times a collision is detected later than 512 bit-times into the transmission of a packet
Excessive Collisions	Frames for which transmission fails due to excessive collisions
MAC Transmit Errors	Frames for which transmission fails due to an internal MAC sublayer transmit error
Carrier Sense Errors	Times that the carrier sense condition was lost or never asserted when attempting to transmit a frame
Frame Too Long	Frames received that exceed the maximum permitted frame size
MAC Receive Errors	Frames for which reception on the Ethernet interface failed due to an internal MAC sublayer receive error

Module LED Indicators

Use This Appendix

This appendix provides LED indicator descriptions for the EtherNet/IP communication modules and adapters.

1756-ENBT EtherNet/IP Communication Module



Network (NET) status indicator

Condition	Indicates	Recommended Action
Off	Not powered, no IP address	Module is not powered, or does not have an IP address. • Verify there is chassis power and the module is completely inserted into the chassis and backplane. • Make sure the module has been configured.
Flashing green	No connections	Module has obtained an IP address, but has no established connections.
Green	CIP connections	Module has an IP address and at least one established connection.
Flashing red	Connection timeout	One or more of the connections in which the module is the target has timed out.
Red	Duplicate IP address	A duplicate IP address has been detected. Make sure that the the IP address assigned to this module is not the same as that for any other device already on the network

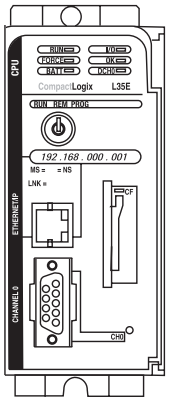
Link status indicator

Condition	Indicates	Recommended Action
Off	No data transmission	Module is not ready to communicate.
Green	Ready	Module is ready to communicate.
Flashing green	Data transmission in progress	Module is communicating over the network.

OK status indicator

Condition	Indicates	Recommended Action
Off	No power	Module does not have 24V DC power. Verify there is chassis power and the module is completely inserted into chassis and backplane.
Flashing green	Standby	Module is not configured.
Green	Operational	Module is operating correctly.
Flashing red	Duplicate IP address	A duplicate IP address has been detected. Make sure that the the IP address assigned to this module is not the same as that for any other device already on the network
	Minor fault	A recoverable fault has been detected. This could be caused by an error in the configuration.
Red	Major fault	An unrecoverable fault has been detected. Cycle power to the module. If this does not clear the fault, replace the module.
Flashing red/green	Self test	Module performing power-up self-test.

1769-L32E, 1769-L35E CompactLogix Controller



Module Status (MS) indicator

Condition	Indicates	Recommended Action
Off	The controller does not have power.	Check the controller power supply.
Flashing green	The controller does not have an IP address and is operating in BOOTP mode.	Verify that the BOOTP server is running.
Green	The controller is operating correctly.	Normal operation. No action required.
Red	The controller is holding the port in reset or the controller is faulted.	Clear the controller fault. If the fault will not clear, replace the controller.
	The controller is performing its power-up self-test.	Normal operation during power-up.
	An unrecoverable fault has occurred.	Cycle power to the controller. If the fault will not clear, replace the controller.
Flashing red	A duplicate IP address has been detected.	Make sure that the IP address assigned to this controller is not the same as that for any other device already on the network
	The controller firmware is being updated.	Normal operation during firmware update. No action required.

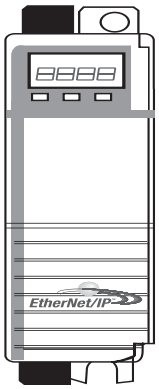
Network Status (NS) indicator

Condition	Indicates	Recommended Action
Off	The controller does not have an IP address and is operating in BOOTP mode.	Verify that the BOOTP server is running.
Flashing green	The controller has an IP address, but no CIP connections are established.	Normal operation if no connections are configured. No action required. If connections are configured, check connection originator for connection error code.
Green	The controller has an IP address and CIP connections (Class 1 or Class 3) are established.	Normal operation. No action required.
Red	A duplicate IP address has been detected.	Make sure that the IP address assigned to this module is not the same as that for any other device already on the network
Flashing red/green	The controller is performing its power-up self-test.	Normal operation during power-up.

Link Status (LNK) indicator

Condition	Indicates	Recommended Action
Off	The port is not connected to a powered Ethernet device. The controller cannot communicate on Ethernet.	Verify that all Ethernet cables are connected. Verify that Ethernet switch is powered.
Flashing green	The controller is performing its power-up self-test.	Normal operation during power-up.
	The controller is communicating on Ethernet.	Normal operation. No action required.
Green	The port is connected to a powered Ethernet device. The controller can communicate on Ethernet.	Normal operation. No action required.

1768-ENBT CompactLogix EtherNet/IP Communication Module



NET (Network) Status Indicator

Condition	Indicates	Recommended Action
Off	Not powered, No IP address	Module is not powered, or does not have an IP address. • Verify there is chassis power and the module is completely inserted into the chassis and backplane. • Make sure the module has been configured.
Flashing green	No connections	Module has obtained an IP address, but has no established connections.
Green	CIP connections	Module has an IP address and at least one established connection.
Flashing red	Connection timeout	One or more of the connections in which the module is the target has timed out.
Red	Duplicate IP address	Module has detected that its IP address is already in use. Assign a unique IP address to the module.

Link Status Indicator

Condition	Indicates	Recommended Action
Off	No data transmission	Module is not ready to communicate.
Green	Ready	Module is ready to communicate.
Flashing green	Data transmission in progress	Module is communicating over the network.

OK Status Indicator

Condition	Indicates	Recommended Action
Off	No power	Module does not have 5V dc power. Verify there is power and the module is properly installed.
Flashing Green	Standby	Module is not configured.
Green	Operational	Module is operating correctly.
Flashing Red	Minor fault	A recoverable fault has been detected. This could be caused by an error in the configuration.
Red	Major fault	An unrecoverable fault has been detected. Recycle power to the module. If this does not clear the fault, replace the module.
Flashing Red and Green	Self-test	Module performing power-up self-test.

1788-ENBT EtherNet/IP Communication Daughtercard

Module status (MS) indicator

Condition	Indicates	Which Means	Recommended Action
Off	No power	The daughtercard does not have power.	Check the host power supply. Verify that the daughtercard is firmly seated in the host's slot. Replace daughtercard and/or host.
Flashing green	Standby	The daughtercard does not have an IP address and is operating in BOOTP mode.	Verify that the BOOTP server is running.
Green	Ok	The daughtercard is operating correctly.	Normal operation. No action required.
Red	Held in reset	The host is holding the daughtercard in reset or the host is faulted.	Verify that the daughtercard is firmly seated in the host's slot. Clear the host's fault. Replace daughtercard and/or host.
	Self-test	The daughtercard is performing its power-up self-test.	Normal operation during power-up.
	Major fault	An unrecoverable fault has occurred.	Cycle power to the host. Replace the daughtercard and/or host.
Flashing red	Duplicate IP address	A duplicate IP address has been detected.	Make sure that the IP address assigned to this module is not the same as that for any other device already on the network
	Updating firmware	The daughtercard firmware is being updated.	Normal operation during firmware update. No action required.

Network status (NS) indicator

Condition	Indicates	Which Means	Recommended Action
Off	Not initialized	The daughtercard does not have an IP address and is operating in BOOTP mode.	Verify that the BOOTP server is running.
Flashing green	No CIP connections established	The daughtercard has an IP address, but no CIP connections are established.	Normal operation if no connections are configured. No action required. If connections are configured, check connection originator for connection error code.
Green	CIP connections established	The daughtercard has an IP address and CIP connections (Class 1 or Class 3) are established.	Normal operation. No action required.
Red	Duplicate IP address	A duplicate IP address has been detected.	Make sure that the IP address assigned to this module is not the same as that for any other device already on the network

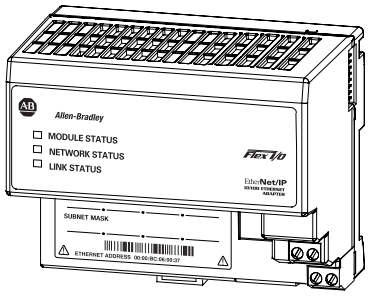
Link status (LNK) indicator

Condition	Indicates	Which Means	Recommended Action
Off	No link	The daughtercard is not connected to a powered Ethernet module. The daughtercard cannot communicate on Ethernet.	Verify that all Ethernet cables are connected. Verify that Ethernet switch is powered.
Flashing green	Self-test	The daughtercard is performing its power-up self-test.	Normal operation during power-up.
	Data transmission and reception	The daughtercard is communicating on Ethernet.	Normal operation. No action required.
Green	Link okay	The daughtercard is connected to a powered Ethernet module. The daughtercard can communicate on Ethernet.	Normal operation. No action required.

Utilization percent (U%) indicator

Condition	Indicates	Which Means	Recommended Action
off	less than 80% utilization	The I/O packet rate to/from this daughtercard is less than 80% of the available packet rate and less than 80% of the 32 available I/O connections are currently in use.	Normal operation. No action required.
flashing green	80% packet rate reached	The I/O packet rate to/from this daughtercard is at least 80% of the available packet rate (4000 packets/sec.). That is, the I/O packet rate is at least 3200 packets/sec.	Normal operation. No action required.
	80% connections in use	At least 80% of the 32 available connections are currently in use. That is, from 26 to 31 I/O connections are in use.	Normal operation. No action required.
green	all connections in use	All 32 of the daughtercard's I/O connections are currently in use.	Normal operation when all 32 I/O connections are in use.
flashing red/green	self-test	The daughtercard is performing its power-up self-test.	Normal operation during power-up.

1794-AENT EtherNet/IP FLEX I/O Adapter



1Module status indicator

Condition	Indicates	Recommended Action
Off	No power	Module does not have 24V DC power. Make sure power is being supplied to the module.
Flashing green	Standby	Module not configured.
Green	Operational	Module operating correctly.
Flashing red	Minor fault	A recoverable fault has been detected. This could be caused by an incorrect or inconsistent configuration.
Red	Major fault	An unrecoverable fault has been detected. Recycle power to the module. If this does not clear the fault, replace the module.
Flashing red/green	Self test	Module performing power-up self test.

Network status indicator

Condition	Indicates	Recommended Action
Off	Not powered, No IP address	Module is not powered, or does not have an IP address. Verify there is power and the module is correctly wired to the power supply. Make sure the module is configured
Flashing green	No connections	Module has obtained an IP address, but has no established connections.
Green	CIP connections	Module has an IP address and at least one established connection.
Flashing red	Connection timeout	One or more of the connections in which the module is the target has timed out.
Flashing red/green	Self test	Module performing power-up self test.

Link status indicator

Condition	Indicates	Recommended Action
Off	No data transmission	Module not ready to communicate.
Flashing green	Ready	Module is ready to communicate.
Green	Data transmission in progress	Module is communicating over the network.

Notes:

EtherNet/IP Network Connections

Use This Appendix

Read this chapter for:

- *1756-ENBT module*
- *1769-L32E, -L35E controller*
- *1768-ENBT module*
- *1788-ENBT card*
- *1794-AENT adapter*

EtherNet/IP communication modules use connections to manage communications. A connection is a point-to-point communication mechanism used to transfer data between a transmitter and a receiver. The EtherNet/IP communication modules use these connections:

- CIP connections for Logix-based communications

  A CIP connection transfers data from one Logix application running on one end-node to a second Logix application running on another end-node. A CIP connection is established over a TCP connection.

- TCP/IP connections for EtherNet/IP communications

  A single TCP connection can support multiple CIP connections.

CIP Connections

Connections are allocations of resources that provide more reliable communications between modules than unconnected messages.

Examples of functions supported by CIP implicit (connected) messaging include:

- Logix controller message transfer to Logix controller
- I/O or produced/consumed tag
- Program upload
- RSLinx DDE/OPC client
- PanelView polling of Logix controller

There are several types of CIP connections:

CIP Connection Type	Description
Bridged connection	A bridged connection is a connection that passes through the EtherNet/IP module. The end point of the connection is a module other than the EtherNet/IP module. **Example:** a connection from a controller through a 1756-ENBT to another controller.
End-node connection	An end-node connection is a connection whose end point is the EtherNet/IP module itself. **Example:** a connection from RSLinx to the EtherNet/IP module to set the module's IP address.
Rack-optimized	A rack-optimized connection is a connection to a rack or assembly object in the EtherNet/IP module. Data from selected I/O modules is collected and produced on one connection (the rack-optimized connection) rather than on a separate direct connection for each module.
Direct	A connection from a controller to an specific I/O module (as opposed to a rack-optimized connection).

The Logix5000 controller supports 250 connections. But the limit of connections ultimately resides in the communication module you use for the connection. If a message path routes through a communication module or card, the connection related to the message also counts towards the connection limit of the communication module or card.

CIP implicit (connected) messaging limits

An implicit (connected) connection is time critical in nature. This includes I/O and produced/consumed tags.

Product:	CIP Connected Messaging Limits:
1756-ENBT	Each module supports: • 128 bridged connections of which 32 connections can be end-node connections In addition to the CIP connections: • 16 controllers can have a rack-optimized connection to the module • 16 controllers can have a rack-optimized, listen-only connection to the module • 64 controllers can consume data from a connection

Product:	CIP Connected Messaging Limits:
1768-ENBT	Each module supports 64 bridged messages, of which 32 connections can be end-node connections In addition to the CIP connections, 32 controllers can consume data from a connection
1788-ENBT	Each module supports 32 bridged connections, of which 20 connections can be end-node connections In addition to the CIP connections, 32 controllers can consume data from a connection
1794-AENT	Each module supports: • 32 end-node connections for messages • 31 end-node connections for I/O and produced/consumed tags • no bridged connections In addition to the CIP connections, 31 controllers can consume data from a direct connection

CIP explicit (unconnected) messaging limits

An explicit (unconnected) connection is non-time critical and is request/reply in nature. The following limits of unconnected messages are the maximum number of outstanding unconnected messages. These are unconnected messages that have been sent to the module and are being processed and have not yet generated a response or timeout.

Product	CIP Unconnected Messaging Limits
1756-ENBT	Each module supports 256 CIP unconnected messages, of which: • 128 can be unconnected messages from the EtherNet/IP port to an object on the module or to the backplane. • 128 can be unconnected messages from the backplane to an object on the module or to the EtherNet/IP port.
1768-ENBT	Each module supports 64 CIP unconnected messages, of which: • 32 can be unconnected messages from the EtherNet/IP port to the host. • 32 can be unconnected messages from the host to the EtherNet/IP port.
1788-ENBT	Each module supports 64 CIP unconnected messages, of which: • 32 can be unconnected messages from the EtherNet/IP port to the host. • 32 can be unconnected messages from the host to the EtherNet/IP port.
1794-AENT	Each module has a maximum of 256 CIP unconnected messages from the EtherNet/IP port. The 1794-AENT can receive messages from the EtherNet/IP port. Because the FLEX I/O backplane uses a polled architecture, the FLEX modules I/O modules do not initiate messages to the 1794-AENT module.

EtherNet/IP connections are not scheduled as on other networks, such as using RSNetWorx software to schedule ControlNet connections, but the EtherNet/IP connections do exchange data at a regular time interval, the requested packet interval (RPI).

See the EtherNet/IP Performance and Application Guide, publication ENET-AP001, for more information on connections.

TCP Connections

An EtherNet/IP module uses one TCP connection for each IP address to which the EtherNet/IP module is connected. Multiple CIP connections can go through a single TCP connection. Examples of TCP connections are:

- HMI (human-machine interface) to a controller that supports EtherNet/IP communications
- Logix MSG instruction to a controller or workstation
- OPC or DDE accessing a controller
- I/O data
- Produced or consumed tag

The 1756-ENBT, 1788-ENBT, and 1794-AENT modules each support 64 TCP connections.

Multicast Address Limit

Connections that produce data over an Ethernet network use multicast addresses. EtherNet/IP modules support a maximum of 32 unique multicast addresses. The actual address (such as 239.192.22.121) is determined by the EtherNet/IP module.

Example 1: An Ethernet adapter that produces data uses a unique multicast address for each I/O connection.

Example 2: A Logix controller that produces tags uses a unique multicast address for each produced tag.

The multicast address limit is independent of the connection limit for a module. Not all connections require a multicast address. And in the case of produced and consumed tags, one produced tag requires one multicast address but it also requires one connection for each consumer. If there are multiple consumers, the one multicast address would be using multiple connections.

Specify the Requested Packet Interval (RPI)

The RPI is the update rate specified for a particular piece of data on the network. The RPI can be specified for an entire rack (using a rack-optimized connection) or for a particular module (using a direct connection).

When you add a module to the I/O configuration of a controller, you must configure the RPI. This value specifies how often to produce the data for that module. For example, if you specify an RPI of 50ms, every 50ms the I/O module sends its data to the controller or that the controller sends its data to the I/O module.

RPIs are used only for implicit connections, such as produced/consumed tags and I/O. For example, a local EtherNet/IP communication module does not require an RPI because it is not a data-producing member of the system; it is used only as a bridge to remote modules.

Set the RPI only as fast as needed by the application. The RPI also determines the number of packets per second that the module will produce on a connection.

Each module has a limit on the total number of implicit packets per second. The total includes the sum of sent and received implicit packets. The packet rate for implicit messages is for implicit only, and is not the same as and does not include the explicit packet rate.

Notes:

EtherNet/IP Network Overview

Use This Appendix

This appendix defines some basic Ethernet network concepts and how the EtherNet/IP protocol is used for control.

Ethernet Protocols

On the most basic level, Ethernet is a wire or cable that connects computers and peripheral modules so that they can communicate. The actual wire used for the network is referred to as the network medium. Beyond the physical medium, all Ethernet networks support protocols that provide sophisticated data transfer and network management functionality.

Protocol Descriptions

Protocol	Description
Transmission control protocol/internet protocol (TCP/IP)	TCP/IP is a transport-layer protocol (TCP) and a network-layer protocol (IP) commonly used in business environments for communication within networks and across internetworks. The EtherNet/IP communication modules use TCP/IP for explicit messaging, that is, messages in which time is not a critical factor, such as uploading or downloading programs.
User datagram protocol/Internet protocol (UDP/IP)	UDP is a much simpler transport protocol. It is connectionless and provides a very simple capability to send datagrams between two modules. UDP is used by applications that implement their own handshaking between modules and only want a minimal transport service. UDP is smaller, simpler, and faster than TCP and can operate in unicast, multicast, or broadcast mode. The EtherNet/IP communication modules use UDP/IP for real time I/O messaging.
CIP	CIP applies a common application layer over Ethernet by encapsulating messages in TCP/UDP/IP. This common application layer is the control and information protocol (CIP), which provides interoperability and interchangeability of industrial automation and control modules on Ethernet. EtherNet/IP supports both real-time I/O (implicit messaging) and explicit messaging. See the EtherNet/IP Performance and Application Guide, publication ENET-AP001, for more information on EtherNet/IP.
Simple network management protocol (SNMP)	SNMP is a standard for network management within TCP/IP environments. This lets client applications monitor and manage network information on host computers and gateways. This protocol is password-protected. SNMP uses a distributed architecture consisting of management systems and agents. Data is passed from SNMP agents, which are hardware and/or software processes reporting activity in each network module (switch, router, bridge, etc.) to the workstation console used to oversee the network. The agents return information contained in a MIB (management information base), which is a data structure that defines what is obtainable from the module and what can be controlled (turned off, on, etc.).
Internet Group Management protocol (IGMP) snooping	IGMP snooping enables switches to route multicast traffic by distributing each packet only to the ports that need to receive it. Many switches support this feature. However, most of these switches require a router be present in the system for IGMP snooping to work. If your control system is a stand-alone network or is required to continue performing if the router is out of service, make sure the switch you are using supports IGMP snooping without a router present. This feature is highly recommended for EtherNet/IP systems the control I/O.

Use of the Common Industrial Protocol (CIP)

The EtherNet/IP communication modules use the Common Industrial Protocol (CIP). CIP is the application layer protocol specified for EtherNet/IP, the Ethernet Industrial Protocol, as well as for ControlNet and DeviceNet.

CIP is a message-based protocol that implements a relative path to send a message from the producing module in a system to the consuming modules.

The producing module contains the path information that steers the message along the proper route to reach its consumers. Since the producing module holds this information, other modules along the path simply pass this information; they do not need to store it. This has two significant benefits:

- You do not need to configure routing tables in the bridging module, which greatly simplifies maintenance and module replacement.
- You maintain full control over the route taken by each message, which enables you to select alternative paths for the same end module.

CIP uses the producer/consumer networking model instead of a source/destination (master/slave) model. The producer/consumer model reduces network traffic and increases speed of transmission.

In traditional I/O systems, controllers poll input modules to obtain their input status. In the CIP system, digital input modules are not polled by a controller. Instead, they produce (multicast) their data either upon a change of state (COS) or periodically. The frequency of update depends upon the options chosen during configuration and where on the network the input module resides. The input module, therefore, is a producer of input data and the controller is a consumer of the data.

The controller can also produce data for other controllers to consume. The produced and consumed data is accessible by multiple controllers over the Logix backplane and over the EtherNet/IP network. This data exchange conforms to the producer/consumer model.

Configuration Requirements

Before you can use an EtherNet/IP module, you must configure its IP address, gateway address, and subnet mask.

IP Address

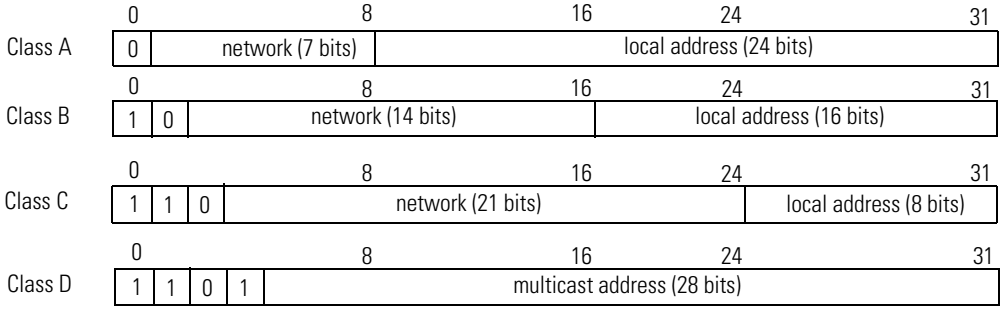
The IP address identifies each node on the IP network (or system of connected networks). Each TCP/IP node on a network must have a unique IP address.

IMPORTANT Contact your network administrator or the Network Information Center for a unique fixed IP address to assign to the EtherNet/IP module.

The IP address is 32 bits long and has a network ID part and a host ID part. Because networks vary in size, there are four classes (formats) of networks.

Network Type	For
Class A	Large networks with many devices
Class B	Medium-sized networks
Class C	Small networks (fewer than 256 devices)
Class D	Multicast addresses

The network class determines how an IP address is formatted.



Each node on the same physical network must have an IP address of the same class and must have the same network ID. Each node on the same network must have a different local address (host ID) thus giving it a unique IP address.

IP addresses are written as four decimal integers (0-255) separated by periods where each integer gives the value of one byte of the IP address.

For example, the 32-bit IP address:

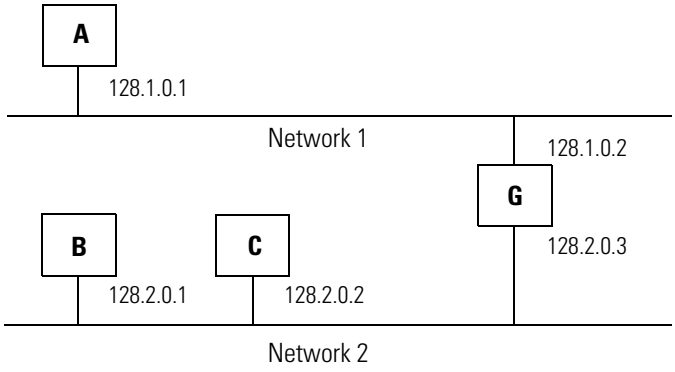
10000010 00000000 00000000 00000001 is written as 130.0.0.1.

You can distinguish the class of an IP address from the first integer in its IP address as follows:

Class	Leftmost Bits	Start Address	Finish Address
A	0xxx	0.0.0.	127.255.255.255
B	10xx	128.0.0.0	191.255.255.255
C	110x	192.0.0.0	223.255.255.255
D	1110	224.0.0.0	239.255.255.255

Gateways

A gateway connects individual physical networks into a system of networks. When a node needs to communicate with a node on another network, a gateway transfers the data between the two networks. The following figure shows gateway G connecting Network 1 with Network 2.



When host B with IP address 128.2.0.1 communicates with host C, it knows from C's IP address that C is on the same network. In an Ethernet environment, B can then resolve C's IP address to a MAC address and communicate with C directly.

When host B communicates with host A, it knows from A's IP address that A is on another network (the network IDs are different). To send data to A, B must have the IP address of the gateway connecting the two networks. In this example, the gateway's IP address on Network 2 is 128.2.0.3.

The gateway has two IP addresses (128.1.0.2 and 128.2.0.3). The first must be used by hosts on Network 1 and the second must be used by hosts on Network 2. To be usable, a host's gateway must be addressed using a net ID matching its own.

Subnet Mask

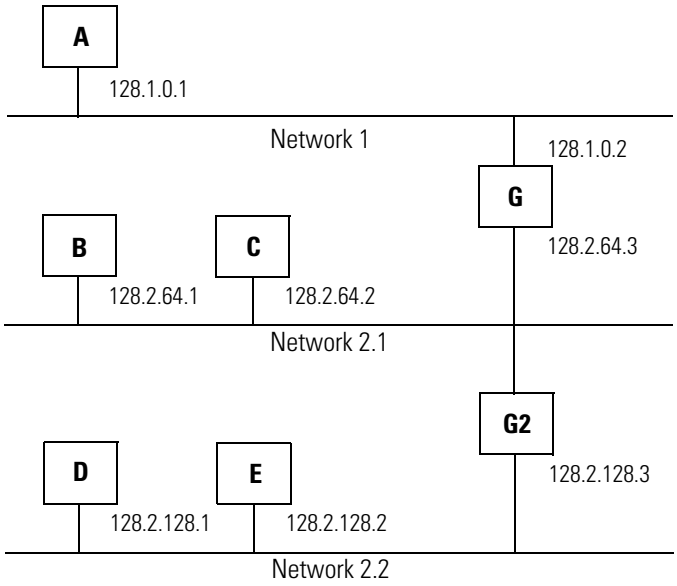
Subnet addressing is an extension of the IP address scheme that allows a site to use a single net ID for multiple physical networks. Routing outside of the site continues by dividing the IP address into a net ID and a host ID via the class. Inside a site, the subnet mask is used to redivide the IP address into a custom net ID portion and host ID portion.

Take Network 2 (a Class B network) in the previous example and add another physical network. Selecting the following subnet mask would add two additional net ID bits allowing for four physical networks:

11111111 11111111 11000000 00000000 = 255.255.192.0

Two bits of the Class B host ID have been used to extend the net ID. Each unique combination of bits in the part of the host ID where subnet mask bits are 1 specifies a different physical network.

The new configuration is:



A second network with Hosts D and E has been added. Gateway G2 connects Network 2.1 with Network 2.2. Hosts D and E will use Gateway G2 to communicate with hosts not on Network 2.2. Hosts B and C will use Gateway G to communicate with hosts not on Network 2.1. When B is communicating with D, G (the configured Gateway for B) will route the data from B to D through G2.

Manual Configuration on an Ethernet Switch

The EtherNet/IP modules support the following Ethernet settings:

- 10 Mbps half duplex or full duplex
- 100 Mbps half duplex or full duplex

Mode selection can be automatic, based on the IEEE 802.3 autonegotiation protocol. Or, with RSLogix 5000 programming software version 12 and later, you can manually set the communication rate and duplex mode of the communication module and the switch port that is connected to the module. If you manually set the communication rate and duplex mode, the settings of the communication module and the switch port must match.

In most cases, autonegotiation results in proper operation between a switch port and an EtherNet/IP module. However, when troubleshooting a network, you can force duplex and speed settings first at the EtherNet/IP module and then at the switch port to eliminate system variables.

Change Ports on an Ethernet Switch

If you reconnect the EtherNet/IP module from one port to another one, regardless whether the new port is located on the same or a different switch (or a hub), do the following:

1. Disconnect the cable from the port to which the EtherNet/IP module is currently connected.
2. Wait until the EtherNet/IP module Link Status LED is off.
3. Connect the cable to the new port.

This procedure restarts the autonegotiation process at the EtherNet/IP module side. Another option is to restart the EtherNet/IP module itself.

For More Information

For more information about TCP/IP and Ethernet technologies, see these publications:

Publication Title:	ISBN Number:
Internetworking with TCP/IP Volume 1: Protocols and Architecture, 2nd ed. by Douglas E. Comer	ISBN 0-13-216987-8
The Ethernet Management Guide – Keeping The Link	ISBN 0-07-046320-4
An Introduction to TCP/IP	ISBN 3-540-96651-X
Computer Networks by Andrew S. Tanenbaum	ISBN 0-13-162959-X

How Are We Doing?

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

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

Pub. Title/Type EtherNet/IP Modules in Logix5000 Control Systems User Manual

Cat. No. 1756-ENBT, 1788-ENBT, 1769-L32E, 1768-ENBT, 1794-AENT Pub. No. ENET-UM001E-EN-P Pub. Date January 2006 Part No. 953002-06

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

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

Your Name _____

Your Title/Function _____

Location/Phone _____

Would you like us to contact you regarding your comments?

___No, there is no need to contact me

___Yes, please call me

___Yes, please email me at _____

___Yes, please contact me via _____

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

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

PLEASE FASTEN HERE (DO NOT STAPLE)

Other Comments

PLEASE FOLD HERE

PLEASE REMOVE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 18235 CLEVELAND OH

POSTAGE WILL BE PAID BY THE ADDRESSEE

Allen-Bradley
RELIANCE ELECTRIC
DODGE
ROCKWELL SOFTWARE
Rockwell Automation

1 ALLEN-BRADLEY DR
MAYFIELD HEIGHTS OH 44124-9705

Rockwell Automation Support

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

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

Installation Assistance

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

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

New Product Satisfaction Return

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

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

www.rockwellautomation.com

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

Headquarters for Allen-Bradley Products, Rockwell Software Products and Global Manufacturing Solutions
Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe: Rockwell Automation SA/NV, Vorstlaan/Boulevard du Souverain 36-BP 3A/B, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Headquarters for Dodge and Reliance Electric Products
Americas: Rockwell Automation, 6040 Ponders Court, Greenville, SC 29615-4617 USA, Tel: (1) 864.297.4800, Fax: (1) 864.281.2433
Europe: Rockwell Automation, Brühlstraße 22, D-74834 Elztal-Dallau, Germany, Tel: (49) 6261 9410, Fax: (49) 6261 17741
Asia Pacific: Rockwell Automation, 55 Newton Road, #11-01/02 Revenue House, Singapore 307987, Tel: (65) 351 6723, Fax: (65) 355 1733

Publication ENET-UM001E-EN-P - January 2006 PN 953002-06
Supersedes Publication ENET-UM001D-EN-P - July 2005