DEVICE•MASTER®
EIP

DEVICE•MASTER®
UP

EtherNet®/IP User Guide

COMTROL®

Trademark Notices

Thirteenth Edition, November 28, 2017

Comtrol Corporation makes no representations or warranties with regard to the contents of this document or to the suitability of the Comtrol product for any particular purpose. Specifications subject to change without notice. Some software or features may not be available at the time of publication. Contact your reseller for current product information.

Document Number: 2000424 Rev M

Patents Pending

Table of Contents

Chapter 1. Introduction

This *User Guide* provides detailed information about the following topics:

- Programming Interface on Page 19
- Embedded Configuration Pages on Page 85
- Diagnostic and Statistics Pages on Page 135

The *EtherNet/IP Hardware Installation and Configuration Guide* provides the following information:

- Connecting the hardware and devices
- Programming the DeviceMaster UP or DeviceMaster EIP IP address,
- Uploading EtherNet/IP firmware

The *EtherNet/IP Interface Configuration Quick Start* provides embedded web page configuration procedures if you have *Read-only or read/write* devices, which provides procedures for your devices.

See *Locating the Latest Software and Documents* on Page 8 to locate the latest firmware, documentation, and tools.

1.1. Product Name Change Notification

Comtrol has implemented a product name change for the DeviceMaster UP 2-port DIN rail models that align with our new 1-port and 4-port DIN rail model names.

Note: *Please note that the form, fit, and function of the DeviceMaster has not changed - only the name.*

Old Name/Description	New Model Name
DeviceMaster UP 2-Port 1E EtherNet/IP	DeviceMaster EIP-2202
DeviceMaster UP 2-Port DB9 1E EtherNet/IP	DeviceMaster EIP-2102
DeviceMaster UP 2-Port 2E EtherNet/IP	DeviceMaster EIP-2402
DeviceMaster UP 2-Port DB9 2E EtherNet/IP	DeviceMaster EIP-2302

In this manual, the DeviceMaster UP or DeviceMaster EIP is simply referred to as the DeviceMaster.

1.2. Audience

The primary audience of this document is the person responsible for installing the DeviceMaster and programming the PLC. This guide assumes you are familiar with the following topics:

- Windows operating system
- EtherNet/IP
- Allen-Bradley ControlLogix family, PLC-5, SLC or MicroLogix PLCs
- RSLogix 5000, RSLogix 500 or RSLogix 5 programs

1.3. Product Overview

Your particular DeviceMaster model may or may not have the EtherNet/IP firmware loaded (depending on the model you purchased).

Note: *Models that have EtherNet/IP loaded on the DeviceMaster are identified in PortVision DX and the DeviceMaster is labeled accordingly.*

This document describes how to configure the DeviceMaster for the EtherNet/IP protocol after basic DeviceMaster installation and configuration procedures.

You can configure and manage the DeviceMaster through one of the following methods:

- Embedded web page interface
- EtherNet/IP Interface Profile objects

1.4. EtherNet/IP Firmware

The following subsections provide EtherNet/IP system information.

1.4.1. Locating the Latest Software and Documents

You can use the links in this table to check for updated software or documentation.

Software and Documentation		FTP
PortVision DX (Windows)	Use PortVision DX to manage Comtrol Ethernet-attached devices to: • Scan the network for attached devices • View networked devices in real-time • Access product-specific network settings configurations • Assign IP addresses and network settings to one or multiple devices • Upload the latest firmware or Bootloader • Save and load configuration files • Access DeviceMaster configuration web pages • Access Telnet/SSH sessions • Remotely reboot devices • Download technical documentation • Enable event logging to assist in monitoring and troubleshooting • Create shortcuts to quickly access your favorite applications • Organize devices into folders and create multiple views • Enter notes about a folder or device	🌐
EtherNet/IP Firmware	This is the application that may or may not have been loaded on the DeviceMaster depending on the model that was ordered. You may need to use PortVision DX to load this firmware.	🌐

Software and Documentation		FTP
EtherNet/IP Hardware Installation and Configuration Guide	This contains hardware installation, configuration information, and connector information. This includes using PortVision DX to configure the IP address and if necessary, how to update the firmware.	🌐
EtherNet/IP Interface Configuration Quick Start	This document with web interface configuration procedures.	🌐
EtherNet/IP User Guide	The *User Guide* contains detailed information about the EtherNet/IP (application) firmware, which includes additional information about the web configuration interface for EtherNet/IP. It also discusses the example PLC programs that were installed on your system and provides a *Programming Interface* chapter.	🌐
DeviceMaster UP *Filtering and Data Extraction Reference Guide*	This *Guide* discusses the data extraction and filtering processes in the DeviceMaster are designed to off load as much work as possible from the PLC and/or application and provide a very simple and easy to use interface for standard RFID and barcode data. This functionality and interface is designed to save dozens, possibly hundreds of lines of ladder logic in a typical PLC program.	🌐

1.4.2. Data Type Definitions

The following list defines the available data types.

Data Type	Definition
USINT	Unsigned short integer (8-bits)
UINT	Unsigned integer (16-bit)
UDINT	Unsigned double integer (32-bits)
INT	Signed integer (16-bits)
DINT	Signed double integer (32-bits)
BYTE	Bit string (8-bits)
WORD	Bit string (16-bits)
DWORD	Bit string (32-bits)
STRING	Character string (1-byte per character)

1.4.3. Terms and Definitions

This section uses the following terms and definitions.

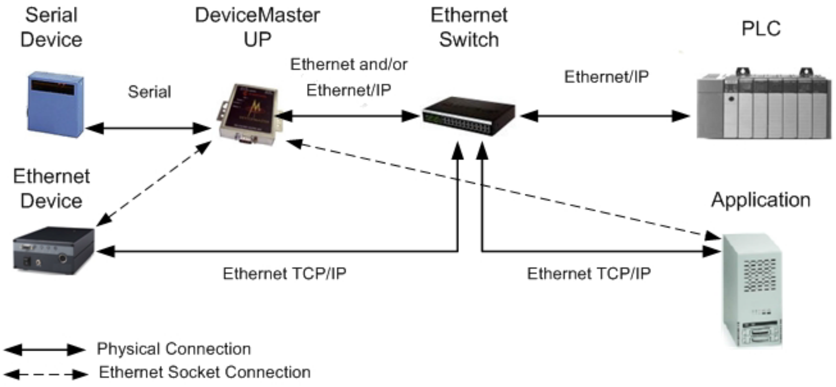
Term	Definition
Class 1	Otherwise called implicit messaging, is a method of communication between EtherNet/IP controllers and devices that: • Uses Ethernet UDP messages. • Is cyclic in nature. Input and/or output data is exchanged between the controllers and devices at regular time intervals.
Class 3	Otherwise called explicit messaging, is a method of communication between EtherNet/IP controllers and devices that: • Uses Ethernet TCP/IP messages. • By itself is not cyclic in nature. The controller and devices must send individual messages to each other.
EtherNet/IP	An Ethernet based industrial communication protocol utilized to communicate between controllers, often times PLCS, and devices.
Ethernet TCP/IP	Standard Ethernet communications protocol utilizing socket communication interfaces that guarantees delivery to the intended device.
Ethernet UDP/IP	Standard Ethernet communications protocol utilizing socket communication interfaces that does not guarantee delivery. The data may or may get to the intended device.
Multicast	Multicast addressing involves Ethernet devices sending messages to each other using a multicast address. Multicast addressing: • Uses a specified IP address range designated for multicast communication. • Allows either one or multiple devices to receive the same messages.
Point-to-Point	Point-to-Point, otherwise called unicast, addressing involves Ethernet devices sending messages directly to each other using their own IP addresses. Messages are sent to only one device.

1.4.4. EtherNet/IP System Architecture

The Ethernet TCP/IP firmware provides a raw/ASCII interface to both serial and Ethernet TCP/IP devices.

For example:

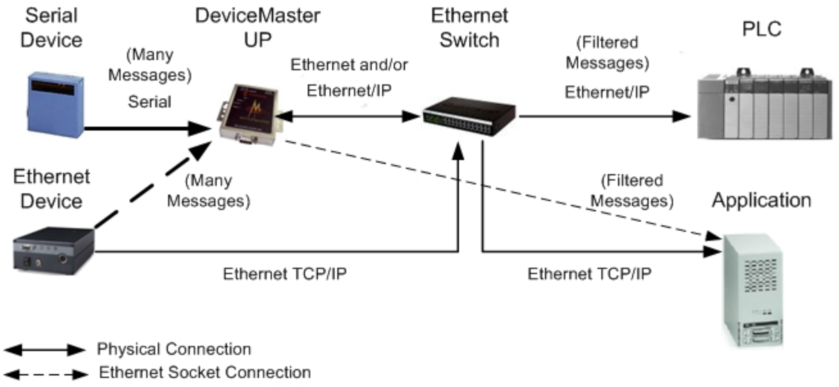
- The DeviceMaster 1-port provides EtherNet/IP support for one serial device and one Ethernet device for a total of two devices.
- The DeviceMaster 2-port provides EtherNet/IP support for two serial devices and two Ethernet devices for a total of four devices.
- The DeviceMaster 4-port provides EtherNet/IP support for four serial devices and four Ethernet devices for a total of eight devices.



Comtrol EtherNet/IP System Architecture (V3.x)

EtherNet/IP firmware provides an application interface for both serial and Ethernet devices. You can connect any application, such as a configuration, database, or control application, via the application socket port to the serial and/or Ethernet devices while the device(s) are attached to the PLC via EtherNet/IP.

1.4.5. Filtering and Data Extraction Functionality (Patent Pending)



EtherNet/IP Filtering Functionality (V3.x)

EtherNet/IP firmware provides the following filtering and data extraction functionality:

- Filtering
  - String Filtering of up to 128 bytes of raw/ASCII data to both the PLC and/or application.
  - RFID filtering of EPCglobal formatted RFID tag data to both the PLC and/or application.
  - Barcode filtering of all UPC/EAN formatted barcodes data to both the PLC and/or application.
- Data extraction
  - RFID data extraction extracts all parameters, such as company code, product code, and serial numbers, from any or all of the 43 EPCglobal tag formats. It then transfers the data to the PLC and/or application in a consistent and simple format.
  - Barcode data extraction extracts the company, product, and numbering codes from UPC/EAN formatted barcodes. It then transfers the data to the PLC and/or application in a consistent and simple format.
- Environment specific support
  - Support for multiple RFID reader tag formats.
  - RFID antenna grouping.
  - Aging of filtered string/RFID/barcode entries.
  - Discarding of unrecognized RFID and barcode messages.

For detailed information about filtering and data extraction, see the *DeviceMaster UP Filtering and Data Extraction Reference Guide*.

1.5. EtherNet/IP Application Setup

Before you can configure the EtherNet/IP firmware on the DeviceMaster, you must have previously performed the following steps:

- Install the hardware
- Install PortVision DX
- If necessary, upload the EtherNet/IP firmware using PortVision DX

  Note: *Models that have EtherNet/IP loaded on the* DeviceMaster *are identified in PortVision DX and the* DeviceMaster *is labeled accordingly.*
- Configure the DeviceMaster IP address using PortVision DX

  Note: *If necessary, refer to the EtherNet/IP Hardware Installation and Configuration Guide for the above procedures.*

Use the following steps to complete the DeviceMaster configuration for EtherNet/IP.

1. Select the appropriate programming procedure for the following interfaces:

Interfaces	Programming Procedure
ControlLogix PLC	• Program the PLC. See the instructions in ControlLogix PLC Programming Example Instructions on Page 153. • **(Optional)** Access the DeviceMaster *Serial* | *Port* and *Ethernet* | *Device Configuration* pages to configure the serial/socket port settings, if you did not configure the serial/socket port setting in the PLC program. See the *DeviceMaster Interface Configuration Guide* for procedures and reference Embedded Configuration Pages on Page 85, if necessary.
SLC or MicroLogix PLC	• Program the SLC or MicroLogix PLC, see the instructions in SLC or MicroLogix PLC Programming Example Instructions on Page 166. • Access the DeviceMaster *Serial* | *Port* and *Ethernet* | *Device Configuration* pages to configure the serial/socket port settings. See the *DeviceMaster Interface Configuration Guide* for procedures and reference Embedded Configuration Pages on Page 85, if necessary.
PLC-5 PLC	• Program the PLC-5 PLC, see the instructions in PLC-5 PLC Programming Example Instructions on Page 193. • Access the DeviceMaster *Serial* | *Port* and *Ethernet* | *Device Configuration* pages to configure the serial/socket port settings. See the *DeviceMaster Interface Configuration Guide* for procedures and reference Embedded Configuration Pages on Page 85, if necessary.

2. Connect your serial device or devices and make sure all Ethernet devices are attached to the same Ethernet subnet. If necessary, refer to the *EtherNet/IP Hardware Installation and Configuration Guide*.

Chapter 2. Data Transfer

This chapter discusses data transfer.

2.1. Data Message Format

The following data message format is used for all interfaces. The ControlLogix interface uses SINT, (8 bit bytes), and the MicroLogix/SLC/PLC-5 interface uses 16 bit words for the data arrays. All data is sent to and received from the PLC in little endian format.

2.1.1. Receive Data Message

The *Receive Data* message format contains a sequence number, a length and a data array.

The following table displays the format of the *Receive Data* message.

Name	Data Type	Data Value(s)	Access Rule
Receive (DeviceMaster to PLC) message data			Read-only
Structure of:			
Produced data sequence number	UINT	0-65535 (FFFF hex)	
Data length (in bytes)	UINT	0-(MSG payload-4)	
Data array	Array of SINT	0-255	

Receive messages have the following characteristics:

- The Produced data sequence number is incremented when the data is updated.
- The data length field indicates the number of valid bytes contained in the message.
- The message received from the PLC determines the actual length of the message returned to the PLC. (This is often greater than the length of the actual *Receive Data* message.)
- All unused bytes in a message returned to the PLC are filled with zeros.
- The GW EIP/ASCII supports serial packets of up 1518 bytes and socket packets up to 2048 bytes in the **Write-To-Tag/File transfer to PLC mode**.
- For large received data packets, see sections on Serial and Socket Data Transfer objects or MicroLogix/ SLC/PLC-5 interface.

2.1.2. Transmit Data Message

Just like the *Receive Data* message, the *Transmit Data* message format contains a sequence number, a length and a data array.

The following table displays the format of the *Transmit Data* message.

Name	Data Type	Data Value(s)	Access Rule
Transmit (PLC to DeviceMaster) message data			Read/Write
Structure of:			
Produced data sequence number	UINT	0-65535 (FFFF hex)	
Data length (in bytes)	UINT	0-(MSG payload-4)	
Data array	Array of SINT	0-255	

Transmit messages have the following characteristics:

- For **Write MSG from PLC** mode, all messages received from a PLC will be transmitted, whether or not the produced data sequence number has changed or not.
- For **Class 1 from PLC** mode, messages received from a PLC will only be transmitted if the sequence number has changed.
- If the **TX MSG Sequence Number Checking** option is selected, produced data sequence numbers that are not incremented from the last transmit message will be identified as errors. Statistics will be provided in the diagnostic web pages.
- The data length field indicates the number of valid bytes contained in the message.
- The actual length of a message received from the PLC may contain extra, unused data.
- It ignores all unused bytes in a message.
- A **Get** returns the last successfully transmitted serial/socket packet.

2.2. Data Transfer Methods

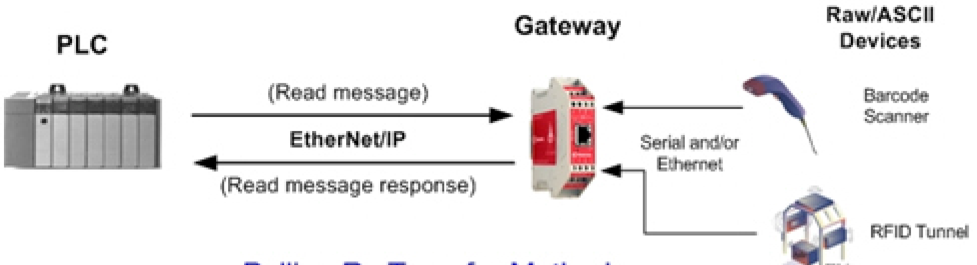
The GW EIP/ASCII gateway provides a selection of data transfer methods and a number of options to customize the data handling for different environments.

2.2.1. Receive Data Methods

The GW EIP/ASCII supports the following receive data methods:

2.2.1.1. Polling-PLC Requests Data

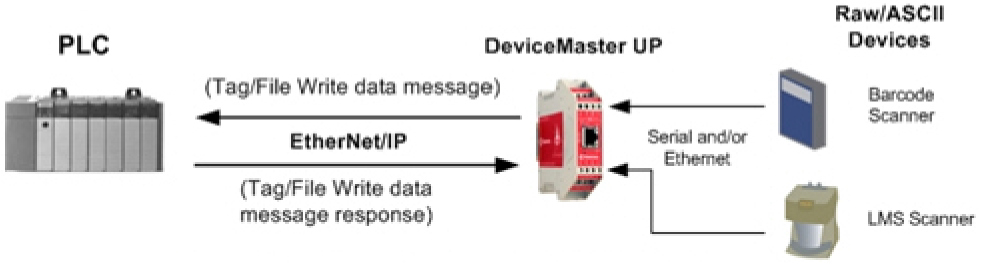
Also called *Slave-Mode* for some industrial protocols, the polling method requires the controller to request received data from the DeviceMaster via messages. The DeviceMaster does not respond until it receives a request for data.



Polling Rx Transfer Method

2.2.1.2. Write-to-Tag/File Gateway Writes Data Directly Into PLC Memory

Also called *Master-Mode* for some industrial protocols, the **Write-to-Tag/File** method requires the DeviceMaster to send messages that write data directly into a tag or file on the PLC. The DeviceMaster sends new data to the PLC immediately.



Write-to-Tag/File Rx Transfer Method

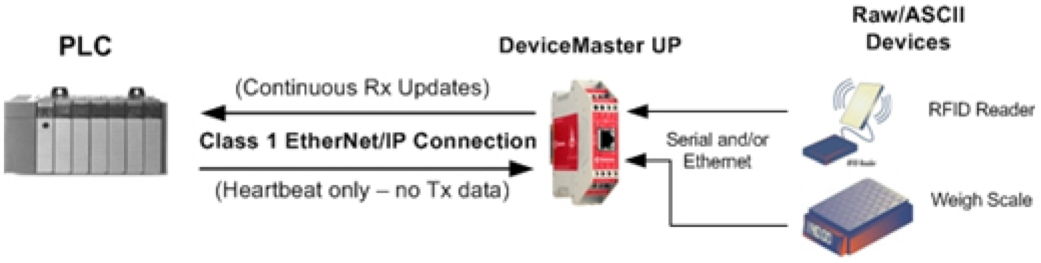
- Serial packets up to 1518 bytes may be received while operating in the Write-To-Tag transfer to plc mode.
- Socket packets up to 2048 bytes may be received while operating in the Write-To-Tag transfer to plc mode.
- ControlLogix family PLCs (ControlLogix/CompactLogix/SoftLogix/FlexLogix, etc):
  - All tags must be single dimension arrays of type SINT. (i.e. Com1_RdData[444], type= SINT)
  - For packets over 440 bytes, the DeviceMaster places the data into a sequence of tags. These tags must meet the following criteria:
    - The entire sequence of tags must be large enough to contain the maximum sized receive packet plus four SINTS for the sequence number and length parameters.
    - All tags except the last of the sequence must be 444 SINTs in size.
    - The tags must have the same base name and numbered in sequence. The first tag is not numbered (i.e. Com1_RxData), the second tag has a 2 appended (i.e. Com1_RxData2), the third has a 3 appended (i.e. Com1_RxData3) and so on.

    - The sequence number and total length is placed in the first tag and the first tag is the last tag updated. Therefore, once the sequence number is updated, the entire serial packet has been received and the PLC can process the data.

- For MicroLogix/SCL/PLC-5 PLCs:
  - All files must be of type integer, (i.e. N10:0, length = 256)
  - For large received data packets:
    - The data is automatically placed in sequential files.
    - The files must be 256 integers in size with the exception of the last file. The last file may be shorter than 256 integers as long as the total length of all files in the sequence is sufficient to hold the largest receive packet, plus two integers for the sequence number and length parameters.
    - All data has been transferred to the PLC when the sequence number is updated.

2.2.1.3. Class 1 Connection (Input Only)-PLC and Gateway Utilize an I/O Connection

Also called *I/O Mode* for some industrial protocols, the Class 1 connection method requires the DeviceMaster and PLC to connect to each via an I/O connection. For EtherNet/IP, a connection over UDP must first be created. Once the connection is established, the gateway sends input data to the PLC at a cyclic rate.



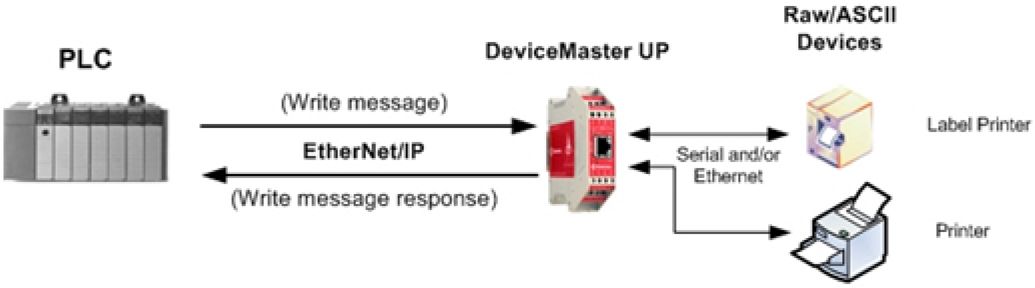
Class 1 Rx Transfer Method

2.3. Transmit Data Methods

The DeviceMaster gateway supports the following transmit data methods.

2.3.1. PLC-Writes

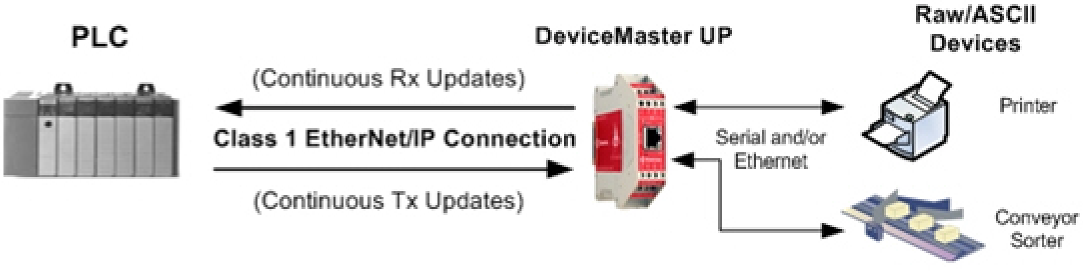
Also called *Slave-Mode* for some industrial protocols, the PLC-Writes method requires the PLC to send data to the DeviceMaster via write messages.



PLC-Writes Tx Transfer Method

2.3.2. Class 1 Connection (Input & Output) - PLC and DeviceMaster Utilize an I/O Connection

Also called *I/O Mode* for some industrial protocols, the Class 1 connection method requires DeviceMaster and PLC to connect via an I/O connection. For EtherNet/IP, a connection over UDP must first be created. Once the connection is established, the PLC and DeviceMaster continually exchange data at a cyclic rate.



Class 1 Rx/Tx Transfer Method

Chapter 3. Programming Interface

3.1. Overview

The DeviceMaster provides an EtherNet/IP interface to:

- Transmit and receive raw/ASCII serial and socket (Ethernet device) data
- Filtering of the data
- Extraction of RFID and barcode parameters
- A connection to an optional application via a TCP/IP socket

The DeviceMaster provides EtherNet/IP connectivity to the entire ControlLogix family of PLCs as well as the SLC, PLC-5, and MicroLogix PLCs.

3.1.1. ControlLogix Family

The DeviceMaster supports ControlLogix PLCs. You can configure DeviceMaster through the PLC program or the DeviceMaster *EtherNet/IP | Device Configuration* web page. The DeviceMaster interface for the ControlLogix PLC is described in *3.2. EtherNet/IP Interface Profile (ControlLogix)* on Page 20.

The *EtherNet/IP Quick Start* describes the quick start for read-only devices such as barcode scanners and read/write devices such as printers.

6.4. ControlLogix PLC Programming Example Instructions on Page 153 provides descriptions of the RSLogix 5000 PLC program examples intended to aid the PLC programmer.

You can also use the *Serial Port and Ethernet Device Configuration* pages on the DeviceMaster when you do not want to use the PLC programming interface to configure the DeviceMaster. *Chapter 4. Embedded Configuration Pages* on Page 85 describes the web pages on the DeviceMaster and provides instructions for configuring the serial/socket port settings.

3.1.2. PLC-5/SLC or MicroLogix

The DeviceMaster supports PLC-5, SLC and MicroLogix PLCs. You must configure the DeviceMaster through the web pages on the DeviceMaster. The DeviceMaster interface for the PLC-5/SLC or MicroLogix PLC is described in *3.3. PLC-5/SLC and MicroLogix Interfaces* on Page 76.

The *EtherNet/IP Quick Start* describes the quick start for read-only devices such as barcode scanners and read/write devices such as printers.

- **SLC or MicroLogix PLC** and want more information than provided in the *EtherNet/IP Quick Start*, see *6.5. SLC or MicroLogix PLC Programming Example Instructions* on Page 166, which contains descriptions of the RSLogix 500 PLC program examples intended to aid the PLC programmer.
- **PLC-5 PLC** and want more information than provided in the *EtherNet/IP Quick Start*, see *6.6. PLC-5 PLC Programming Example Instructions* on Page 193, which contains descriptions of the RSLogix 5 PLC program examples intended to aid the PLC programmer.

3.1.3. What is EtherNet/IP?

EtherNet/IP is an industrial application layer protocol for industrial automation applications. The IP stands for ‘Industrial Protocol’. Built on the standard TCP/IP and UDP/IP protocols, it uses long established Ethernet hardware and software to define an application layer protocol for configuring access and controlling industrial automation devices. The EtherNet/IP application layer protocol is based on the Common Industrial Protocol (CIP) layer. Building on these protocols, EtherNet/IP provides a seamless integrated system from the Industrial floor to the enterprise network.

3.2. EtherNet/IP Interface Profile (ControlLogix)

This section describes the EtherNet/IP objects included in the ControlLogix EtherNet/IP interface and supported by the DeviceMaster.

3.2.1. Serial Port Configuration Object Definition (70 Hex)

The *Serial Port Configuration* vendor specific object defines the protocol by which:

- A PLC can communicate with a serial port device through a DeviceMaster over EtherNet/IP.
- An optional application can communicate with a serial device through the DeviceMaster over an Ethernet TCP/IP socket port.
- The optional serial data filtering and data extraction functions can be implemented.

Note: *The instance number corresponds to the associated serial port number on the DeviceMaster. (Port numbers are numbered from one to N.)*

You can disregard this object definition if you configure the DeviceMaster using the *Serial Port Configuration* web page. See *Chapter 4. Embedded Configuration Pages* on Page 85 to configure the DeviceMaster using the embedded web page.

3.2.1.1. Class Attributes

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Revision	UINT	1	Get
2	Max Instance	UINT	Number of ports on the DeviceMaster	Get
3	Num Instances	UINT	Number of ports on the DeviceMaster	Get

3.2.1.2. Instance Attributes

Note: *Get returns the last command sent.*

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1 (Default=0)	Serial Port Device Type	UDINT	0=Raw Data Device	Set/Get
2 (Default=0)	Serial Port Commands	DWORD	1=Reset Serial Port 2=Save in Flash 4=Clear Sequence Counters 8=Clear Statistics Counters	Set/Get
3 (Default=9600)	Baud Rate	UDINT	Valid rates: 300, 600, 1200, 2400, 4800, 9600 (default), 19200, 38400, 57600, 115200, 230400	Set/Get

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
4 (Default=0 RS-232)	Interface Mode	USINT	0=RS-232 (Default) 1=RS-422 2=RS-485	Set/Get
5 (Default=0 None)	Parity	USINT	0=None (Default) 1=Even 2=Odd	Set/Get
6 (Default=8)	Data Bits	USINT	Valid Values: 5-8	Set/Get
7 (Default=1)	Stop Bits	USINT	Valid Values: 1 or 2	Set/Get
8 (Default=0 None)	Flow Control	USINT	0=None (Default) 1=RTS/CTS 2=XON/XOFF 3=Half Duplex	Set/Get
9 (Default=0 Off)	DTR Control	USINT	0=Off (Default) 1=On	Set/Get
10 (Default=0 No STX)	PLC Transmit STX Append Value **Structure of**:			Set/Get
	Length	USINT	0,1,2 (0=No STX)	
	Value1 (Only valid if Length not zero)	USINT	0 to 255	
	Value2 (Only valid if Length=2)	USINT	0 to 255	
11 (Default=0)	PLC Transmit ETX Append Value **Structure of**:			Set/Get
	Length	USINT	0,1,2 (0=No ETX) (Default=0)	
	Value1 (Only valid if Length not zero)	USINT	0 to 255	
	Value2 (Only valid if Length=2)	USINT	0 to 255	
12	Reserved	UINT	0	Get
13 (Default=1)	Receive STX Detect Value **Structure of:**			Set/Get
	Length	USINT	0,1,2 (0=No STX) (Default=1)	
	Value1 (Only valid if Length not zero)	USINT	0 to 255	
	Value2 (Only valid if Length=2)	USINT	0 to 255 (Default: Value1 = 2)	

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
14 (Default=Value1 Value1=3)	Receive ETX Detect Value **Structure of:** Length Value1 (Only valid if Length not zero) Value2 (Only valid if Length=2)	 USINT USINT USINT	 0,1,2 (0=No ETX) (Default=1) 0 to 255 0 to 255 (Default: Value1 = 3)	Set/Get
15 (Default=200)	Receive Timeout Between Packets (if no ETX or time to wait for ETX value)	UINT (msec)	0 to 65535 (Default = 200 msec)	Set/Get
16 (Default=03)	Serial Port Transfer Options	WORD (bitwise OR)	01 Hex = Strip received STX/ETX characters to the PLC 02 Hex = Discard received packets with errors 04 Hex = (PLC-5/SLC) Rx MS Byte First 08 Hex = (PLC-5/SLC) Tx MS Byte First 10 Hex = Tx Sequence Number Checking 20 Hex = Disable Queuing of Non-Filtered Rx Messages 40 Hex = Strip received STX/ETX characters to the application (Default = 03) 80 Hex = Drop oversized received packets	Set/Get
17 (Default=4 Class 1)	Receive (DeviceMaster to PLC) Data Transfer Method	USINT	0=OFF 1=Unsolicited - Write-to-Tag 2=Unsolicited - Write-to-Tag-Synced 3=Polling 4=Class1 (Default=4)	Set/Get
18 (Default = 1 Class 1)	Transmit (PLC to DeviceMaster) Data transfer Method	USINT	0=Write-Message 1=Class1 (Default = 1)	Set/Get
19	Maximum Receive Data Packet Size	UINT	1-1518 (Default is dependent on the number of serial ports)	Set/Get
20 (Default=40)	Maximum Rx PLC Update Rate (No more than one message per time interval in ms.)	UINT (ms)	10-65535 (Default = 40)	Set/Get
21	Reserved	USINT	0	Get
22 (Default=0)	PLC Controller Slot Number	USINT	0 to Max Slot Number on PLC (Default=0)	Set/Get
23	PLC IP Address	UDINT	00000000 Hex to FFFFFFFF Hex (Mask= 255.255.255.255)	Set/Get
24	Receive (DeviceMaster to PLC) Produced Data Tag Name	STRING (Array of 40 SINTs)	ASCII string	Set/Get

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
25	Application Socket Enable	USINT	0=Disabled 1=Enabled	Set/Get
26	Application Listen Enable	USINT	0=Disabled 1=Enabled	Set/Get
27	Application Connect Mode	USINT	0=Never 1=Connect Always 2=Connect On Data	Set/Get
28	Application Disconnect Mode	USINT	0=Never 1=Disconnect On Idle	Set/Get
29	Application Listen Socket Port	UINT	0-65535	Set/Get
30	Application Connect Socket Port	UINT	0-65535	Set/Get
31	Application Connect IP Address	UDINT	00000000 Hex to FFFFFFFF Hex (Mask=255.255.255.255)	Set/Get
32	Application Idle Timeout	UDINT (ms)	0 to FFFFFFFF Hex	Set/Get
33	To PLC Filter Mode	USINT	0=Off 1=String (128 byte maximum) 2=RFID (EPCglobal formats) 3= Barcode (UPC/EAN formats)	Set/Get
34	To Application Filter Mode	USINT	0=Off 1=String (128 byte maximum) 2=RFID (EPCglobal formats) 3= Barcode (UPC/EAN formats)	Set/Get
35	Discard Unrecognized Data Mode (RFID and Barcode Filter mode only)	USINT	0=Off 1=To PLC 2=To application 3=To PLC/application	Set/Get
36	RFID Antenna Grouping	USINT	0=None 1=Groups of Twos 2=Groups of Threes 3=Groups of Fours 4=First Two Only 5=First Three Only	Set/Get
37	To PLC Filter Options	WORD (Bitwise OR)	01 Hex = Encoding Scheme 02 Hex = Filter Code 04 Hex = Antenna Number 08 Hex = Company Code 10 Hex = Product/Location Code 20 Hex = Serial Number	Set/Get

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
38	To Application Filter Options	WORD (Bitwise OR)	01 Hex = Encoding Scheme 02 Hex = Filter Code 04 Hex = Antenna Number 08 Hex = Company Code 10 Hex = Product/Location Code 20 Hex = Serial Number	Set/Get
39	Filter Age Time	UDINT (ms)	0 - FFFFFFFF Hex	Set/Get
40	RFID Reader Interface Type	UINT	0=Unspecified 10=Alien (Text Mode) 11=Alien (Terse Mode) 20=Intermec (Hex ASCII Mode)	Set/Get
41	Barcode Formats (Barcode Filtering Only)	UINT	**Standard 12-14 Digit Format** (Mask = 000F Hex) 00 Hex=NONE 01 Hex=Five Company/Five Product Digits 02 Hex=Six Company/Four Product Digits 03 Hex=Seven Company/Three Product Digits 04 Hex=Eight Company/Two Product Digits 05 Hex=Nine Company/One Product Digits **Eight Digit Format** (Mask = 00F0 Hex) 00 Hex=NONE 10 Hex=EAN-8; Two Company/Five Product Digits 20 Hex=EAN-8; Three Company/Four Product Digits 30 Hex=UPC-E	Set/Get
42 (Default=0)	Application Transmit STX Append Value **Structure of:**			Set/Get
	Length	USINT	0,1,2 (0=No STX) (Default=0)	
	Value1 (Only valid if Length not zero)	USINT	0 to 255	
	Value2 (Only valid if Length=2)	USINT	0 to 255	

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
43 (Default=0)	Application Transmit ETX Append Value			Set/Get
	Structure of:			
	Length	USINT	0,1,2 (0=No ETX) (Default=0)	
	Value1 (Only valid if Length not zero)	USINT	0 to 255	
	Value2 (Only valid if Length=2)	USINT	0 to 255	
44	Maximum Transmit Data Packet Size	UINT	1-1518 (Default =dependent on the number of serial ports) ***Note:** Applicable only to Class1 Transmit Transfer mode.*	Set/Get

3.2.1.3. Common Services

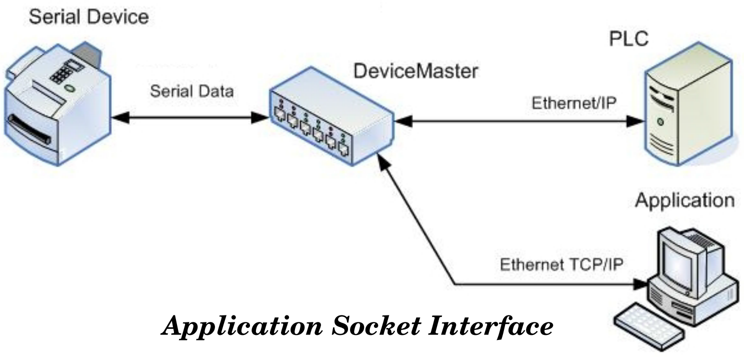
Service Code	Implemented in Class	Implemented in Instance	Service Name
01 Hex	Yes	Yes	Get_Attributes_All
02 Hex	No	Yes	Set_Attributes_All
0E Hex	Yes	Yes	Get_Attribute_Single
10 Hex	No	Yes	Set_Attribute_Single

3.2.1.4. Instance Attribute Definitions

Attribute	Description
Attribute 1 Serial Port Device	This attribute indicates the **Serial Port Device Type**. Raw Data device is the only currently supported option.
Attribute 2 Serial Port Commands	The DeviceMaster supports the following commands: • **Reset serial port** - This option resets the serial port hardware and statistics counters. You must reset the DeviceMaster after modifying any of the serial port configuration options, including: baud rate, interface mode, parity, data bits, stop bits, flow control, or DTR control. It does not clear the sequence counters. • **Save in Flash** - This option saves the port configuration in flash memory. These settings are restored when you reboot the DeviceMaster. • **Clear sequence counters** - This option clears the Receive Produced and Consumed Sequence counters for the selected port. • **Clear statistics counters** - This option clears the statistics counters for the selected port.
Attributes 3 to 9 Standard Serial Port Settings	These are standard serial port settings. Attribute 10 - PLC Transmit STX Append Value - You can set this attribute to append an STX (start of transmission) byte sequence which is configurable as 1 or 2-bytes to the beginning of the serial packet before it is sent. The length indicates the number of STX bytes. The valid values for length are: • **0 (zero)** - Setting this attribute to zero disables this function. • **1 (one STX byte)** - Inserts one STX byte before the data. • **2 (two STX bytes)** - Inserts two STX bytes before the data. • **Value1** - Specifies the transmit character associated with the first STX byte. (Only if the length is not zero.) You can specify a value between 0 and 255. • **Value2** - Specifies the transmit character associated with the second STX byte. (Only if length is two bytes.) You can specify a value between 0 and 255.

Attribute	Description
Attribute 10 PLC Transmit STX Append Value	You can set this attribute to append an STX (start of transmission) byte sequence which is configurable as 1 or 2-bytes to the beginning of the serial packet before it is sent. The length indicates the number of STX bytes. The valid values for length are: • 0 (zero) - Setting this attribute to zero disables this function. • 1 (one STX byte) - Inserts one STX byte before the data. • 2 (two STX bytes) - Inserts two STX bytes before the data. • Value1 - Specifies the transmit character associated with the first STX byte. (Only if the length is not zero.) You can specify a value between 0 and 255. • Value2 - Specifies the transmit character associated with the second STX byte. (Only if length is two bytes.) You can specify a value between 0 and 255.
Attribute 11 PLC Transmit ETX Append Value	You can set this attribute to append an ETX (end of transmission) byte sequence which is configurable as 1 or 2 -bytes to the end of the serial packet before it is sent. The length indicates the number of ETX bytes. The valid values for length are: • **0 (zero)** - Setting this attribute to zero disables this function. • **1 (one ETX byte)** - Inserts one ETX byte at the end of the data. • **2 (two ETX bytes)** - Inserts two ETX bytes at the end of the data. • **Value** - Specifies the transmit character associated with the first ETX byte. (Only if the length is not zero.) You can specify a value between 0 and 255. • **Value2** - Specifies the transmit character associated with the second ETX byte. (Only if length is 2-bytes.) You can specify a value between 0 and 255.
Attribute 13 Receive STX Detect Value	This attribute detects an STX (start of transmission) byte sequence which is configurable as 1 or 2-bytes when it receives a serial packet. The length indicates the number of STX bytes. The valid values for length are: • **0 (zero)** - Setting this attribute to zero disables this function. When disabled, the DeviceMaster accepts the first byte received after the last ETX byte(s) as the start of the next data packet. • **1 (one STX byte)** - Scans serial data for one STX byte. If the DeviceMaster finds an STX byte it collects the data. If the first byte is not the STX byte, the DeviceMaster discards the byte. The DeviceMaster continues to discard the bytes until the DeviceMaster finds an STX byte. • **2 (two STX bytes)** - Scans serial data for two STX bytes. If the DeviceMaster finds two STX bytes it collects the data. If the first two bytes are not the STX bytes, the DeviceMaster discards the bytes. Bytes continue to be discarded until the DeviceMaster finds two STX bytes. • **Value1** - Specifies the character that represents the first STX byte. DeviceMaster looks for this character in the first STX byte. (Only if the length is not zero.) You can specify a value between 0 and 255. • **Value2** - Specifies the character that represents the second STX byte. DeviceMaster looks for this character in the first STX byte. (Only if length is two bytes.) You can specify a value between 0 and 255.

Attribute	Description
Attribute 14 Receive ETX Detect Value	This attribute detects an ETX (end of transmission) byte sequence which is configurable as 1 or 2 bytes marking the end of the serial packet. The length indicates the number of ETX bytes. The valid values for length are: • **0 (zero) -** Setting this attribute to zero disables this function. When disabled, the DeviceMaster uses the Receive Timeout Between Packets (attribute 15) to indicate the end of data packet. • **1 (one ETX byte) -** Scans serial data for one ETX byte. When the DeviceMaster finds an ETX byte it identifies the data as a serial packet. • **2 (two ETX bytes) -** Scans serial data for two ETX bytes. When the DeviceMaster finds two ETX bytes it identifies the data as a serial packet. • **Value1** - Specifies the character to scan for in the first ETX byte. (Only if the length is not zero.) You can specify a value between 0 and 255. • **Value2** - Specifies the character to scan for in the second ETX byte. (Only if length is 2-bytes.) You can specify a value between 0 and 255.
Attribute 15 Receive Timeout Between Packets	This attributes specifies: • How long DeviceMaster waits (in milliseconds) if the Receive ETX length is not zero (0) and it does not receive an ETX byte sequence. • The time to wait (in milliseconds) between serial packets if the Receive ETX Detect length is set to zero (0).
Attribute 16 Serial Port Transfer Options	This attribute specifies special serial port transfer options. The following options are supported: • **01 Hexadecimal (Bit 0)** - DeviceMaster strips received STX/ETX characters from received packets before sending the packet to the PLC. • **02 Hexadecimal (Bit 1)** - DeviceMaster discards received packets with errors. • **04 Hexadecimal (Bit 2)** - (PLC-5/SLC only) DeviceMaster receives Most Significant (MS) byte of 16-bit integer first. The default is transmit Least Significant (LS) byte first. **08 Hexadecimal (Bit 3)** - (PLC-5/SLC only) DeviceMaster transmit Most significant (MS) byte of 16-bit integer first. The default is transmit Least Significant (LS) byte first. • **10 Hexadecimal (Bit 4)** - Enable Transmit Sequence Number Checking. - DeviceMaster rejects messages with duplicate sequence numbers (that is, the same sequence number as the previous transmit data message) and increments the **Duplicate Transmit Sequence Error Count**. - DeviceMaster transmits messages with unexpected transmit sequence numbers (that is, sequence numbers that are not the same as or are not equal to the previous sequence number plus one) and increments the **Unexpected Transmit Sequence Error Count**. • **20 Hexadecimal (Bit 5) -** Disable Queuing of Non-Filtered Rx messages to PLC. If filtering is disabled, only the last message received is sent to the PLC. • **40 Hexadecimal (Bit 6) -** DeviceMaster strips received STX/ETX characters from received packets before sending the packet to the application. • **80 Hexadecimal (Bit 7) -** Drop oversized received data packets.
Attribute 17 Receive (DeviceMaster to PLC) Data Transfer Method	This attribute specifies the Receive Ethernet data transfer method used by the DeviceMaster. There are four methods that DeviceMaster can use to transfer data received from a serial device to the PLC. Refer to *Chapter 2. Data Transfer* on Page 13 for a discussion on transfer modes.

Attribute	Description
Attribute 18 Transmit (PLC to DeviceMaster) Data Transfer Method	This attribute specifies the Transmit Ethernet data transfer method used by the DeviceMaster. There are two methods that DeviceMaster can use to transfer data from the PLC to a serial device. Refer to *Chapter 2. Data Transfer* on Page 13 for a discussion on transfer modes.
Attribute 19 Maximum Receive Data Packet Size	Specifies the maximum acceptable size of a received serial packet. The maximum received serial packet size is 1518 bytes while operating in Class 1, *Write-to-Tag/File* or *Write-to-Tag-File-Synced receive* modes.
Attribute 20 Maximum PLC Update Rate	The maximum rate (or minimum time interval) in milliseconds, that messages are sent to the PLC tag in the *Write-To-Tag receive* method. This attribute configures the DeviceMaster to space the messages to the PLC to prevent overrunning of data before the PLC can process it.
Attribute 22 PLC Controller Slot Number	This attribute specifies the slot number on the PLC where the controller resides. The slot numbers typically start at zero (0) for the first slot. ***Note:*** *The Polling and Class 1 methods do not use this attribute.*
Attribute 23 PLC IP Address	This attribute specifies the IP address in hexadecimal format for the PLC EtherNet/IP card. For example, an IP address of 10.1.2.100 is 0A010264 in hexadecimal. ***Note:*** *The Polling and Class 1 methods do not use this attribute.*
Attribute 24 Receive (DeviceMaster to PLC) Produced Data Tag Name	This attributes specifies the PLC tag name. It indicates where to write received data while operating in the *Unsolicited - Write-to-Tag* or *Unsolicited - Write-to-Tag-Synced receive* method. The maximum length for the tag name is 40 characters.
Attribute 25 Application Socket Enable	This setting enables/disables the Application Socket Interface. Enabling this function allows an application to be connected to the serial port. If both the PLC and application are connected to the serial port, both can transmit to and receive data from the serial port. However, the PLC and application cannot communicate directly to each other.  Serial Device Serial Data DeviceMaster PLC Ethernet/IP Application Ethernet TCP/IP ***Application Socket Interface***
Attribute 26 Application Listen Enable	Enabling this setting allows the application to connect to the DeviceMaster via an Ethernet TCP/IP socket. • 0 = Disables listening - The DeviceMaster does not accept connection attempts. • 1 = Enables listening - The DeviceMaster accepts connection attempts from the application socket port.
Attribute 27 Application Connect Mode	This setting controls if and how the DeviceMaster attempts to connect to the application at the application connection IP address and application connection socket port. • 0 = Never - The DeviceMaster does not attempt to connect to the application. • 1 = Connect Always - The DeviceMaster attempts connecting to the application socket port until a connection is made. • 2 = Connect On Data - The DeviceMaster does not attempt to connect to the application socket port until there is data to send to the application. Once data is received from the serial device, the DeviceMaster attempts connecting to the application.

Attribute	Description
Attribute 28 Application Disconnect Mode	This setting controls if and how the DeviceMaster disconnects from an application. • 0 = Never - The DeviceMaster does not disconnect from the application socket port. • 1 = Disconnect On Idle - The DeviceMaster disconnects when there has been no transmit or received data between the serial device and application socket port for a specified **Idle** period (Attribute 32: Application Connection IP Address).
Attribute 29 Application Listen Socket Port	This is the socket port number on the DeviceMaster the application connects to if **Application Listen Enable** is enabled.
Attribute 30 Application Connection Socket Port	This is the application socket port number the DeviceMaster connects to if the **Application Connect Mode** is set to either **Connect Always** or **Connect On Data**.
Attribute 31 Application Connection IP Address	This is the application IP address in hexadecimal format that the DeviceMaster connects to if the **Application Connect Mode** is set to either **Connect Always** or **Connect On Data**. For example, an IP address of 10.1.2.100 is 0A010264 in hexadecimal.
Attribute 32 Application Idle Timeout	The idle timeout period in milliseconds that is used if the **Application Disconnect Mode** is set to **Disconnect On Idle**.
Attribute 33 To PLC Filter/Data Extraction Mode	The *filter/data extraction* mode to be employed on data to be sent to the PLC. • 0 = Off • 1 = String (128 char max) - Raw/ASCII data is filtered up to 128 characters (or bytes) in length. • 2 = RFID (EPCglobal formats) - RFID data in any of the EPCglobal formats is filtered, the associated parameters is extracted, and the extracted data and RFID tag is sent to the PLC in a specified format. • 3 = Barcode (UPC/EAN formats) - Barcode data in specified UPC/EAN formats is filtered, the associated parameters is extracted, and the extracted data and barcode is sent to the PLC in a specified format. See the barcode format definitions in Attribute 41 on Page 32. See the *DeviceMaster UP Filtering and Data Extraction Reference Guide* (Page 8) for further details.

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td><b>Attribute 34</b>
To Application Filter/ Data Extraction Mode</td><td>The <i>filter/data extraction</i> mode to be employed on data to be sent to the application.
• 0 = Off
• 1 = String (128 char max) - Raw/ASCII data is filtered up to 128 characters (or bytes) in length.
• 2 = RFID (EPCglobal formats) - RFID data in any of the EPCglobal formats is filtered, the associated parameters is extracted, and the extracted data and RFID tag is sent to the application in a specified format.
• 3 = Barcode (UPC/EAN formats) - Barcode data in specified UPC/EAN formats is filtered, the associated parameters is extracted, and the extracted data and barcode is sent to the application in a specified format. See the barcode format definitions in Attribute 41 on Page 32.
The <i>application filter</i> mode can be set independently of the <i>PLC filtering</i> mode. The only exceptions are:
• If the PLC filter mode is set to <b>RFID</b>, the <i>application filter</i> mode cannot be set to <b>Barcode</b>.
• If the PLC filter mode is set to <b>Barcode</b>, the <i>application filter</i> mode cannot be set to <b>RFID</b>.
See the <i>DeviceMaster UP Filtering and Data Extraction Reference Guide</i> (Page 8) for further details.</td></tr>
<tr><td><b>Attribute 35</b>
Discard Unrecognized Data Mode</td><td>This attribute controls what to do with unrecognized RFID or barcode data.
• 0 = Off - Send unrecognized data to the PLC and/or application.
• 1 = Discard unrecognized data to the PLC. Allow sending of unrecognized data to the application.
• 2 = Discard unrecognized data to the application. Allow sending of unrecognized data to the PLC.
• 3 = Discard unrecognized data to both the PLC and application.</td></tr>
<tr><td><b>Attribute 36</b>
RFID Antenna Grouping</td><td>This attribute is applicable only to RFID filtering and only if the <b>Antenna</b> filtering option is enabled. It allows the DeviceMaster to filter RFID tags based on Antenna groupings. The possible groupings are:
<table>
<tr><th>Setting</th><th>Group 1 Antennas</th><th>Group 2 Antennas</th><th>Group 3 Antennas</th><th>Group N Antennas</th></tr>
<tr><td>None</td><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td>Groups of Twos</td><td>1,2</td><td>3,4</td><td>5,6</td><td>Etc.</td></tr>
<tr><td>Groups of Threes</td><td>1,2,3</td><td>4,5,6</td><td>7,8,9</td><td>Etc.</td></tr>
<tr><td>Groups of Fours</td><td>1,2,3,4</td><td>5,6,7,8</td><td>9,10,11,12</td><td>Etc.</td></tr>
<tr><td>First Two Only</td><td>1,2</td><td>3</td><td>4</td><td>N+1</td></tr>
<tr><td>First Three Only</td><td>1,2,3</td><td>4</td><td>5</td><td>N+2</td></tr>
</table>
</td></tr>
</table>

Attribute	Description
Attribute 37 To PLC Filtering Options	This attribute defines the RFID filtering criteria to the PLC. If an option is enabled, it is used to decide when an RFID tag can be filtered or sent to the PLC. • 01 Hex = **Encoding/Numbering -** Include the **Encoding/Numbering** code in the filtering criteria, which is part of the RFID tag or barcode data. • 02 Hex = **Filter Value -** Include the **Filter Value** in the filtering criteria, which is part of the RFID tag data. • 04 Hex = **Antenna -** Include the **Antenna** number in the filtering criteria. This is data from the RFID reader and not from the RFID tag. • 08 Hex = **Company -** Include the **Company** code in the filtering criteria, which is part of the RFID tag or barcode data. • 10 Hex = **Product/Location -** Include the **Product/Location** code in the filtering criteria, which is part of the RFID tag or barcode data. • 20 Hex = **Serial Number -** Include the **Serial Number** in the filtering criteria, which is part of the RFID tag data. See the *DeviceMaster UP Filtering and Data Extraction Reference Guide* (Page 8) for further details.
Attribute 38 To Application Filtering Options	This attribute defines the RFID filtering criteria to the application. If an option is enabled, it is used to decide when an RFID tag can be filtered or sent to the application. • 01 Hex = **Encoding/Numbering -** Include the **Encoding/Numbering** code in the filtering criteria, which is part of the RFID tag or barcode data. • 02 Hex = **Filter Value -** Include the **Filter Value** in the filtering criteria, which is part of the RFID tag data. • 04 Hex = **Antenna -** Include the **Antenna** number in the filtering criteria. This is data from the RFID reader and not from the RFID tag. • 08 Hex = **Company -** Include the **Company** code in the filtering criteria, which is part of the RFID tag or barcode data. • 10 Hex = **Product/Location -** Include the **Product/Location** code in the filtering criteria, which is part of the RFID tag or barcode data. • 20 Hex = **Serial Number -** Include the **Serial Number** in the filtering criteria, which is part of the RFID tag data. See the *DeviceMaster UP Filtering and Data Extraction Reference Guide* (Page 8) for further details.
Attribute 39 Filter Age Time	This attribute defines the time a filter string, RFID tag, or barcode continues to be filtered after the last time it was received. If an entry is received before the **Filter Age Time** has passed, the entry is filtered and the data is not be sent to the PLC and/or application. However, if the **Filter Age Time** has passed, it passes filtering and be sent to the PLC and/or application.
Attribute 40 RFID Reader Interface Type	This attribute defines the expected RFID data format. Each format is unique and pertains to the RFID reader manufacturer. If a RFID reader is to be used and it provides a similar format to the ones listed below, it can also be used. • 0=Unspecified • 10 (Decimal) = Alien (Text Mode) • 11 (Decimal) = Alien (Terse Mode) • 20 (Decimal) = Intermec (Hex ASCII Mode) See the *DeviceMaster UP Filtering and Data Extraction Reference Guide* for details.

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td><b>Attribute 41</b>
Barcode Formats</td><td>
<p>This attribute defines barcode format to be used for both standard and eight digit UPC labels. The term standard refers to UPC-A, EAN-13, JAN, and EAN-14 barcodes which all have ten company/product digits.</p>
<p>The standard and eight digit formats are selected independently and each operates independently. Barcode filtering/data extraction does not function if no format is selected.</p>
<table>
<tr><th><u>Format</u></th><th>Numbering <u>Digits</u></th><th>Company <u>Digits</u></th><th>Product <u>Digits</u></th><th>Check <u>Digit</u></th></tr>
<tr><td><b><i>Standard Formats</i></b></td><td></td><td></td><td></td><td></td></tr>
<tr><td>None</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr>
<tr><td>Company-5/ Product-5</td><td>1-3</td><td>5</td><td>5</td><td>1</td></tr>
<tr><td>Company-6/ Product-4</td><td>1-3</td><td>6</td><td>4</td><td>1</td></tr>
<tr><td>Company-7/ Product-3</td><td>1-3</td><td>7</td><td>3</td><td>1</td></tr>
<tr><td>Company-8/ Product-2</td><td>1-3</td><td>8</td><td>2</td><td>1</td></tr>
<tr><td>Company-9/ Product-1</td><td>1-3</td><td>9</td><td>1</td><td>1</td></tr>
<tr><td><b><i>Eight Digit Formats</i></b></td><td></td><td></td><td></td><td></td></tr>
<tr><td>EAN-8 Number-2/Product 5</td><td>2</td><td>0</td><td>5</td><td>1</td></tr>
<tr><td>EAN-8 Number-3/Product 4</td><td>3</td><td>0</td><td>4</td><td>1</td></tr>
<tr><td>UPC-E</td><td></td><td></td><td></td><td></td></tr>
</table>
<p>See the <i><u>DeviceMaster UP Filtering and Data Extraction Reference Guide</u></i> for details.</p>
</td></tr>
<tr><td><b>Attribute 42</b>
Application Transmit STX Append Value</td><td>
<p>You can set this attribute to append an STX (start of transmission) byte sequence which is configurable as 1 or 2-bytes to the beginning of the serial packet before it is sent.</p>
<p>The length indicates the number of STX bytes. The valid values for length are:</p>
<ul>
<li><b>0 (zero)</b> - Setting this attribute to zero disables this function.</li>
<li><b>1 (one STX byte)</b> - Inserts one STX byte before the data.</li>
<li><b>2 (two STX bytes)</b> - Inserts two STX bytes before the data.</li>
<li><b>Value1</b> - Specifies the transmit character associated with the first STX byte. (Only if the length is not zero.) You can specify a value between 0 and 255.</li>
<li><b>Value2</b> - Specifies the transmit character associated with the second STX byte. (Only if length is two bytes.) You can specify a value between 0 and 255.</li>
</ul>
</td></tr>
<tr><td><b>Attribute 43</b>
Application Transmit ETX Append Value</td><td>
<p>You can set this attribute to append an ETX (end of transmission) byte sequence which is configurable as 1 or 2 -bytes to the end of the serial packet before it is sent.</p>
<p>The length indicates the number of ETX bytes. The valid values for length are:</p>
<ul>
<li><b>0 (zero)</b> - Setting this attribute to zero disables this function.</li>
<li><b>1 (one ETX byte)</b> - Inserts one ETX byte at the end of the data.</li>
<li><b>2 (two ETX bytes)</b> - Inserts two ETX bytes at the end of the data.</li>
<li><b>Value</b> - Specifies the transmit character associated with the first ETX byte. (Only if the length is not zero.) You can specify a value between 0 and 255.</li>
<li><b>Value2</b> - Specifies the transmit character associated with the second ETX byte. (Only if length is 2-bytes.) You can specify a value between 0 and 255.</li>
</ul>
</td></tr>
<tr><td><b>Attribute 44</b>
Maximum Transmit Data Packet Size</td><td>
<p>Specifies the maximum acceptable size of transmit serial packet. The maximum transmit serial packet size is 1518 bytes while operating in the <b>Class1 transmit</b> mode. This attribute is not used in the <b>Write-Msg transmit</b> mode.</p>
</td></tr>
</table>

3.2.2. Serial Port Data Transfer Object Definition (71 Hex)

The *Serial Port Data Transfer vendor specific* object defines the attributes by which the PLC can transfer data to and from a serial port device through a DeviceMaster over EtherNet/IP.

Note: *There is one instance of this object per serial port. The instance number corresponds to the associated serial port number on the DeviceMaster. (Port numbers are numbered from one to N.)*

3.2.2.1. Class Attributes

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Revision	UINT	1	Get
2	Max Instance	UINT	Number of ports on the DeviceMaster	Get
3	Num Instances	UINT	Number of ports on the DeviceMaster	Get

3.2.2.2. Instance Attributes

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Transmit (PLC to DeviceMaster) message data.			Set/Get
	Structure of:			
	Produced data sequence number	UINT	0-65535 (FFFF hex)	
	Data length	UINT	1-440**	
	Data array	Array of USINT	0-255	
2	Receive (DeviceMaster to PLC) message data			Get
	Structure of:			
	Produced data sequence number	UINT	0-65535 (FFFF hex)	
	Data length	UINT	0-440*	
	Data array	Array of USINT	0-255	
3	Receive (DeviceMaster to PLC) produced data sequence number			
	Normally sent to the PLC from the DeviceMaster in data transfer.	UINT	0-65535 (FFFF hex)	Set/Get
4	Receive (DeviceMaster to PLC) consumed sequence number			
	Normally updated by the PLC in the *Write-to-Tag-Synced receive* method.	UINT	0-65535 (FFFF hex)	Set/Get
5	Transmit (PLC to DeviceMaster) produced data sequence number			
	Normally sent to the DeviceMaster from the PLC in data transfer.	UINT	0-65535 (FFFF hex)	Set/Get

* *For the Polling receive method only. The maximum sized serial port message in the Class 1 Write-to-Tag and Write-to-Tag-Synced receive method is 1518 bytes.*

** *For Write-Msg Transmit method only. The maximum sized serial port message in Class 1 is 1518 bytes.*

3.2.2.3. Common Services

Service Code	Implemented in Class	Implemented in Instance	Service Name
0E Hex	Yes	Yes	Get_Attribute_Single
10 Hex	No	Yes	Set_Attribute_Single

3.2.2.4. Instance Attribute Definitions

Attribute	Description
Attribute 1 Transmit (PLC to DeviceMaster) Message Data	This attribute transmits data out of a serial port on the DeviceMaster. • In most cases, incrementing the sequence number is optional. However, it is required if you enable the Transmit Sequence Number Checking option. (For more information, see Attribute 16 - Serial Port Transfer Options on Page 27.) • The length must be at least one and a maximum of 440-bytes. • A **Get** returns the last successfully transmitted data message.
Attribute 2 Receive (DeviceMaster to PLC) Message Data	This attribute provides the receive data while operating in the *Polling communication* method. • DeviceMaster increments the sequence number for each new serial port packet received. • A length of zero indicates no data was received on the specified serial port. • Two or more **Gets** may return the same data packet, but the messages will also have the same sequence number.
Attribute 3 Receive (DeviceMaster to PLC) Produced Data Sequence Number	Use this attribute to get and set the *Produced Data Sequence number*. This is the same *Produced Data Sequence number* sent to the PLC in all the *receive communication* methods.
Attribute 4 Receive (DeviceMaster to PLC) Consumed Data Sequence Number	Use this attribute to get and set the *Consumed Data Sequence number*. You can only specify this attribute when you are using the *Unsolicited - Write-to-Tag-Synced receive* method under Attribute 17 in the *Serial Port Configuration* object definition. When used, the PLC increments this attribute to indicate the data received has been consumed and it is now ready for another serial data packet. For more information, see the description of *Unsolicited -Write-to-Tag-Synced* in Attribute 17 on Page 27.
Attribute 5 Transmit (PLC to DeviceMaster) Produced Data Sequence Number	Use this attribute to get and set the *Transmit Produced Data Sequence number*. This is the same *Produced Data Sequence number* sent to the DeviceMaster in the *Transmit Message data*.

3.2.3. Serial Port Statistics Object Definition (72 Hex)

The *Serial Port Statistics* object defines the statistics gathered by the DeviceMaster on a serial port basis.

Note: *There is one instance of this object per serial port. The instance number corresponds to the associated serial port number on the DeviceMaster. (Port numbers are numbered from one to N.)*

3.2.3.1. Class Attributes

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Revision	UINT	1	Get
2	Max Instance	UINT	Number of ports on the DeviceMaster	Get
3	Num Instances	UINT	Number of ports on the DeviceMaster	Get

3.2.3.2. Instance Attributes

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Receive Byte Count	UDINT	0=default	Get
2	Receive Packet Count	UDINT	0=default	Get
3	Transmit Byte Count	UDINT	0=default	Get
4	Transmit Packet Count	UDINT	0=default	Get
5	Dropped Packet to PLC Count	UDINT	0=default	Get
6	Parity Error Count	UDINT	0=default	Get
7	Framing Error Count	UDINT	0=default	Get
8	Overrun Error Count	UDINT	0=default	Get
9	Received Consumed Sequence Error Count	UDINT	0=default	Get
10	Duplicate Transmit Sequence Number errors	UDINT	0=default	Get
11	Unexpected Transmit Sequence Number errors	UDINT	0=default	Get
12	Dropped Packet to Application Count	UDINT	0=default	Get

3.2.3.3. Common Services

Service Code	Implemented in Class	Implemented in Instance	Service Name
01 Hex	Yes	Yes	Get_Attribute_All
0E Hex	Yes	Yes	Get_Attribute_Single

3.2.3.4. Instance Attribute Definitions

Attribute	Description
Attribute 1 Receive Byte Count	This attribute counts the number of bytes received over the serial port.
Attribute 2	Receive Packet Count: This attribute counts the packets received over the serial port.
Attribute 3 Transmit Byte Count	This attribute counts the number of bytes sent out the serial port.
Attribute 4 Transmit Packet Count	This attribute counts the number of packets sent out the serial port.
Attribute 5 Dropped Packet to PLC Count	This attribute counts the number of received serial packets intended for the PLC dropped due to: • No STX byte(s) found • No ETX byte(s) found • Time-outs • Too large of packet • Receive buffer queue overflows
Attribute 6 Parity Error Count	This attribute counts the number of received serial packets dropped due to parity errors.
Attribute 7 Framing Error Count	This attribute counts the number of received serial packets dropped due to framing errors.
Attribute 8 Overrun Error Count	This attribute counts the number of received serial packets dropped due to overrun error incidents.
Attribute 9 Received Consumed Sequence Error Count	This attribute counts the number of received consumed sequence number errors. The DeviceMaster only increments this number when all of the following statements are true: • You selected the *Unsolicited - Write-to-Tag-Synced* method of receive communication. • DeviceMaster receives a serial packet. • The *Consumed Sequence number* is out of sync. (It is not equal to the *Produced Sequence number* or equal to the *Produced Sequence number* minus one.)
Attribute 10 Duplicate Transmit Sequence Number Error Count	This attribute counts the number of *Duplicate Transmit Sequence Number* errors. The DeviceMaster only increments this number when the following statements are true: • You enabled the *Transmit Sequence Number Checking configuration* option. (See *Attribute 16 - Serial Port Transfer Options* on Page 27 for additional information.) • DeviceMaster receives a transmit message with a sequence number equal to the previous sequence number. (The DeviceMaster expects this sequence number to be incremented by one from the sequence number in the previous transmit message.)

Attribute	Description
Attribute 11 Unexpected Transmit Sequence Number Error Count	This attribute counts the number of *Unexpected Transmit Sequence Number* errors. The DeviceMaster increments this number when the following statements are true: • You enabled the *Transmit Sequence Number Checking* configuration option. (See *Attribute 16 - Serial Port Transfer Options* on Page 27 for additional information.) • DeviceMaster receives a transmit message with a sequence number that is not equal to either the previous transmit sequence number or the previous transmit sequence number plus one. (The DeviceMaster expects this sequence number to be incremented by one with each new transmit message.)
Attribute 12 Dropped Packet to Application Count	This attribute counts the number of received serial packets intended for the application dropped due to: • No STX byte(s) found • No ETX byte(s) found • Time-outs • Too large of packet • Receive buffer queue overflows

3.2.4. Socket Port Configuration Object Definition (73 Hex)

The Socket Port Configuration vendor specific object defines the protocol by which:

- A PLC can communicate with an Ethernet TCP/IP device through a DeviceMaster over Ethernet/IP.
- An optional application can communicate with the Ethernet device through the DeviceMaster over an Ethernet TCP/IP socket port.
- The optional data filtering and data extraction functions can be implemented.

Note: *The instance number corresponds to the associated socket port number on the* DeviceMaster. *(Socket port numbers can be numbered form 1 to N.)*

You can disregard this object definition if you configure the DeviceMaster using the embedded *Ethernet | Device Configuration* web page. Use *Chapter 4. Embedded Configuration Pages* on Page 85 to configure the DeviceMaster using the embedded web pages.

3.2.4.1. Class Attributes

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Revision	UINT	1	Get
2	Max Instance	UINT	Number of ports on the DeviceMaster	Get
3	Num Instances	UINT	Number of ports on the DeviceMaster	Get

3.2.4.2. Instance Attributes

Note: *Get returns the last command sent.*

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Socket Port Device	UDINT	0=Raw/ASCII Data Device	Set/Get

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
2	Socket Port Commands	DWORD	1=Reset Socket Port 2=Save in Flash 4=Clear Sequence Counters 8=Clear Statistics Counters	Set/Get
3	Device Socket Enable	USINT	0=Disabled 1=Enabled	Set/Get
4	Device Listen Enable	USINT	0=Disabled 1=Enabled	Set/Get
5	Device Listen Socket Port	UINT	0-65535	Set/Get
6	Device Connect Mode	USINT	0=Never 1=Connect Always 2=Connect On Data	Set/Get
7	Device Disconnect Mode	USINT	0=Never 1=Disconnect On Idle	Set/Get
8	Device Connect Socket Port	UINT	0-65535	Set/Get
9	Device Connect IP Address	UDINT	00000000 Hex to FFFFFFFF Hex (Mask=255.255.255.255)	Set/Get
10	Device Idle Timeout	UDINT (msec)	0 to FFFFFFFF Hex	Set/Get
11	Receive Timeout Between Packets (if no ETX or time to wait for ETX value)	UINT (msec)	0 to 65535 (Default = 0)	Set/Get
12	PLC IP Address	UDINT	0000000 Hex to FFFFFFFF Hex (Mask = 255.255.255.255)	Set/Get
13	Receive (DeviceMaster to PLC) Ethernet Data Transfer Method	USINT	0=OFF 1=Unsolicited - Write-to-Tag 2=Unsolicited - Write-to-Tag-Synced 3=Polling 4=Class 1 (Default=4)	Set/Get
14	PLC Controller Slot Number	USINT	0 to Max Slot Number on PLC (Default = 0)	Set/Get
15	Socket Port Transfer Options	WORD (bitwise OR)	01 Hex = (PLC/SLC) Rx MS Byte First 02 Hex = (PLC/SLC) Tx MS Byte First 04 Hex = Tx Sequence Number Checking 08 Hex = Disable Queuing of Non-Filtered Rx Messages 10 Hex = Drop oversized received packets	Set/Get
16	Maximum PLC Update Rate (No more than one message per time period.)	UINT (msec)	10-65535 (Default = 40)	Set/Get

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
17	Maximum Receive Data Packet Size	UINT	1-2048 Default = Dependent on the number of serial ports.	Set/Get
18	Received (DeviceMaster to PLC) Produced Data Tag Name	STRING (Array of 40 SINTS)	ASCII String	Set/Get
19	Application Socket Enable	USINT	0=Disabled 1=Enabled	Set/Get
20	Application Listen Enable	USINT	0=Disabled 1=Enabled	Set/Get
21	Application Connect Mode	USINT	0=Never 1=Connect Always 2=Connect On Data	Set/Get
22	Application Disconnect Mode	USINT	0=Never 1=Disconnect On Idle	Set/Get
23	Application Listen Socket Port	UINT	0-65535	Set/Get
24	Application Connect Socket Port	UINT	0-65535	Set/Get
25	Application Connect IP Address	UDINT	00000000 Hex to FFFFFFFF Hex (Mask=255.255.255.255)	Set/Get
26	Application Idle Timeout	UDINT (msec)	0 to FFFFFFFF Hex	Set/Get
27	To PLC Filter Mode	USINT	0=Off 1=String (128 byte maximum) 2=RFID (EPC Global formats) 3= Barcode (UPC/EAN formats)	Set/Get
28	To Application Filter Mode	USINT	0=Off 1=String (128 byte maximum) 2=RFID (EPC Global formats) 3= Barcode (UPC/EAN formats)	Set/Get
29	Discard Unrecognized Data Mode (RFID and Barcode Filter mode only)	USINT	0=Off 1=To PLC 2=To application 3=To PLC/application	Set/Get
30	RFID Antenna Grouping	USINT	0=None 1=Groups of Twos 2=Groups of Threes 3=Groups of Fours 4=First Two Only 5=First Three Only	Set/Get

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
31	To PLC Filter Options	WORD (Bitwise OR)	01 Hex = Encoding Scheme 02 Hex = Filter Code 04 Hex = Antenna Number 08 Hex = Company Code 10 Hex = Product/Location Code 20 Hex = Serial Number	Set/Get
32	To Application Filter Options	WORD (Bitwise OR)	01 Hex = Encoding Scheme 02 Hex = Filter Code 04 Hex = Antenna Number 08 Hex = Company Code 10 Hex = Product/Location Code 20 Hex = Serial Number	Set/Get
33	Filter Age Time	UDINT (msec)	0 - FFFFFFFF Hex	Set/Get
34	RFID Reader Interface Type	UINT	0=Unspecified 10=Alien (Text Mode) 11=Alien (Terse Mode) 20=Intermec (Hex ASCII Mode)	Set/Get
35	Barcode Formats (Barcode Filtering Only)	UINT	**Standard 12-14 Digit Format** (mask = 000F Hex) 00 Hex=NONE 01 Hex=Five Company/Five Product Digits 02 Hex=Six Company/Four Product Digits 03 Hex=Seven Company/Three Product Digits 04 Hex=Eight Company/Two Product Digits 05 Hex=Nine Company/One Product Digits **Eight Digit Format** (mask = 00F0 Hex) 00 Hex=NONE 10 Hex=EAN-8; Two Company/Five Product Digits 20 Hex=EAN-8; Three Company/Four Product Digits 30 Hex=UPC-E	Set/Get

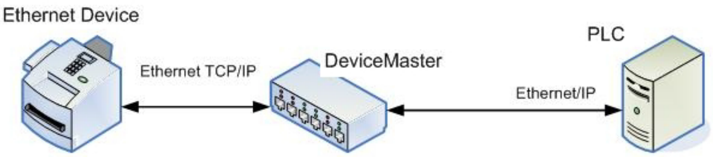
Attribute ID	Name	Data Type	Data Value(s)	Access Rule
36	*As defined for Get/Set All*			Set/Get
	PLC Transmit STX Append Value			
	Structure of:			
	Value1 (Only valid if Length not zero)	USINT	0 to 255	
	Value2 (Only valid if Length=2)	USINT	0 to 25	
	As defined for Get/Set Single			
	PLC Transmit STX Append Value			
	Structure of:			
	Length	USINT	0,1,2 (0=No STX) (Default=0)	
	Value1 (Only valid if Length not zero)	USINT	0 to 255	
	Value2 (Only valid if Length=2)	USINT	0 to 255	
37	*As defined for Get/Set All*			Set/Get
	PLC Transmit ETX Append Value			
	Structure of:			
	Value1 (Only valid if Length not zero)	USINT	0 to 255	
	Value2 (Only valid if Length=2)	USINT	0 to 25	
	As defined for Get/Set Single			
	PLC Transmit ETX Append Value **Structure of**:			
	Length	USINT	0,1,2 (0=No ETX) (Default=0)	
	Value1 (Only valid if Length not zero)	USINT	0 to 255	
	Value2 (Only valid if Length=2)	USINT	0 to 255	

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
38	Receive STX Detect Value			Set/Get
	Structure of:			
	Length	USINT	0,1,2 (0=No STX) (Default=0)	
	Value1 (Only valid if Length not zero)	USINT	0 to 255	
	Value2 (Only valid if Length=2)	USINT	0 to 255	
39	Receive ETX Detect Value			Set/Get
	Structure of:			
	Length	USINT	0,1,2 (0=No ETX) (Default=0)	
	Value1 (Only valid if Length not zero)	USINT	0 to 255	
	Value2 (Only valid if Length=2)	USINT	0 to 255	
40	Application Transmit STX Append Value			Set/Get
	Structure of:			
	Length	USINT	0,1,2 (0=No STX) (Default=0)	
	Value1 (Only valid if Length not zero)	USINT	0 to 255	
	Value2 (Only valid if Length=2)	USINT	0 to 255	
41	Application Transmit ETX Append Value			Set/Get
	Structure of:			
	Length	USINT	0,1,2 (0=No ETX) (Default=0)	
	Value1 (Only valid if Length not zero)	USINT	0 to 255	
	Value2 (Only valid if Length=2)	USINT	0 to 255	
42	Transmit (PLC to DeviceMaster) Data Transfer Method	USINT	0=Write-Message 1=Class1 (Default = 1)	Set/Get
43	Reserved	USINT	0	Get
44	Maximum Transmit Data Packet Size	UINT	1-2048 (Default= dependent on number of serial ports). ***Note:*** *Applicable only to Class1 Transmit Transfer mode*	Set/Get

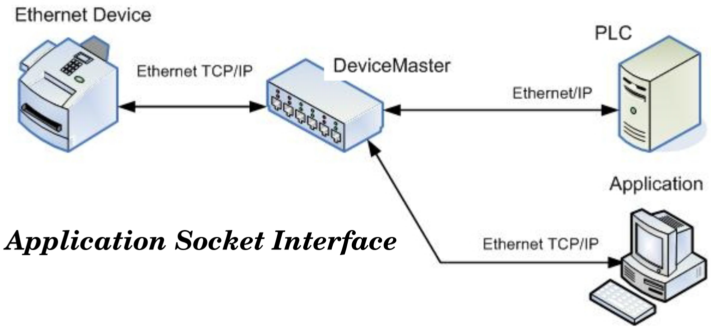
3.2.4.3. Common Services

Service Code	Implemented in Class	Implemented in Instance	Service Name
01 Hex	Yes	Yes	Get_Attributes_All
02 Hex	No	Yes	Set_Attributes_All
0E Hex	Yes	Yes	Get_Attribute_Single
10 Hex	No	Yes	Set_Attribute_Single

3.2.4.4. Instance Attribute Definitions

Attribute	Description
Attribute 1 Socket Port Device Type	This attribute indicates the socket port device type. **Raw/ASCII Data Device** is the only currently supported option for EtherNet/IP.
Attribute 2 Socket Port Commands	DeviceMaster supports the following commands: • **Reset socket port -** This option resets the device socket port, application socket port, and the statistics counters. Generally this is not required when changing socket port settings because the DeviceMaster detects the changes and, if necessary, automatically reset the socket connection(s). • **Save in Flash** - This option saves the socket port configuration in flash memory. These settings are restored when you reboot the DeviceMaster. • **Clear sequence counters** - This option clears the Receive Produced, Receive Consumed, and Transmit Produced sequence counters for the selected socket port. • **Clear statistics counters** - This option clears the statistics counters for the selected socket port.
Attribute 3 Device Socket Enable	This attribute enables/disables the Device Socket Interface. Enabling this function allows an Ethernet device to be connected to the DeviceMaster via an Ethernet TCP/IP socket. ***Device Socket Interface***  Ethernet Device Ethernet TCP/IP DeviceMaster PLC Ethernet/IP
Attribute 4 Device Listen Enable	Enabling this setting allows the device to connect to the DeviceMaster via an Ethernet TCP/IP socket. • 0 = Disables listening - The DeviceMaster does not accept connection attempts. • 1 = Enables listening - The DeviceMaster does accept connection attempts from the specified socket.
Attribute 5 Device Listen Socket Port	This is the socket port number on the DeviceMaster the device connects to if **Device Listen Enable** is enabled.

Attribute	Description (Continued)
Attribute 6 Device Connect Mode	This setting controls if and how the DeviceMaster attempts to connect to the device at the specified IP Address and socket. • 0 = Never - The DeviceMaster does not attempt to connect to the device. • 1 = Connect Always - Attempts to connect to the device until a connection is made. • 2 = Connect On Data - The DeviceMaster does not attempt to connect to the device until there is data to send to the device from either the PLC or application. Once data is received for the socket device, the DeviceMaster attempts to connect to the device until a connection is made.
Attribute 7 Device Disconnect Mode	This setting controls if and how the DeviceMaster disconnects from a socket device. • 0 = Never - The DeviceMaster does not disconnect from the device. • 1 = Disconnect On Idle - The DeviceMaster disconnects when there has been no transmit or received data between the socket device and PLC and/or application for a specified Idle period. (Please see Attribute 10.)
Attribute 8 Device Connection Socket Port	This is the device socket port number the DeviceMaster connects to if the **Device Connect Mode** is set to either **Connect Always** or **Connect On Data**.
Attribute 9 Device Connection IP Address	This is the device IP address the DeviceMaster connects to if the **Device Connect Mode** is set to either **Connect Always** or **Connect On Data**. The IP address is in hexadecimal format. For example, an IP address of 10.1.2.100 is 0A010264 in hexadecimal.
Attribute 10 Device Idle Timeout	The idle timeout period in milliseconds that is used if the **Device Disconnect Mode** is set to **Disconnect On Idle**.
Attribute 11 Receive Timeout Between Packets	This attributes specifies: • How long DeviceMaster waits (in milliseconds) if the Receive ETX length is not zero (0) and it does not receive an ETX byte sequence. • The time to wait (in milliseconds) between Ethernet packets if the Receive ETX Detect length is set to zero (0).
Attribute 12 PLC IP Address	This attribute specifies the IP address in hexadecimal format for the PLC EtherNet/IP card. For example, an IP address of 10.1.2.100 is 0A010264 in hexadecimal. ***Note:*** *The Polling method does not use this attribute.*
Attribute 13 Receive (DeviceMaster to PLC) Ethernet Data Transfer Method	This attribute specifies *Ethernet data transfer* method used by the DeviceMaster for the specified socket port. Refer to *Chapter 2. Data Transfer* on Page 13 for a detailed explanation of these methods.
Attribute 14 PLC Controller Slot Number	This attribute specifies the slot number on the PLC where the controller resides. The slot numbers typically start at zero (0) for the first slot. ***Note:*** *This is generally zero for CompactLogix PLCs. The Polling method does not use this attribute.*

Attribute	Description (Continued)
Attribute 15 Socket Port Data Transfer Options	This attribute specifies the socket port transfer options. The following options are supported: • **01 Hexadecimal (Bit 0)** = (PLC-5/SLC only) Receive Most Significant (MS) byte of 16 bit integer first. The default is transmit least significant (LS) byte first. • **02 Hexadecimal (Bit 1)** = (PLC-5/SLC only) Transmit Most Significant (MS) byte of 16 bit integer first. The default is transmit least significant (LS) byte first. • **04 Hexadecimal (Bit 2)** = Tx Sequence Number Checking - DeviceMaster rejects messages with duplicate sequence numbers, (that is, the same sequence number as the previous transmit data message), and increments the Duplicate Transmit Sequence Error Count. - DeviceMaster transmits messages with unexpected transmit sequence numbers, (that is, sequence numbers that are not the same as or are not equal to the previous sequence number plus one) increments the Unexpected Transmit Sequence Error Count. • **08 Hexadecimal (Bit 3)** = Disable Queuing of Non-Filtered Rx Messages to the PLC. If filtering is disabled, only the last message received is sent to the PLC. • **10 Hexadecimal (Bit 4)** = Drop oversized received data packets.
Attribute 16 Maximum PLC Update Rate	The maximum rate (or minimum time interval) in milliseconds, that messages are sent to the PLC tag in the *Write-To-Tag receive* method. This attribute configures the DeviceMaster to space the messages to the PLC to prevent overrunning of data before the PLC can process it.
Attribute 17 Maximum Receive Data Packet Size	Specifies the maximum acceptable size of a received Ethernet packet. The maximum received Ethernet packet size is 2048 bytes while operating in Class 1, *Write-to-Tag/File* or *Write-to-Tag-File-Synced receive* modes.
Attribute 18 Receive (DeviceMaster to PLC) Data Tag Name	This attribute specifies the PLC tag name. It indicates where to write received data while operating in the *Write-to-Tag* or *Write-to-Tag-Synced receive* method. ***Note:*** *This attribute is not used in the Polling or Class 1 methods.*
Attribute 19 Application Socket Enable	This attribute enables/ disables the Application Socket Interface. Enabling this function allows an application to be connected to the device socket port. If both the PLC and application are connected to the device socket port, both can transmit to and receive data from the device socket port. However, the PLC and application cannot communicate directly to each other.  Ethernet Device Ethernet TCP/IP DeviceMaster PLC Ethernet/IP Application Ethernet TCP/IP ***Application Socket Interface***
Attribute 20 Application Listen Enable	Enabling this setting allows the application to connect to the DeviceMaster via an Ethernet TCP/IP socket. • 0 = Disables listening - The DeviceMaster does not accept connection attempts. • 1 = Enables listening - The DeviceMaster accepts connection attempts from the specified socket.

Attribute	Description (Continued)
Attribute 21 Application Connect Mode	This setting controls if and how the DeviceMaster attempts to connect to the application at the specified IP address and socket. • 0 = Never - The DeviceMaster does not attempt to connect to the application. • 1 = Connect Always - Attempts to connect to the application until a connection is made. • 2 = Connect On Data - The DeviceMaster does not attempt to connect to the application until there is data to send to the application. Once data is received from the socket device, the DeviceMaster attempts to connect to the application until a connection is made.
Attribute 22 Application Disconnect Mode	This setting controls if and how the DeviceMaster disconnects from an application. • 0 = Never - The DeviceMaster does not disconnect from the application. • 1 = Disconnect On Idle - The DeviceMaster disconnects when there has been no transmit or received data between the socket device and application for a specified Idle period. (See Attribute 32.)
Attribute 23 Application Listen Socket Port	This is the socket port number on the DeviceMaster the application connects to if **Application Listen Enable** is enabled.
Attribute 24 Application Connection Socket Port	This is the application socket port number the DeviceMaster connects to if the **Application Connect Mode** is set to either **Connect Always** or **Connect On Data**.
Attribute 25 Application Connection IP Address	This is the application IP address the DeviceMaster connects to if the **Application Connect Mode** is set to either **Connect Always** or **Connect On Data**.
Attribute 26 Application Idle Timeout	The idle timeout period in milliseconds that is used if the **Application Disconnect Mode** is set to **Disconnect On Idle**.
Attribute 27 To PLC Filter/Data Extraction Mode	The *filter/data extraction* mode to be employed on data to be sent to the PLC. • 0 = Off • 1 = String (128 char max) - Raw/ASCII data is filtered up to 128 characters (or bytes) in length. • 2 = RFID (EPCglobal formats) - RFID data in any of the EPCglobal formats is filtered, the associated parameters is extracted, and the extracted data and RFID tag is sent to the PLC in a specified format. • 3 = Barcode (UPC/EAN formats) - Barcode data in specified UPC/EAN formats is filtered, the associated parameters is extracted, and the extracted data and barcode is sent to the PLC in a specified format. See the barcode format definitions in Attribute 41 on Page 32. See the *DeviceMaster UP Filtering and Data Extraction Reference Guide* (Page 8) for further details.

Attribute	Description (Continued)
Attribute 28 To Application Filter/ Data Extraction Mode	The *filter/data extraction* mode to be employed on data to be sent to the application. • 0 = Off • 1 = String (128 char max) - Raw/ASCII data is filtered up to 128 characters (or bytes) in length. • 2 = RFID (EPCglobal formats) - RFID data in any of the EPCglobal formats is filtered, the associated parameters is extracted, and the extracted data and RFID tag is sent to the application in a specified format. • 3 = Barcode (UPC/EAN formats) - Barcode data in specified UPC/EAN formats is filtered, the associated parameters is extracted, and the extracted data and barcode is sent to the application in a specified format.See the barcode format definitions in Attribute 41 on Page 32. The *application filter* mode can be set independently of the *PLC filtering* mode. The only exceptions are: • If the *PLC filter* mode is set to RFID, the *application filter* mode cannot be set to **Barcode**. • If the *PLC filter* mode is set to Barcode, the *application filter* mode cannot be set to **RFID**. See the *DeviceMaster UP Filtering and Data Extraction Reference Guide* (Page 8) for further details.
Attribute 29 Discard Unrecognized Data Mode	This attribute controls what to do with unrecognized RFID or barcode data. • 0 = Off - Send unrecognized data to the PLC and/or application. • 1 = Discard unrecognized data to the PLC. Allow sending of unrecognized data to the application. • 2 = Discard unrecognized data to the application. Allow sending of unrecognized data to the PLC. • 3 = Discard unrecognized data to both the PLC and application.
Attribute 30 RFID Antenna Grouping	This attribute is applicable only to RFID filtering and only if the Antenna filtering option is enabled. It allows the DeviceMaster to filter RFID tags based on Antenna groupings. The possible groupings are:

Setting	Group 1 Antennas	Group 2 Antennas	Group 3 Antennas	Group N Antennas
None	1	2	3	4
Groups of Twos	1,2	3,4	5,6	Etc.
Groups of Threes	1,2,3	4,5,6	7,8,9	Etc.
Groups of Fours	1,2,3,4	5,6,7,8	9,10,11,12	Etc.
First Two Only	1,2	3	4	N+1
First Three Only	1,2,3	4	5	N+2

Attribute	Description (Continued)
Attribute 31 To PLC Filtering Options	This attribute defines the RFID filtering criteria to the PLC. If an option is enabled, it is used to decide when an RFID tag can be filtered or sent to the PLC. • 01 Hex = **Encoding/Numbering -** Include the **Encoding/Numbering** code in the filtering criteria, which is part of the RFID tag or barcode data. • 02 Hex = **Filter Value -** Include the **Filter Value** in the filtering criteria, which is part of the RFID tag data. • 04 Hex = **Antenna -** Include the **Antenna** number in the filtering criteria. This is data from the RFID reader and not from the RFID tag. • 08 Hex = **Company -** Include the **Company** code in the filtering criteria, which is part of the RFID tag or barcode data. • 10 Hex = **Product/Location** - Include the **Product/Location** code in the filtering criteria, which is part of the RFID tag or barcode data. • 20 Hex = **Serial Number** - Include the **Serial Number** in the filtering criteria, which is part of the RFID tag data. See the *DeviceMaster UP Filtering and Data Extraction Reference Guide* (Page 8) for further details.
Attribute 32 To Application Filtering Options	This attribute defines the RFID filtering criteria to the application. If an option is enabled, it is used to decide when an RFID tag can be filtered or sent to the application. • 01 Hex = **Encoding/Numbering -** Include the **Encoding/Numbering** code in the filtering criteria, which is part of the RFID tag or barcode data. • 02 Hex = **Filter Value -** Include the **Filter Value** in the filtering criteria, which is part of the RFID tag data. • 04 Hex = **Antenna -** Include the **Antenna** number in the filtering criteria. This is data from the RFID reader and not from the RFID tag. • 08 Hex = **Company -** Include the **Company** code in the filtering criteria, which is part of the RFID tag or barcode data. • 10 Hex = **Product/Location** - Include the **Product/Location** code in the filtering criteria, which is part of the RFID tag or barcode data. • 20 Hex = **Serial Number** - Include the **Serial Number** in the filtering criteria, which is part of the RFID tag data. See the *DeviceMaster UP Filtering and Data Extraction Reference Guide* (Page 8) for further details.
Attribute 33 Filter Age Time	This attribute defines the time a filter string, RFID tag, or barcode continues to be filtered after the last time it was received. If an entry is received before the **Filter Age Time** has passed, the entry is filtered and the data is not sent to the PLC and/or application. However, if the **Filter Age Time** has passed, it passes filtering and be sent to the PLC and/or application.
Attribute 34 RFID Reader Interface Type	This attribute defines the expected RFID data format. Each format is unique and pertains to the RFID reader manufacturer. If a RFID reader is to be used and it provides a similar format to the ones listed below, it can also be used. • 0=Unspecified • 10 (Decimal) = Alien (Text Mode) • 11 (Decimal) = Alien (Terse Mode) • 20 (Decimal) = Intermec (Hex ASCII Mode) See the *DeviceMaster UP Filtering and Data Extraction Reference Guide* (Page 8) for further details.

<table>
<tr><th>Attribute</th><th>Description (Continued)</th></tr>
<tr><td><b>Attribute 35</b>
Barcode Formats</td><td>This attribute defines barcode format to be used for both standard and eight digit UPC labels. The term <i>standard</i> refers to UPC-A, EAN-13, JAN, and EAN-14 bar codes which all have ten company/product digits.
The standard and eight digit formats are selected independently and each operates independently. It is important to note that the barcode filtering/data extraction does not function if no format is selected.
<table>
<tr><th><u>Format</u></th><th>Numbering <u>Digits</u></th><th>Company <u>Digits</u></th><th>Product <u>Digits</u></th><th>Check <u>Digit</u></th></tr>
<tr><td><b><i>Standard Formats</i></b></td><td></td><td></td><td></td><td></td></tr>
<tr><td>None</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr>
<tr><td>Company-5/ Product-5</td><td>1-3</td><td>5</td><td>5</td><td>1</td></tr>
<tr><td>Company-6/ Product-4</td><td>1-3</td><td>6</td><td>4</td><td>1</td></tr>
<tr><td>Company-7/ Product-3</td><td>1-3</td><td>7</td><td>3</td><td>1</td></tr>
<tr><td>Company-8/ Product-2</td><td>1-3</td><td>8</td><td>2</td><td>1</td></tr>
<tr><td>Company-9/ Product-1</td><td>1-3</td><td>9</td><td>1</td><td>1</td></tr>
<tr><td><b><i>Eight Digit Formats</i></b></td><td></td><td></td><td></td><td></td></tr>
<tr><td>EAN-8 Number-2/Product 5</td><td>2</td><td>0</td><td>5</td><td>1</td></tr>
<tr><td>EAN-8 Number-3/Product 4</td><td>3</td><td>0</td><td>4</td><td>1</td></tr>
<tr><td>UPC-E</td><td></td><td></td><td></td><td></td></tr>
</table>
See the <i>DeviceMaster UP Filtering and Data Extraction Reference Guide</i> (Page 8) for further details.</td></tr>
<tr><td><b>Attribute 36</b>
PLC Transmit STX Append Value</td><td>You can set this attribute to append an STX (start of transmission) byte sequence which is configurable as 1 or 2-bytes to the beginning of the Ethernet packet before it is sent. The length indicates the number of STX bytes. The valid values for length are:
• <b>0 (zero) -</b> Setting this attribute to zero disables this function.
• <b>1 (one STX byte) -</b> Inserts one STX byte before the data.
• <b>2 (two STX bytes) -</b> Inserts two STX bytes before the data.
• <b>Value1 -</b> Specifies the transmit character associated with the first STX byte. (Only if the length is not zero.) You can specify a value between 0 and 255.
• <b>Value2 -</b> Specifies the transmit character associated with the second STX byte. (Only if length is two bytes.) You can specify a value between 0 and 255.</td></tr>
<tr><td><b>Attribute 37</b>
PLC Transmit ETX Append Value</td><td>You can set this attribute to append an ETX (end of transmission) byte sequence which is configurable as 1 or 2-bytes to the end of the Ethernet packet before it is sent. The length indicates the number of ETX bytes.
The valid values for length are:
• <b>0 (zero) -</b> Setting this attribute to zero disables this function.
• <b>1 (one ETX byte) -</b> Inserts one ETX byte at the end of the data.
• <b>2 (two ETX bytes) -</b> Inserts two ETX bytes at the end of the data.
• <b>Value -</b> Specifies the transmit character associated with the first ETX byte. (Only if the length is not zero.) You can specify a value between 0 and 255.
• <b>Value2 -</b> Specifies the transmit character associated with the second ETX byte. (Only if length is 2-bytes.) You can specify a value between 0 and 255.</td></tr>
</table>

Attribute	Description (Continued)
Attribute 38 Receive STX Detect Value	This attribute detects an STX (start of transmission) byte sequence which is configurable as 1 or 2-bytes when it receives a Ethernet packet. The length indicates the number of STX bytes. The valid values for length are: • **0 (zero)** - Setting this attribute to zero disables this function. When disabled, the DeviceMaster accepts the first byte received after the last ETX byte(s) as the start of the next data packet. • **1 (one STX byte)** - Scans Ethernet data for one STX byte. If the DeviceMaster finds an STX byte it collects the data. If the first byte is not the STX byte, the DeviceMaster discards the byte. The DeviceMaster continues to discard the bytes until the DeviceMaster finds an STX byte. • **2 (two STX bytes)** - Scans Ethernet data for two STX bytes. If the DeviceMaster finds two STX bytes it collects the data. If the first two bytes are not the STX bytes, the DeviceMaster discards the bytes. Bytes continue to be discarded until the DeviceMaster finds two STX bytes. • **Value1** - Specifies the character that represents the first STX byte. DeviceMaster looks for this character in the first STX byte. (Only if the length is not zero.) You can specify a value between 0 and 255. • **Value2** - Specifies the character that represents the second STX byte. DeviceMaster looks for this character in the first STX byte. (Only if length is two bytes.) You can specify a value between 0 and 255.
Attribute 39 Receive ETX Detect Value	This attribute detects an ETX (end of transmission) byte sequence which is configurable as 1 or 2 bytes marking the end of the Ethernet packet. The length indicates the number of ETX bytes. The valid values for length are: • **0 (zero)** - Setting this attribute to zero disables this function. When disabled, the DeviceMaster uses the Receive Timeout Between Packets (attribute 11) to indicate the end of data packet. • **1 (one ETX byte)** - Scans Ethernet data for one ETX byte. When the DeviceMaster finds an ETX byte it identifies the data as a Ethernet packet. • **2 (two ETX bytes)** - Scans Ethernet data for two ETX bytes. When the DeviceMaster finds two ETX bytes it identifies the data as a Ethernet packet. • **Value1** - Specifies the character to scan for in the first ETX byte. (Only if the length is not zero.) You can specify a value between 0 and 255. • **Value2** - Specifies the character to scan for in the second ETX byte. (Only if length is 2-bytes.) You can specify a value between 0 and 255.
Attribute 40 Application Transmit STX Append Value	You can set this attribute to append an STX (start of transmission) byte sequence which is configurable as 1 or 2-bytes to the beginning of the Ethernet packet before it is sent. The length indicates the number of STX bytes. The valid values for length are: • **0 (zero)** - Setting this attribute to zero disables this function. • **1 (one STX byte)** - Inserts one STX byte before the data. • **2 (two STX bytes)** - Inserts two STX bytes before the data. • **Value1** - Specifies the transmit character associated with the first STX byte. (Only if the length is not zero.) You can specify a value between 0 and 255. • **Value2** - Specifies the transmit character associated with the second STX byte. (Only if length is two bytes.) You can specify a value between 0 and 255.

Attribute	Description (Continued)
Attribute 41 Application Transmit ETX Append Value	You can set this attribute to append an ETX (end of transmission) byte sequence which is configurable as 1 or 2-bytes to the end of the Ethernet packet before it is sent. The length indicates the number of ETX bytes. The valid values for length are: • **0 (zero) -** Setting this attribute to zero disables this function. • **1 (one ETX byte) -** Inserts one ETX byte at the end of the data. • **2 (two ETX bytes) -** Inserts two ETX bytes at the end of the data. • **Value1** - Specifies the transmit character associated with the first ETX byte. (Only if the length is not zero.) You can specify a value between 0 and 255. • **Value2** - Specifies the transmit character associated with the second ETX byte. (Only if length is 2-bytes.) You can specify a value between 0 and 255.
Attribute 42 Transmit (PLC to DeviceMaster) Data Transfer Method	This attribute specifies the *Transmit Data Transfer* method used by the DeviceMaster. There are two methods that DeviceMaster can use to transfer data from the PLC to an Ethernet device. Refer to *Chapter 2. Data Transfer* on Page 13 for a discussion on transfer modes.
Attribute 44 Maximum Transmit Data Packet Size	Specifies the maximum acceptable size of transmit Ethernet packet. The maximum transmit Ethernet packet size is 2048 bytes while operating in the *Class1 transmit mode*. This attribute is not used in the *Write-Msg transmit* mode.

3.2.5. Socket Port Data Transfer Definition Object (74 Hex)

The *Socket Port Data Transfer vendor specific* object defines the attributes by which the PLC can transfer data to and from an Ethernet device, via a socket port through a DeviceMaster over EtherNet/IP.

Note: *There is one instance of this object per socket port. The instance number corresponds to the associated socket port number on the DeviceMaster. (Port numbers are numbered from one to N.)*

3.2.5.1. Class Attributes

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Revision	UINT	1	Get
2	Max Instance	UINT	Number of ports on the DeviceMaster	Get
3	Num Instances	UINT	Number of ports on the DeviceMaster	Get

3.2.5.2. Instance Attributes

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Transmit (PLC to DeviceMaster) message data			Set/Get
	Structure of:			
	Produced data sequence number	UINT	0-65535 (FFFF hex	
	Data length	UINT	1-440**	
	Data array	Array of USINT	0-255	
2	Receive (DeviceMaster to PLC) message data			
	Structure of:			Get
	Produced data sequence number	UINT	0-65535 (FFFF hex)	
	Data length	UINT	0-440*	
	Data array	Array of USINT	0-255	
3	Receive (DeviceMaster to PLC) produced data sequence number			
	Normally sent to the PLC from the DeviceMaster in data transfer.	UINT	0-65535 (FFFF hex)	Set/Get
4	Receive (DeviceMaster to PLC) consumed sequence number			
	Normally updated by the PLC in the *Write-to-Tag-Synced receive* method.	UINT	0-65535 (FFFF hex)	Set/Get
5	Transmit (PLC to DeviceMaster) produced data sequence number			
	Normally sent to the DeviceMaster from the PLC in data transfer.	UINT	0-65535 (FFFF hex)	Set/Get

* *For the Polling receive method only. The maximum sized socket port message in the Class 1, Write-to-Tag and Write-to-Tag-Synced receive method is 2048 bytes.*

** *For Write-to-Msg Transfer only. The maximum sized socket port message for Class 1 is 2048 bytes.*

3.2.5.3. Common Services

Service Code	Implemented in Class	Implemented in Instance	Service Name
0E Hex	Yes	Yes	Get_Attribute_Single
10 Hex	No	Yes	Set_Attribute_Single

3.2.5.4. Instance Attribute Definitions

Attribute	Description
Attribute 1 Transmit (PLC to DeviceMaster) Message Data	This attribute transmits data out of a socket port on the DeviceMaster. In most cases, incrementing the sequence number is optional. However, it is required if you enable the *Transmit Sequence Number Checking* option. (For more information, see *Attribute 16 - Serial Port Transfer Options* on Page 27.) The length must be at least one and a maximum of 440-bytes. A **Get** returns the last successfully transmitted data message.
Attribute 2 Receive (DeviceMaster to PLC) Message Data	This attribute provides the receive data while operating in the *Polling communication* method. • DeviceMaster increments the sequence number for each new socket port packet received. • A length of zero indicates no data was received on the specified socket port. • Two or more **Gets** may return the same data packet, but the messages will also have the same sequence number.
Attribute 3 Receive (DeviceMaster to PLC) Produced Data Sequence Number	Use this attribute to get and set the *Produced Data Sequence number*. This is the same *Produced Data Sequence number* sent to the PLC in all the *receive communication* methods.
Attribute 4 Receive (DeviceMaster to PLC) Consumed Data Sequence Number	Use this attribute to get and set the *Consumed Data Sequence number*. You can only specify this attribute when you are using the *Unsolicited - Write-to-Tag-Synced receive* method under Attribute 17 in the *Serial Port Configuration* object definition. When used, the PLC increments this attribute to indicate the data received has been consumed and it is now ready for another socket data packet. For more information, see the description of *Unsolicited -Write-to-Tag-Synced* in Attribute 17 on Page 27.
Attribute 5 Transmit (PLC to DeviceMaster) Produced Data Sequence Number	Use this attribute to get and set the *Transmit Produced Data Sequence number*. This is the same *Produced Data Sequence number* sent to the DeviceMaster in the *Transmit Message data*.

3.2.6. Assembly Object (For Class 1 Interface)

The EtherNet/IP specification requires that all Class 1 interfaces be provided through the Assembly Object interface. The Assembly Object interface is used to directly tie Vendor Specific objects to a standard interface, which the EtherNet/IP controller, or PLC, uses to communicate to the device.

For the DeviceMaster gateway, the Assembly Object corresponds to the Serial and Socket Data Transfer objects. Each instance of the Assembly Object corresponds to one or more of the Serial and/or Socket Data Transfer Object attributes.

The Assembly object defines the interface by which a Class 1 PLC or controller can:

- Request the receive data packets from the DeviceMaster.
- Write the transmit data to the DeviceMaster.

3.2.6.1. Class Attributes

This table shows the Class Attributes for the Assembly Object for a Class 1 interface.

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Revision	UINT	2	Get
2	Max Instance	UINT	113: (1-Port Models) 114: (2-Port Models) 116: (4-Port Models)	Get
3	Num Instances	UINT	4: (1-Port Models) 8: (2-Port Models 16: (4-Port Models)	Get
4	Optional Attribute List	UNIT	4	Get
6	Maximum Number Class Attribute	UNIT	7	Get
7	Maximum Number Instance Attribute	UNIT	4	Get

3.2.6.2. Instance Attributes

This table shows the Instance Attributes for the Assembly Object for a Class 1 interface.

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
3	Data	Array of BYTE	0-255	Get/Set
4	Data Length	UINT	Maximum number of bytes in attribute 3	Get

3.2.6.2.1. Instance Attribute Definitions: Attribute 3-Request/Write Data

Dependent on the instance number, this is either the receive data block and/or the transmit data block.

3.2.6.2.2. Instance Attribute Definitions: Attribute 4-Data Length

This is the maximum data length for each Assembly instance.

3.2.6.3. Common Services

This table shows the Common Services for the Assembly Object for a Class 1 interface.

Service Code	Implemented in Class	Implemented in Instance	Service Name
01 hex	Yes	No	Get_Attributes_All
0E hex	Yes	Yes	Get_Attribute_Single
10 hex	No	Yes	Set_Attribute_Single
02 hex	No	No	Set_Attribute_All

3.2.6.4. Instance Definitions (1-Port Models)

This table shows the Instance Definitions for the Assembly Object for a Class 1 interface for the 1-port models.

3.2.6.4.1. Assembly Input Instances

Assembly Instance Number	Description	Data Type	Data Values	Access Rule
101	Receive data from Serial Port 1	BYTE Array Length = (4+Maximum RX packet size)	0-255	Get
105	Receive data from Socket Port 1	BYTE Array Length = (4+Maximum RX packet size)	0-255	Get

3.2.6.4.2. Assembly Output Instances

Assembly Instance Number	Description	Data Type	Data Values	Access Rule
101	Transmit data to Serial Port 1	BYTE Array Length = (4+Maximum RX packet size)	0-255	Set
105	Transmit data to Socket Port 1	BYTE Array Length = (4+Maximum RX packet size)	0-255	Set

3.2.6.5. Instance Definitions (2-Port Models)

This table shows the Instance Definitions for the Assembly Object for a Class 1 interface for the 2-port models.

3.2.6.5.1. Assembly Input Instances

Assembly Instance Number	Description	Data Type	Data Values	Access Rule
101	Receive data from Serial Port 1	BYTE Array Length = (4+Maximum RX packet size)	0-255	Get
102	Receive data from Serial Port 2	BYTE Array Length = (4+Maximum RX packet size)	0-255	Get
105	Receive data from Socket Port 1	BYTE Array Length = (4+Maximum RX packet size)	0-255	Get
106	Receive data from Socket Port 2	BYTE Array Length = (4+Maximum RX packet size)	0-255	Get

3.2.6.5.2. Assembly Output Instances

Assembly Instance Number	Description	Data Type	Data Values	Access Rule
109	Transmit data to Serial Port 1	BYTE Array Length = (4+Maximum RX packet size)	0-255	Set
110	Transmit data to Serial Port 2	BYTE Array Length = (4+Maximum RX packet size)	0-255	Set
113	Transmit data to Socket Port 1	BYTE Array Length = (4+Maximum RX packet size)	0-255	Set
114	Transmit data to Socket Port 2	BYTE Array Length = (4+Maximum RX packet size)	0-255	Set

3.2.6.6. Instance Definitions (4-Port Models)

This table shows the Instance Definitions for the Assembly Object for a Class 1 interface for the 4-port models.

3.2.6.6.1. Assembly Input Instances

Assembly Instance Number	Description	Data Type	Data Values	Access Rule
101	Receive data from Serial port 1	BYTE Array Length = (4+Maximum RX packet size)	0-255	Get
102	Receive data from Serial port 2	BYTE Array Length = (4+Maximum RX packet size)	0-255	Get
103	Receive data from Serial port 3	BYTE Array Length = (4+Maximum RX packet size)	0-255	Get
104	Receive data from Serial port 4	BYTE Array Length = (4+Maximum RX packet size)	0-255	Get
105	Receive data from Socket port 1	BYTE Array Length = (4+Maximum RX packet size)	0-255	Get
106	Receive data from Socket port 2	BYTE Array Length = (4+Maximum RX packet size)	0-255	Get
107	Receive data from Socket port 3	BYTE Array Length = (4+Maximum RX packet size)	0-255	Get
108	Receive data from Socket port 4	BYTE Array Length = (4+Maximum RX packet size)	0-255	Get

3.2.6.6.2. Assembly Output Instances

Assembly Instance Number	Description	Data Type	Data Values	Access Rule
109	Transmit data to Serial Port 1	BYTE Array Length = (4+Maximum RX packet size)	0-255	Set
110	Transmit data to Serial Port 2	BYTE Array Length = (4+Maximum RX packet size)	0-255	Set
111	Transmit data to Serial Port 3	BYTE Array Length = (4+Maximum RX packet size)	0-255	Set
112	Transmit data to Serial Port 4	BYTE Array Length = (4+Maximum RX packet size)	0-255	Set
113	Transmit data to Socket Port 1	BYTE Array Length = (4+Maximum RX packet size)	0-255	Set
114	Transmit data to Socket Port 2	BYTE Array Length = (4+Maximum RX packet size)	0-255	Set
115	Transmit data to Socket Port 3	BYTE Array Length = (4+Maximum RX packet size)	0-255	Set
116	Transmit data to Socket Port 4	BYTE Array Length = (4+Maximum RX packet size)	0-255	Set

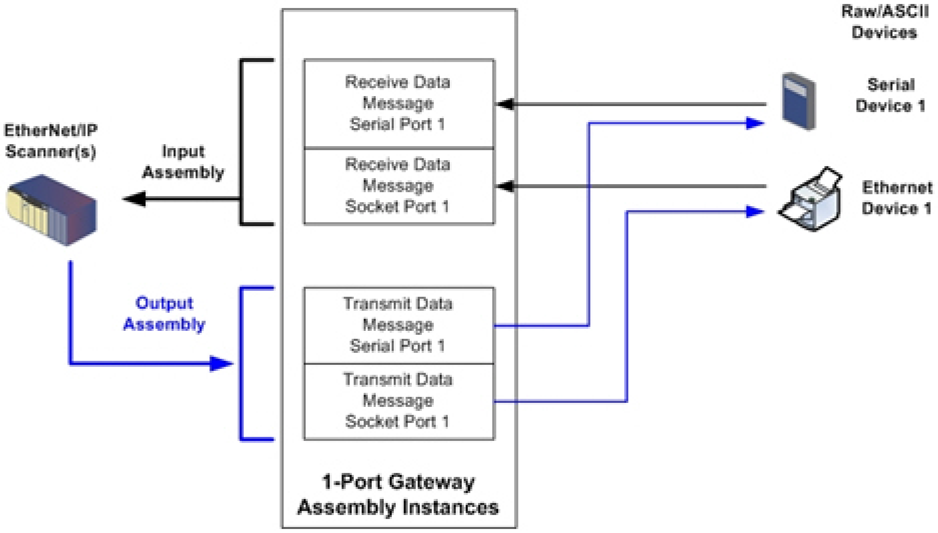
3.2.6.7. Overview of Assembly Interface

The Assembly interface is designed to:

- Provide access to all Receive and Transmit assemblies.
- Maximize flexibility for the PLC programmer.
- Minimize required PLC and gateway communication bandwidth.
- Be as easy to use as possible.

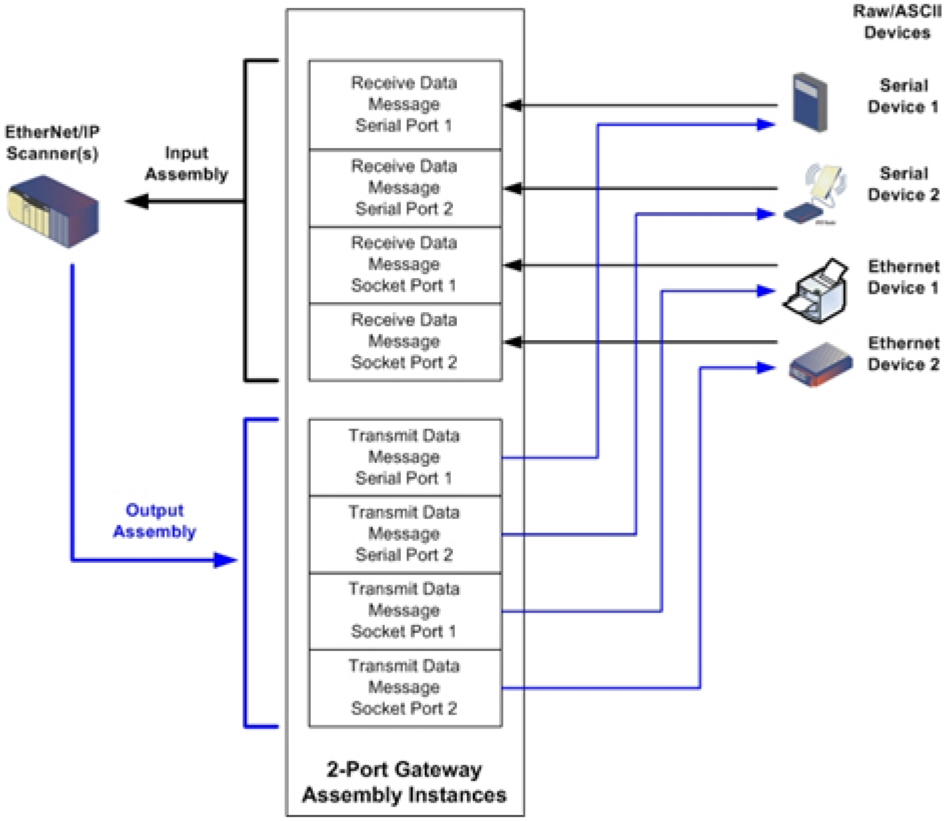
3.2.6.7.1. 1-Port Gateways

The following diagram illustrates the Assembly instances for a one port DeviceMaster gateway when all serial and socket interfaces are configured to Class 1. There is one Assembly input and output instance assigned to each serial and socket port.


Raw/ASCII Devices
EtherNet/IP Scanner(s)
Input Assembly
Receive Data Message Serial Port 1
Receive Data Message Socket Port 1
Serial Device 1
Ethernet Device 1
Output Assembly
Transmit Data Message Serial Port 1
Transmit Data Message Socket Port 1
1-Port Gateway Assembly Instances

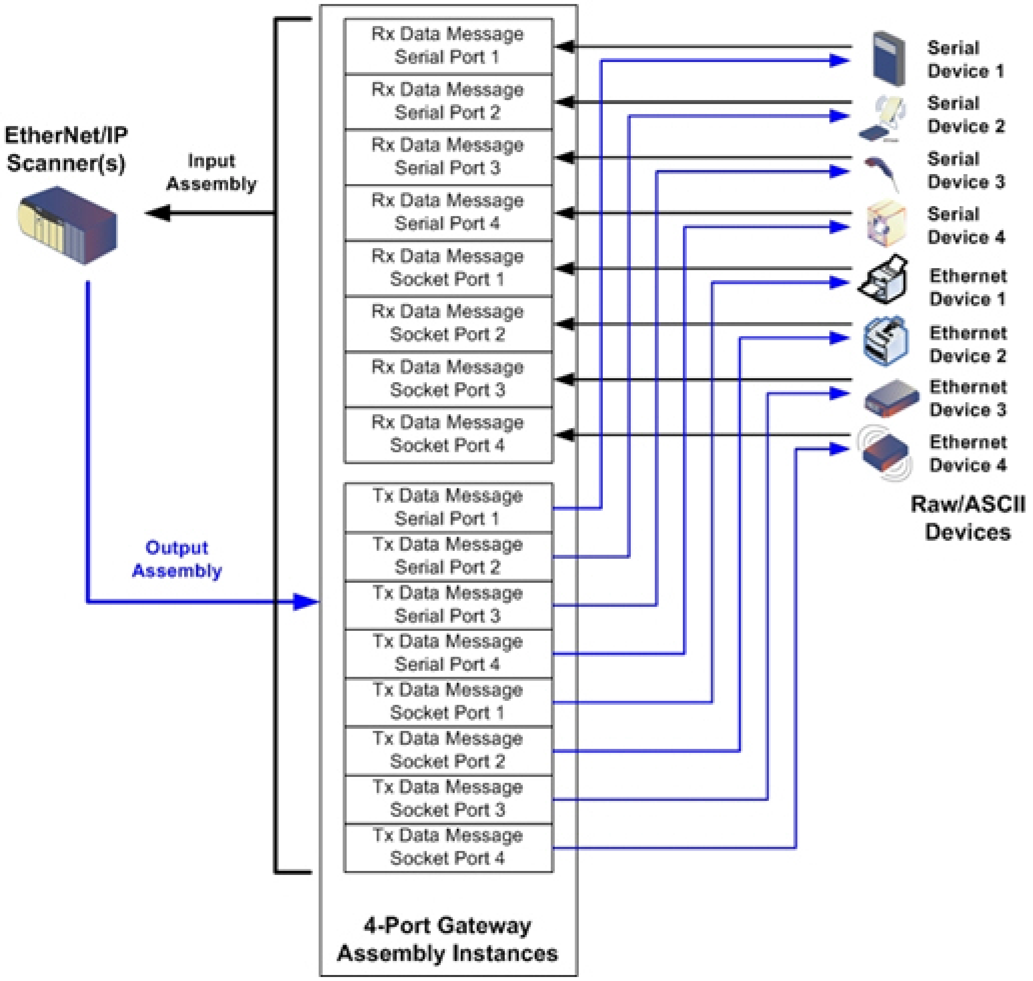
3.2.6.7.2. 2-Port Gateways

The following diagram illustrates the Assembly instances for a two port DeviceMaster gateway when all serial and socket interfaces are configured to Class 1. There is one Assembly input and output instance assigned to each serial and socket port.


Raw/ASCII Devices
EtherNet/IP Scanner(s)
Input Assembly
Receive Data Message Serial Port 1
Receive Data Message Serial Port 2
Receive Data Message Socket Port 1
Receive Data Message Socket Port 2
Serial Device 1
Serial Device 2
Ethernet Device 1
Ethernet Device 2
Output Assembly
Transmit Data Message Serial Port 1
Transmit Data Message Serial Port 2
Transmit Data Message Socket Port 1
Transmit Data Message Socket Port 2
2-Port Gateway Assembly Instances

3.2.6.7.3. 4-Port Gateways

The following diagram illustrates the Assembly instances for a four port DeviceMaster gateway when all serial and socket interfaces are configured to Class 1. There is one Assembly input and output instance assigned to each serial and socket port.


EtherNet/IP Scanner(s)
Input Assembly
Output Assembly
Rx Data Message Serial Port 1
Rx Data Message Serial Port 2
Rx Data Message Serial Port 3
Rx Data Message Serial Port 4
Rx Data Message Socket Port 1
Rx Data Message Socket Port 2
Rx Data Message Socket Port 3
Rx Data Message Socket Port 4
Tx Data Message Serial Port 1
Tx Data Message Serial Port 2
Tx Data Message Serial Port 3
Tx Data Message Serial Port 4
Tx Data Message Socket Port 1
Tx Data Message Socket Port 2
Tx Data Message Socket Port 3
Tx Data Message Socket Port 4
4-Port Gateway Assembly Instances
Serial Device 1
Serial Device 2
Serial Device 3
Serial Device 4
Ethernet Device 1
Ethernet Device 2
Ethernet Device 3
Ethernet Device 4
Raw/ASCII Devices

3.2.6.8. Grouping of Assembly Instances

In order to minimize the number of required I/O connections, the input and output assembly instances are organized as follows. The Input Assembly instances are grouped into one continuous array with no gaps between the instances. The same is also true for Output Assembly Instances.

3.2.6.8.1. 1-Port Models - Assembly Controller Access

		Serial Port 1		Socket Port 1	
	Assembly Instance Number	**Receive**	**Transmit**	**Receive**	**Transmit**
Read (Input) Rx Msg Data	101 (Serial Port 1)	X			
	105 (Socket Port 1)	X		X	
Write (Output) Tx Msg Data	109 (Serial Port 1)		X		
	113 (Socket Port 1)		X		X

Where:

- All accessible data can be read (input) and written (output) from one I/O connection.
- Controller Read (Input) access:
  - One or more input instances may be read with one I/O connection. (i.e. If addressing the instance 101, all input instances for both serial and socket data, 101 and 105 (for 1-port models), may be read in one connection.)
  - The length of the Read (Input) connection can range from 1 to the total length for all input instances.
  - Multiple controllers can have read access to the Input Assembly instances at one time.
- Controller Write (Output) access:
  - Only output instances may be written.
  - One or more output instances may be written to with one connection.
  - Starting at output instance 109, the length of the Write (Output) connection must be equal to either:
    - The length of the output instance 109.
    - The total length of all output instances (109 and 113).
  - Starting at output instance 113, the length of the Write (Output) connection must be equal to the length of the output instance 113.
  - Only one controller may have write access to an output instance.

3.2.6.8.2. 2-Port Models - Assembly Controller Access

	Assembly Instance Number	**Serial Port 1**		**Serial Port 2**		**Socket Port 1**		**Socket Port 2**	
		Rx	**Tx**	**Rx**	**Tx**	**Rx**	**Tx**	**Rx**	**Tx**
Read (Input) Rx Msg Data	101 (Serial Port 1)	X							
	102 (Serial Port 2)	X		X					
	105 (Socket Port 1)	X		X		X			
	106 (Socket Port 2)	X		X		X		X	

	Assembly Instance Number	Serial Port 1		Serial Port 2		Socket Port 1		Socket Port 2	
		Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx
Write (Output)Tx Msg Data	109 (Serial Port 1)		X						
	110 (Serial Port 2)		X		X				
	113 (Socket Port 1)		X		X		X		
	114 (Socket Port 2)		X		X		X		X

Where:

- All accessible data can be read (input) and written (output) from one I/O connection.
- Controller Read (Input) access:
  - One or more input instances may be read with one I/O connection. (i.e. If addressing the instance 101, all input instances for both serial and socket data, 101, 102, 105 and 106 (for 2-port models), may be read in one connection.)
  - The length of the Read (Input) connection can range from 1 to the total length of that instance and all following input instances.
  - Multiple controllers can have read access to the Input Assembly instances at one time.
- Controller Write (Output) access:
  - Only output instances may be written.
  - One or more output instances may be written to with one connection.
  - Starting at output instance 109, the length of the Write (Output) connection must be equal to either:
    - The length of output instance 109 (serial port 1).
    - The lengths of output instances 109 and 110 (serial ports 1 and 2).
    - The lengths of output instances 109, 110 and 112 (serial ports 1 and 2, socket port 1).
    - The total length of all output instances 109, 110, 113, and 114 (serial ports 1 and 2, socket ports 1 and 2).
  - Starting at output instance 110, the length of the Write (Output) connection must be equal to either:
    - The length of output instance 110 (serial port 2).
    - The lengths of output instances 110 and 113 (serial port 2 and socket port 1).
    - The lengths of output instances 110, 113, and 114 (serial port 2, socket ports 1 and 2).
  - Starting at output instance 113, the length of the Write (Output) connection must be equal to either:
    - The length of output instance 113 (socket port 1).
    - The lengths of the output instances 113 and 114 (socket port 1 and 2).
  - Starting at output instance 114, the length of the Write (Output) connection must be equal to the length of output instance 114.
  - Only one controller may have write access to an output instance.

3.2.6.8.3. 4-Port Models - Assembly Controller Access

	Assembly Instance Number	Serial Port 1		Serial Port 2		Serial Port 3		Serial Port 4		Socket Port 1		Socket Port 2		Socket Port 3		Socket Port 4	
		Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx
Read (Input) Rx Msg Data	101 Serial Port 1	X															
	102 Serial Port 2	X		X													
	103 Serial Port 3	X		X		X											
	104 Serial Port 4	X		X		X		X									
	105 Socket Port 1	X		X		X		X		X							
	106 Socket Port 2	X		X		X		X		X		X					
	107 Socket Port 3	X		X		X		X		X		X		X			
	108 Socket Port 4	X		X		X		X		X		X		X		X	
Write (Output) Tx Msg Data	109 Serial Port 1		X														
	110 Serial Port 2		X		X												
	111 Serial Port 3		X		X		X										
	112 Serial Port 4		X		X		X		X								
	113 Socket Port 1		X		X		X		X		X						
	114 Socket Port 2		X		X		X		X		X		X				
	115 Socket Port 3		X		X		X		X		X		X		X		
	116 Socket Port 4		X		X		X		X		X		X		X		X

Where:

- All accessible data can be read (input) and written (output) from one I/O connection.
- Controller Read (Input) access:
  - One or more input instances may be read with one I/O connection. (i.e. If addressing the instance 101,

all input instances for both serial and socket data, 101 to 108 (for 4-port models), may be read in one connection.)

- The length of the Read (Input) connection can range from 1 to the total length of that instance and all following input instances.
- Multiple controllers can have read access to the Input Assembly instances at one time.

- Controller Write (Output) access:
  - Only output instances may be written. One or more output instances may be written to with one connection.
  - Starting at an output instance, the length of the Write (Output) connection must be equal to the total length of the desired sequential output instances. The length cannot include only part of an output instance.
  - Only one controller may have write access to an output instance.

3.2.7. Informational Objects

The following object definitions are included for informational purposes only. While some software packages such as RSLinx make use of these objects, few PLC programmers will have a need to directly access them.

3.2.7.1. Identity Object (01 Hex, 1 instance)

The *Identity Object* provides identification of and general information about the DeviceMaster.

3.2.7.1.1. Class Attributes

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Revision	UINT	1	Get
2	Max Class	UINT	1	Get
3	Max Instance	UINT	1	Get
6	Maximum Number	UINT	7	Get
7	Maximum Number Instance Attribute	UINT	7	Get

3.2.7.1.2. Instance Attributes

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Vendor ID	UINT	909 (Comtrol)	Get
2	Device Type	UINT	43 (Generic product device)	Get
3	Product Code	UINT	As defined by Comtrol	Get
4	Revision (product or software release)			
	Structure of: Major Revision Minor Revision	 USINT USINT	 1 to 127 1 to 999	Get
5	Status	WORD	See *3.2.7.1.3. Status Word*, below.	Get
6	Serial Number	UDINT	1-FFFFFFFF hex	Get

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
7	Product Name **Structure of:** Name Length Name String	 USINT STRING	Length of string: DeviceMaster **xPnE** (where **x** is the number of serial ports and **n** is the number of Ethernet ports depending on the model)	Get

3.2.7.1.3. Status Word

Refer to Pages 52 of Volume 3.5 of the *CIP Common Specification*. The following applies to the Identity Object status word for the GW EIP/ASCII gateway.

Status Word Bit	Setting	Description
0	0	Ownership Flag. Does not apply to the DeviceMaster gateway.
1	0	Reserved.
2	0	DeviceMaster gateway is operating on the default configuration.
	1	The DeviceMaster gateway has a configuration other than the default configuration.
3	0	Reserved.
4-7	0101 (0x50)	Indicates that there is a major fault (either Bit 10 or Bit 11 is set).
	0100 (0x40)	Indicates the stored configuration is invalid.
	0011 (0x30)	Indicates the system is operational and there are no I/O (Class 1) connections.
	0110 (0x60)	Indicates the system is operational and there is at least one active I/O (Class 1) connection.
	0000	Indicates the system is not operational. It may be in any of the following states: • System startup. • Configuration in process. • Idle • Critical (major) fault.
8	0	No recoverable minor fault. No error history entry reported within the last ten seconds.
	1	Recoverable minor fault. The DeviceMaster gateway has reported an error within the last ten seconds and a major fault has not been detected.
9	1	Unrecoverable minor fault. Does not apply to the DeviceMaster gateway.
10	0	No recoverable major fault.
	1	A major recoverable fault exists. This is a fault that the DeviceMaster may be able to recover from by a system reset. If the system does not recover automatically, a system reset message or a power cycle of the DeviceMaster gateway may be required.

Status Word Bit	Setting	Description
11	0	No major unrecoverable fault.
	1	A major unrecoverable fault has occurred in the DeviceMaster. If the major fault is not corrected with a system reset or a power cycle, refer to the User Manual or call Comtrol support.
12-15	0	Reserved.

3.2.7.1.4. Common Services

Service Code	Implemented in Class	Implemented in Instance	Service Name
01 hex	No	Yes	Get_Attribute_All
05 hex	No	Yes	Reset
0E hex	Yes	Yes	Get_Attribute_Single

3.2.7.2. Message Router Object (02 Hex)

The *Message Router Object* provides a messaging connection point through which a client may address a service to any object or instance residing in the physical device.

3.2.7.2.1. Class Attributes

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Revision	UINT	1	Get
2	Max Class	UINT	1	Get
3	Max Instance	UINT	1	Get
4	Optional Attribute List	UINT	2	Get
5	Option Service List	UINT	1	Get
6	Maximum Number Class Attribute	UINT	7	Get
7	Maximum Number Instance Attribute	UINT	2	Get

3.2.7.2.2. Instance Attributes

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Object List **Structure of:** Number Classes	 UINT Array of UINT	 Number of supported class codes List of supported class codes	Get
2	Max Connections	UINT	128	Get

3.2.7.2.3. Common Services

Service Code	Implemented in Class	Implemented in Instance	Service Name
01 Hex	No	Yes	Get_Attribute_All (Class only)
0E Hex	Yes	Yes	Get_Attribute_Single (Instance only)

3.2.8. Connection Manager Object (06 Hex)

The *Connection Manager Object* provides services for connection and connection-less communications.

3.2.8.1. Class Attributes Object (06 Hex)

The following table displays the Class Attributes for the Connection Manager Object (06 hex).

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Revision	UINT	1	Get
2	Max Class	UINT	1	Get
3	Max Instance	UINT	1	Get
4	Optional Attribute List	UINT	8	Get
6	Maximum Number Class Attribute	UINT	7	Get
7	Maximum Number Instance Attribute	UINT	8	Get

3.2.8.2. Instance Attributes (06 Hex)

This table displays the Instance Attributes for the Connection Manager Object (06 hex).

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Open Requests	UINT	0-0xffffffff	Set/Get
2	Open Format Rejects	UINT	0-0xffffffff	Set/Get
3	Open Resource Rejects	UINT	0-0xffffffff	Set/Get
4	Open Other Rejects	UINT	0-0xffffffff	Set/Get
5	Close Requests	UINT	0-0xffffffff	Set/Get
6	Close Format Requests	UINT	0-0xffffffff	Set/Get
7	Close Other Requests	UINT	0-0xffffffff	Set/Get
8	Connection Time Outs	UINT	0-0xffffffff	Set/Get

3.2.8.3. Common Services Object (06 Hex)

This table displays the Common Services for the Connection Manager Object (06 hex).

Service Code	Implemented in Class	Implemented in Instance	Service Name
01 hex	Yes	Yes	Get_Attribute_All
02 hex	No	Yes	Set_Attribute_ALL
0E hex	Yes	Yes	Get_Attribute_Single
10 hex	No	Yes	Set_Attribute_Single
4E hex	N/A	N/A	Forward_Close
52 hex	N/A	N/A	Unconnected_Send
54 hex	N/A	N/A	Forward_Open
5A hex	N/A	N/A	Get_Connection_Owner
5B hex	N/A	N/A	Large_Forward_Open

3.2.9. Port Object (F4 Hex - 1 Instance)

The *Port Object* enumerates the CIP ports on the DeviceMaster.

3.2.9.1. Class Attributes

Attribute ID	**Name**	**Data Type**	**Data Value(s)**	Access Rule
1	Revision	UINT	2	Get
2	Max Instance	UINT	2	Get
3	Num Instances	UINT	2	Get
6	Maximum Number Class Attributes	UINT	9	Get
7	Maximum Number Class Attributes	UINT	10	Get
8	Entry Port	UINT	1	Get
9	All Ports	Array of UINT	[0]=0 [1]=0 [2]=1 (Vendor specific) [3]=1 (Backplane) [4]=TCP_IP_PORT_TYPE (4) [5]=TCP_IP_PORT_NUMBER(2)	Get

3.2.9.2. Instance Attributes

This table illustrates the Instance Attributes for the Port Object (F4 hex – Instance 1).

Attribute ID	**Name**	**Data Type**	**Data Value(s)**	**Access Rule**
1	Port Type	UINT	1	Get
2	Port Number	UINT	1	Get
3	Port Object **Structure of:** 16-bit word count in path Path	UINT Array of UINT	2 [0]=6420 Hex [1]=0124	Get
4	Port Name **Structure of:** String Length Port Name	USINT Array of USINT	10 Backplane	Get
7	Node Address	USINT[2]	10 hex, 0 hex	Get

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
10	Port Routing Capabilities	UDINT	17 hex: Bit 0: Routing of incoming Unconnected Messaging supported Bit 1: Routing of outgoing Unconnected Messaging supported Bit 2: Routing of incoming Transport Class 0/1 Connections supported Bit 4: Routing of incoming Transport Class 2/3 Connections supported	Get

This table illustrates the Instance Attributes for the Port Object (F4 hex – Instance 2).

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Port Type	UINT	4 (TCP/IP)	Get
2	Port Number	UINT	2 (TCP/IP)	Get
3	Port Object **Structure of:** 16-bit word count in path Path	UINT Array of UINT	2 [0]=F520 hex [1]=0124 hex	Get
4	Port Name **Structure of:** String Length Port Name	USINT Array of USINT	17 “Ethernet/IP Port”	Get
7	Node Address	USINT[2]	10 hex, 0 hex	Get
10	Port Routing Capabilities	UDINT	17 hex: Bit 0: Routing of incoming Unconnected Messaging supported Bit 1: Routing of outgoing Unconnected Messaging supported Bit 2: Routing of incoming Transport Class 0/1 Connections supported Bit 4: Routing of incoming Transport Class 2/3 Connections supported	Get

3.2.9.3. Common Services

Service Code	Implemented in Class	Implemented in Instance	Service Name
01 Hex	Yes	Yes	Get_Attribute_All
0E Hex	Yes	Yes	Get Attribute Single

3.2.10. TCP Object (F5 Hex - 1 Instance)

The *TCP/IP Interface Object* provides the mechanism to retrieve the TCP/IP attributes for DeviceMaster.

3.2.10.1. Class Attributes

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Revision	UINT	4	Get
2	Max Instance	UINT	1	Get
3	Num Instances	UINT	1	Get
4	Optional Attribute List	UINT	4	Get
6	Maximum Number Class Attribute	UINT	7	Get
7	Maximum Number Instance Attribute	UINT	13	Get

3.2.10.2. Instance Attributes

This table provides the Instance Attributes for the TCP Object (F5 Hex).

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Status	DWORD	0 = The Interface Configuration attribute has not been configured. 1 = The Interface Configuration attribute contains configuration obtained from DHCP or nonvolatile storage.	Get
2	Configuration Capability	DWORD	14 Hex (DHCP and SETTABLE)	Get
3	Configuration Control	DWORD	0=Use stored IP address (static IP address) 2=DHCP	Get
4	Physical Link Object **Structure of:** Path Size Path	 UINT Array of USINT	 2 [0]=20 Hex [1]=F6 Hex [2]=24 Hex [3]=01 Hex	Get

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
5	Interface Configuration **Structure of:** IP Address Network Mask Gateway Address Name Server Name Server 2 Domain Name Length Domain Name	UDINT UDINT UDINT UDINT UDINT UINT STRING	<IP address> <Network mask> <Gateway Addr> <Name server> <Name server2> <Length of name> <Domain name>	Set
6	Host Name **Structure of:** Host Name Length Host Name	UINT STRING	0 to 15 <Default=IP address>	Set
8	TTL (Time-to-Live) value for IP multicast packets	USINT	1 to 255 <Default = 1>	Set
9	IP Multicast Address Configuration	**Struct of:** USINT – Allocation Control USINT - Reserved UINT – Num Mcast UDINT –Start Multicast Address	Allocation Control 0=Default Algorithm 1 = Configuration Num Multicast Addresses: 1-32 Start Multicast Address: 239.192.1.0 to 239.255.255.255	Set
13	Encapsulation Inactivity Timeout	USINT	Number of seconds of inactivity before TCP connection or the DTLS session is closed.	Set

3.2.10.3. Common Services

Service Code	Implemented in Class	Implemented in Instance	Service Name
01 Hex	Yes	Yes	Get_Attribute_All
02 Hex	No	Yes	Set_Attribute_All
0E Hex	Yes	Yes	Get_Attribute_Single
10 Hex	No	Yes	Set_Attribute_Single

3.2.11. Ethernet Link Object (F6 Hex)

The *Ethernet Link* object maintains link-specific counters and status information for the Ethernet communications on the DeviceMaster.

3.2.11.1. Class Attributes

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Revision	UINT	4	Get
2	Max Instance	UINT	1 = One Ethernet Port Models 3 = Two Ethernet Port Models	Get
3	Num Instances	UINT	1 = One Ethernet Port Models 3 = Two Ethernet Port Models	Get
4	Optional Attribute List	UINT	4	Get
6	Maximum Number Class Attributes	UINT	7	Get
7	Maximum Number Instance Attributes	UINT	11	Get

3.2.11.2. Instance Attributes

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Interface speed (current operational speed)	UDINT	10=10-Mbit 100=100-Mbit	Get
2	Interface Flags (Current operational status)	DWORD	Bit 0 =link status (0=inactive) (1=active) Bit 1=Half/Full Duplex (0=half duplex) (2=full duplex) Bits 2-4: 00 = negotiation in progress 01 = negotiation failed 02 = negotiation failed speed OK 03 = negotiation success	Get
3	Physical Address	Array of 6 USINT	MAC address	Get
7	Interface Type	USINT	2 = Twisted Pair	Get
8	Interface State	USINT	1 = Interface is enabled and operational	Get
9	Admin State	USINT	1 = Interface enabled	Get

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
10	Interface Label	USINT16 Array of USINT	Length= 1 to 64 ASCII characters <Default = IP address in “xxx.xxx.xxx.xxx” format>	Get
11	Interface Capability	UDINT Struct of USINT Array of: UINT USINT	Capability bits - Interface capabilities; other than speed/ duplex Value = 6 Bit 1: Auto-negotiate Bit 2: Auto-MDIX Speed/Duplex Array Count = 1 Interface Speed - 10 or 100 Interface Duplex Mode 0 = Half duplex 1 = Full duplex	Get

3.2.11.3. Common Services

Service Code	Implemented in Class	Implemented in Instance	Service Name
01 Hex	Yes	Yes	Get_Attribute_All
0E Hex	Yes	Yes	Get_Attribute_Single

3.2.12. PCCC Object (67 Hex)

The *PCCC object* provides the ability to encapsulates, and then transmit and receive PCCC messages between devices on an EtherNet/IP network. This is the primary interface for the PLC-5 and SLC PLCs.

The PCCC Object does not support the following:

- Class Attributes
- Instance Attributes

3.2.12.1. Class Attributes

Not supported.

3.2.12.2. Instance Attributes

Not supported.

3.2.12.3. Instances

Supports Instance 1.

3.2.12.4. Common Services

Service Code	Implemented in Class	Implemented in Instance	Service Name
4B Hex	No	Yes	Execute_PCCC

3.2.12.5. Message Structure for Execute PCCC

Request Message Name	Data Type	Description
Length	USINT	Length of requestor ID
Vendor	UINT	Vendor number of requestor
Serial Number	UDINT	ASA serial number of requestor
CMD	USINT	Command byte
STS	USINT	0
TNSW	UINT	Transport word
FNC	USINT	Function code
PCCC_params	Array of USINT	CMD/FMC specific parameters

Response Message Name	Data Type	Description
Length	USINT	Length of requestor ID
Vendor	UINT	Vendor number of requestor
Serial Number	UDINT	ASA serial number of requestor
CMD	USINT	Command byte
STS	USINT	Status byte
TNSW	UINT	Transport word - Same value as request.
EXT_STS	USINT	Extended status (if error)
PCCC_params	Array of USINT	CMD/FMC specific result data

Supported PCCC Command Types	FNC	Description
0F Hex	67 Hex	PLC-5 typed write
0F Hex	68 Hex	PLC-5 typed read
0F Hex	A2 Hex	SLC 500 protected typed read with 3 address fields
0F Hex	AA Hex	SLC 500 protected typed write with 3 address fields

3.3. PLC-5/SLC and MicroLogix Interfaces

EtherNet/IP firmware supports EtherNet/IP connections to the PLC-5 and SLC PLCs. The PLC-5 and SLC interface supports:

- *Polling*, *Write-to-File*, and *Write-to-File-Synced receive* communication methods.
- PCCC based messages transferred by means of the PCCC object, including:
  - SLC Typed Read Message
  - SLC Typed Write Message
  - PLC-5 Typed Read Message (Logical ASCII address format)
  - PLC-5 Typed Write Message (Logical ASCII address format)
- Configuration of the DeviceMaster only through the embedded web page. There is no DeviceMaster configuration available through the PLC-5 and SLC PLCs. See the following topics for more information:
  - *6.5. SLC or MicroLogix PLC Programming Example Instructions* on Page 166
  - *6.6. PLC-5 PLC Programming Example Instructions* on Page 193
- Receive, transmit and statistics data.
- Standard PLC-5/SLC file naming conventions.
- Ethernet device interface via Ethernet TCP/IP sockets.
- Application interface via Ethernet TCP/IP sockets.
- Filtering and data extraction:
  - String filtering of up to 128 bytes.
  - RFID - EPCglobal tag data filtering and data extraction.
  - Barcode - UPC/EAN barcode data filtering and data extraction.
  - Independent filtering criteria to the PLC and application.
- Serial data packet transfers up to 1518 bytes and socket packet data transfers up to 2048 bytes in the *Write-to-File* and *Write-to-File-Synced receive* methods.
- Controlled message rate to the PLC when operating in the *Write-to-File receive* method. This is accomplished by setting the **Maximum PLC Update Rate**.

The primary differences between the PLC-5/SLC interface and the ControlLogix interfaces are:

- Since the PLC-5 and SLC PLCs operate on a file memory system, the PLC-5/SLC interface provides *Write-to-File* and *Write-to-File-Synced* communication methods. They operate in a very similar manner to the *Write-to-Tag* and *Write-to-Tag-Synced* methods available for the ControlLogix family of PLCs.
- Polling is performed through the PLC-5/SLC specific messages instead of accessing the Serial Port Data Transfer object.
- When configuring the DeviceMaster to operate in *Write-to-File* or *Write-to-File-Synced*, enter the file name starting with a **$** (i.e. **$N10:0**) for SLC and PLC-5, and with a # (i.e. **#N10:0**) for MicroLogix.
- The maximum serial and socket port packet sizes in *polling* mode are smaller due to PCCC message size limitations.
- The configuration options cannot be set through SLC or PLC-5 messages.

Note: *While ControlLogix PLCs support the SLC and PLC-5 messages, using those messages on ControlLogix PLCs is not recommended due to data size and performance considerations. For ease of programming, Comtrol recommends the Write-to-File receive method used with the Maximum PLC Update Rate option*

3.3.1. Requirements

Your PLC-5/SLC must support:

- MultipHop
- ControlLogix devices
- EtherNet/IP

The following tables list PLCs that support EtherNet/IP and the required firmware version for each PLC.

Note: *Older versions of the PLC firmware may or may not provide EtherNet/IP functionality. You must verify that an older version of the PLC firmware provides EtherNet/IP functionality before you can use it with* DeviceMaster. *If you need to update your PLC firmware, contact your Rockwell distributor.*

3.3.1.1. SLC 5/05

Models	Catalog Numbers	Required Firmware Version for EtherNet/IP
SLC 5/05	1747-L551	Series A: FRN 5 or later
	1747-L552, 1747-L553	Series C: FRN 3 or later

Reference: *SLC 500 Instruction Set, Appendix A Firmware History*, Rockwell Publication 1747-RM001D-EN-P

3.3.1.2. PLC-5

Models	Catalog Numbers	Required Firmware Version for EtherNet/IP
Ethernet PLC-5	1785-L20E, 1785-L40E, 1785-L80E	**Base EtherNet/IP functionality:** Series C: Revision N or later Series D: Revision E or later Series E: Revision D or later **Full EtherNet/IP Compliance:** Series C: Revision R or later Series D: Revision H or later Series E: Revision G or later
Enhanced PLC-5 attached to Ethernet Module	1785-L11B, 1785-L20B	Series B: Revision N.1 or later
	1785-L30B, 1785-L40B	Series C: Revision N or later
	1785-L40L, 1785-L60B	Series D: Revision E or later
	1785-L60L, 1785-L80B	Series E: Revision D or later
ControlNet PLC-5 attached to Ethernet Module	1785-L30C15	Series C: Revision N or later
	1785-L40C15	Series D: Revision E or later
	1785-L60C15	Series E: Revision D or later
	1785-L80C15	All revisions
Ethernet Module	1785-Enet	Series B: **Base EtherNet/IP functionality:** All revisions **Full EtherNet/IP Compliance: Revision D or later**

References:

- *Enhanced & Ethernet PLC-5 Series and Enhancement History*, Rockwell Publication G19099
- *ControlNet Processor Phase, Series and Enhancement History*, Rockwell Publication G19102
- *PLC-5 Programmable Controllers System Selection Guide*, Rockwell Publication 1785-SG001A-EN-P - March 2004
- *Ethernet Interface Module Series B, Revision D Product Release Notes*, Rockwell Publication 1785-RN191E-EN-P - December 2002

3.3.2. Messages

The PLC-5 and SLC 5/05 PLCs support the following *PCCC* messages:

Message Type	PCCC Message ID	Maximum Message Size	Maximum Serial Packet Size
SLC Typed Read	162	CLX: 242 SINTs (121 INTs) SLC: 206 SINTs (103 INTs) PLC-5: 240 SINTs (120 INTs)	CLX: 238 SINTs (119 INTs) SLC: 202 SINTs (101 INTs) PLC-5: 236 SINTs (118 INTs)
SLC Typed Write	170	CLX: 220 SINTs (110 INTs) SLC: 206 SINTs 103 INTs) PLC-5: 238 SINTs (119 INTs)	216 SINTs (108 INTs) SLC: 202 SINTs (101 INTs) PLC-5: 234 SINTs (117 INTs)
PLC-5 Typed Read	104	CLX: 234 SINTs (117 INTs) SLC: 252 SINTs (126 INTs) PLC-5: 238 SINTs (119 INTs)	230 SINTS (115 INTs) SLC: 248 SINTs (124 INTs) PLC-5: 234 SINTs (117 INTs)
PLC-5 Typed Write	103	CLX: 226 SINTs (113 INTs) SLC: 226 SINTs (113 INTs) PLC-5: 224 SINTs (112 INTs)	CLX: 222 SINTs (111 INTs) SLC: 222 SINTs (111 INTs) PLC-5: 220 SINTs (110 INTs)

3.3.3. DeviceMaster File Addressing

The following tables display the DeviceMaster file addressing for the PLC-5/SLC messages.

Serial Port Number	Receive Data	Receive Data Produced Sequence Number	Receive Data Consumed Sequence Number	Transmit Data	Transmit Data Sequence Number	Statistics
1	N10:0	N10:128	N10:129	N11:0	N11:128	N12:0
2	N20:0	N20:128	N20:129	N21:0	N21:128	N22:0
3	N30:0	N30:128	N30:129	N31:0	N31:128	N32:0
4	N40:0	N40:128	N40:129	N41:0	N41:128	N42:0

Socket Port Number	Receive Data	Receive Data Produced Sequence Number	Receive Data Consumed Sequence Number	Transmit Data	Transmit data Sequence Number
1	N50:0	N50:128	N50:129	N51:0	N51:128
2	N60:0	N60:128	N60:129	N61:0	N61:128
3	N70:0	N70:128	N70:129	N71:0	N71:128
4	N80:0	N80:128	N80:129	N81:0	N81:128

3.3.4. Receive Data Message

The *Receive Data message* format is similar to the format used in the *Serial Port Data Transfer* Object. However, the data is in 16-bit integer format instead of byte format. The 16-bit integer format is required when connecting to PLC-5 and SLC PLCs.

The following table displays the format of the *Receive Data* message.

Name	Data Type	Data Value(s)	Access Rule
Receive (DeviceMaster to PLC) message data			Read only
Structure of:			
Produced data sequence number	UINT	0-65535 (FFFF hex)	
Data length (in bytes)	UINT	0-222 (SLC) 0-248 (PLC-5)*	
Data array	Array of UINT	0-65535	

Receive messages have the following characteristics:

- It returns all data in UINTs.
- The data length field indicates the number of valid bytes contained in the message.
- The message received from the PLC determines the actual length of the message returned to the PLC. (This is often greater than the length of the actual Receive Data Message.)
- All unused bytes in a message returned to the PLC are filled with zeroes.
- The default order of the bytes is Least Significant Byte First. However, you can select the *(PLC-5/SLC) Rx MS Byte First* option in the web page to return bytes by **Most Significant Byte First**. For more information, see *(PLC-5/SLC) Rx MS Byte First* under *4.3.3. Serial | Serial Settings | SerialPacket Identification* on Page 97.
- The DeviceMaster supports serial packets of up 1518 bytes and socket packets up to 2048 bytes in the *Write-To-File* and *Write-to-File-Synced receive* methods.
- For large received data packets:
  - The data is automatically placed in sequential files.
  - The files must be 256 integers in size with the exception of the last file. The last file may be shorter than 256 integers as long as the total length of all files in the sequence is sufficient to hold the largest receive packet, plus two integers for the sequence number and length parameters.
  - All data has been transferred to the PLC when the sequence number is updated.

3.3.5. Transmit Data Message

The *Transmit Data message* format is similar to the format used in the *Serial Port Data Transfer* Object. However, the data is in 16-bit integer format instead of byte format. The 16-bit integer format is required when connecting to PLC-5 and SLC PLCs.

The following table displays the format of the *Transmit Data* message.

Name	Data Type	Data Value(s)	Access Rule
Transmit (PLC to DeviceMaster) message data			Read/Write
Structure of:			
Produced data sequence number	UINT	0-65535 (FFFF hex)	
Data length (in bytes)	UINT	1-202 (SLC) 1-222 (PLC-5)	
Data array	Array of UINT	0-65535	

Transmit messages have the following characteristics:

- It transfers all data in UINTs.
- The data length field indicates the number of valid bytes contained in the message.
- The actual length of a message received from the PLC may contain extra, unused data.
- It ignores all unused bytes in a message.
- The default order of the bytes is **Least Significant Byte First**. However, you can select the *(PLC-5/SLC) Tx MS Byte First* option in the web page to transmit bytes by **Most Significant Byte First**. For more information, see *(PLC-5/SLC) Tx MS Byte First* under *4.3.3. Serial | Serial Settings | SerialPacket Identification* on Page 97.
- A **Get** returns the last successfully transmitted serial/socket packet.

3.3.6. Sequence Number Messages

PLC-5/SLC typed read and typed write messages can read and modify both receive and transmit produced data sequence numbers. These are the same sequence numbers returned to the PLC in the *Receive Data Message* and sent to the DeviceMaster in the *Transmit Data* message. Access to these sequence numbers are provided primarily for initialization purposes at the start of the PLC program when you may want to initialize the sequence numbers on the PLC, DeviceMaster or both.

PLC-5/SLC typed read and write messages can also read and modify the consumed receive sequence number(s). The consumed receive sequence number(s) are used in the *Write-to-File-Synced* communication method.

3.3.7. Retrieve Statistics Message

The data returned from the *Retrieve Statistics* message is identical to the data returned for the *Serial Port Statistics* Object. The *Retrieve Statistics* message formats the data into 32-bit integers and returns data in an array of s just like all data sent to a PLC-5 or SLC PLC. The first contains the least significant word and the second contains the most significant word.

The following table displays the format of the *Retrieve Statistics* message.

Index	Name	Data Type	Data Value(s)	Access Rule
1	Receive Byte Count	UDINT	0=default	Read only
2	Receive Packet Count	UDINT	0=default	Read only
3	Transmit Byte Count	UDINT	0=default	Read only
4	Transmit Packet Count	UDINT	0=default	Read only
5	Dropped Packet to PLC Count	UDINT	0=default	Read only
6	Parity Error Count	UDINT	0=default	Read only
7	Framing Error Count	UDINT	0=default	Read only
8	Overrun Error Count	UDINT	0=default	Read only
9	Received Consumed Sequence Error Count	UDINT	0=default	Read only
10	Duplicate Transmit Sequence Number errors	UDINT	0=default	Read only
11	Unexpected Transmit Sequence Number errors	UDINT	0=default	Read only
12	Dropped Packet to Application Count	UDINT	0=default	Get

The *Retrieve Statistics* messages have the following characteristics.

Retrieve Statistics Message Description	
Receive Byte Count	This attribute counts the number of bytes received on the serial port.
Receive Packet Count	This attribute counts the number of packets received on the serial port.
Transmit Byte Count	This attribute counts the number of bytes transmitted on the serial port.
Transmit Packet Count	This attribute counts the number of packets transmitted on the serial port.
Dropped Packet to PLC Count	This attribute counts the number of dropped receive packets on the serial port intended for the PLC due to: • No STX byte(s) found • No ETX byte(s) found • Time-outs • Too large of packet • Receive buffer queue overflows

Retrieve Statistics Message Description (Continued)	
Parity Error Count	This attribute counts the number of packets with parity errors received on the serial port.
Framing Error Count	This attribute counts the number of packets with framing errors received on the serial port.
Overrun Error Count	This attribute counts the number of packets with overrun type errors received on the serial port.
Received Consumed Sequence Error Count	This attribute counts the number of received consumed sequence number errors. The DeviceMaster only increments this number when all of the following statements are true: • You selected the *Unsolicited - Write-to-Tag-Synced* method of receive communication. • DeviceMaster receives a serial packet. • The Consumed Sequence number is out of sync. (It is not equal to the Produced Sequence number or equal to the Produced Sequence number minus one.)
Duplicate Transmit Sequence Number Error Count	This attribute counts the number of Duplicate Transmit Sequence Number errors. The DeviceMaster increments this number when the following statements are true: • You enabled the Transmit Sequence Number Checking configuration option. See *Transmit Sequence Number Checking* under *4.6.1. Ethernet/IP Settings* on Page 104 for additional information. • DeviceMaster receives a transmit message with a sequence number equal to the previous sequence number. (The DeviceMaster expects this sequence number to be incremented by one from the sequence number in the previous transmit message.)
Unexpected Transmit Sequence Number Error Count	This attribute counts the number of Unexpected Transmit Sequence Number errors.The DeviceMaster increments this number when the following statements are true. • You enabled the T*ransmit Sequence Number Checking* configuration option. See **Transmit Sequence Number Checking** under *4.6.1. Ethernet/IP Settings*. • DeviceMaster receives a transmit message with a sequence number that is not equal to either the previous sequence number or the previous sequence number plus one. (The DeviceMaster increments this sequence number by one with each new transmit message.)
Dropped Packet to Application Count	This attribute counts the number of dropped receive packets on the serial port intended for the application due to: • No STX byte(s) found • No ETX byte(s) found • Time-outs • Too large of packet • Receive buffer queue overflows

3.3.8. Receive Communication Methods

There are three methods of transferring received data to the PLC from the DeviceMaster.

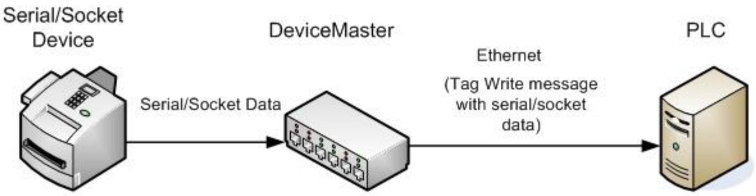
- *Unsolicited - Write to File Receive Method* on Page 83
- *Unsolicited - Write to File Synced Receive Method* on Page 83
- *Polling Receive Method* on Page 84

3.3.8.1. Unsolicited - Write to File Receive Method

***Note:** This is the recommended receive method.*

When a serial/socket packet is received on the DeviceMaster, the data packet is immediately written to a file data location on the PLC. The following diagram shows the data flow.

Unsolicited - Write to File Receive Data Flow



The following restrictions apply to this method:

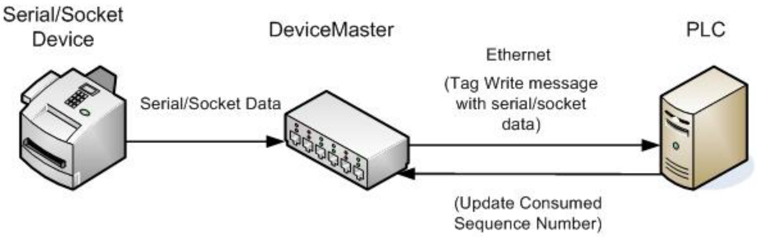
- The **Receive Data File Name** must be the same file name and offset defined to receive data on the PLC.
- The file on the PLC must be of integer type and must be of sufficient size to contain the sequence number, length, and data field associated with the maximum sized received data structure. See *3.3.4. Receive Data Message* on Page 79 for more information.
- New data is indicated with an incremented sequence number.
- The PLC program must be able to process the new data faster than the data can be received. To accomplish this, set the Maximum PLC Update Rate to an interval time that allows your PLC to process the data. The default of 40 milliseconds may or may not need to be increased for your PLC application.

3.3.8.2. Unsolicited - Write to File Synced Receive Method

This method provides a syncing option to allow the PLC to control the data flow by indicating when it is ready for the next serial data packet.

In this method, the serial/socket packet is not written into the file on the PLC until the consumed receive sequence number has been updated by the PLC to match the produced receive data sequence number. Then the data is written into the file data location on the PLC in the same way as the *Write-to-File* method.

Unsolicited - Write to File Synced Receive Data Flow


Serial/Socket Device
DeviceMaster
PLC
Ethernet
(Tag Write message with serial/socket data)
Serial/Socket Data
(Update Consumed Sequence Number)

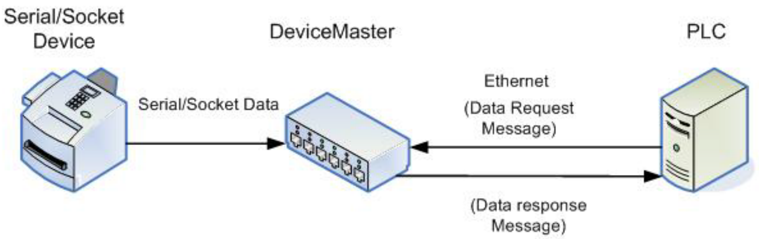
The following restrictions apply to this method:

- The **Receive Data File Name** must be the same file name and offset defined to receive data on the PLC.
- The file on the PLC must be of integer type and must be of sufficient size to contain the sequence number, length, and data field associated with the maximum sized received data structure. See *3.3.4. Receive Data Message* on Page 79 for more information.
- New data is indicated with an incremented sequence number.
- New data is not written to the file on the PLC until the consumed receive sequence number has been incremented to match the last produced receive sequence number.
- While the DeviceMaster queues received serial/socket port data, the PLC program must consume the new data faster than the data can be received on the serial port to ensure the receive buffers on the DeviceMaster do not overflow. (Example: If two packets can be received per second on the serial port, then the consumption rate would need to be at least one packet every 500 msec.)

3.3.8.3. Polling Receive Method

This method provides the *polling* method that allows the PLC to request data on a periodic basis.

Polling Receive Data Flow



In this method, the serial/socket data is returned in the response to the data request message.

The following restrictions apply to this method:

- The file on the PLC must be of integer type and must be of sufficient size to contain the sequence number, length, and data field associated with the maximum sized received data structure. See *3.3.4. Receive Data Message* on Page 79 for more information.
- New data is indicated with an incremented sequence number.
- The same data may be returned more than once. However, the same data packet also returns the same sequence number.
- No data is indicated with a length of zero.
- While the DeviceMaster queues received serial port data, the PLC program must poll for new data faster than the data can be received on the serial port to ensure the receive queues on the DeviceMaster do not overflow. (Example: If two packets can be received per second on the serial port, then the polling rate would need to be at least once every 500 msec.)

Chapter 4. Embedded Configuration Pages

This chapter provides detailed information about the following web pages:

- *Serial* menu, which includes:
  - *Serial Port Overview* page
  - Port Serial Settings - the default *Port Configuration* page
  - Port EtherNet/IP Settings - the *Port EtherNet/IP Configuration* page
  - Port Filtering - the *Port Filtering/Data Extraction Configuration* page
  - Port Application Interface - the *Application TCP Configuration* page
  - Port Class 1 Interface - the *Port Class 1 Interface* page
  - *Class 1 Overview* page
- *Ethernet* menu, which includes:
  - *Ethernet Device Overview* page
  - Device Socket Connection - *Device Interface Configuration* page
  - Device EtherNet/IP Settings - the *Device EtherNet/IP Configuration* page
  - Device Filtering - *Device Filtering/Data Extraction Configuration* page
  - Device Application Interface - the *Device Application TCP Configuration* page
  - *Device Class 1 Interface* page
  - *Class 1 Overview* page
- *Network* menu
  - *Network Configuration* page
  - *Password Configuration* page
  - *Security Settings* page
  - *Key and Certificate Management* page
  - *EtherNet/IP Stack Configuration* page
- *System* menu
  - *Update Firmware* page
  - *Configuration File* page
  - *System Snapshot* page
  - *Restore Defaults* page
  - *Reboot* page

Note: *The Diagnostics menu is discussed in Chapter 5. Diagnostic and Statistics Pages on Page 135.*

The latest EtherNet/IP firmware must be installed before you can configure network or serial/socket port characteristics. For firmware installation and setup information, see the *EtherNet/IP Hardware Installation and Configuration Guide* or the PortVision DX help system.

Use the *EtherNet/IP Interface Configuration Quick Start* to locate configuration procedures for your site and use this chapter as a reference if you need information about specific fields. The *Interface Configuration Quick Start* is intended to provide you with a way to quickly configure.devices such as barcode scanners, RFID readers, and printers. In addition, there is also a section that discusses configuring read/write devices such as some printers and weigh scales.

Note: *ControlLogix PLC environments can optionally change the serial/socket port settings through the ControlLogix PLC using the Serial Port Configuration (3.2.1. Serial Port Configuration Object Definition (70 Hex) on Page 20) or Socket Port Configuration (3.2.4. Socket Port Configuration Object*

Definition (73 Hex) on Page 37) objects.

4.1. Configuration Overview

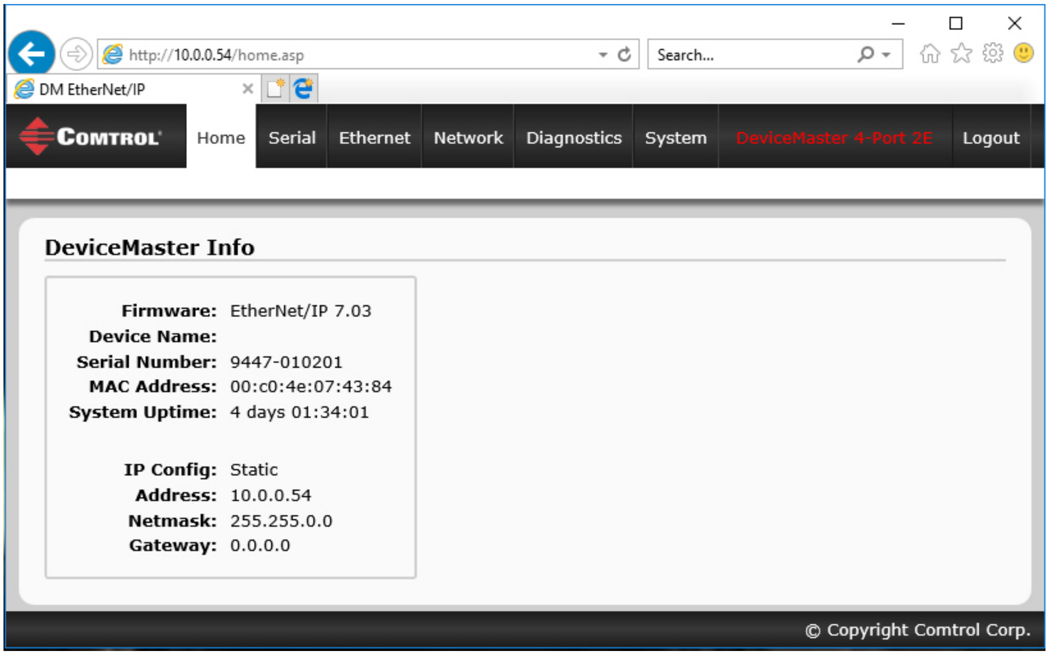
The following overview shows how to access the DeviceMaster embedded web page and configure serial and Ethernet device interfaces.

If you have not configured the network information into the DeviceMaster during initial setup, you must configure the network information before configuring serial/socket port characteristics. See the *EtherNet/IP Hardware Installation and Configuration Guide* or the PortVision DX help system for help configuring the network settings.

1. From PortVision DX, highlight the DeviceMaster that you want to configure and select **Webpage**.

   Note: *Optionally, enter the IP address of the device in the* ***Address*** *box of your web browser.*

2. Select the appropriate procedure for your environment.



DeviceMaster Info

Firmware: EtherNet/IP 7.03
Device Name:
Serial Number: 9447-010201
MAC Address: 00:c0:4e:07:43:84
System Uptime: 4 days 01:34:01

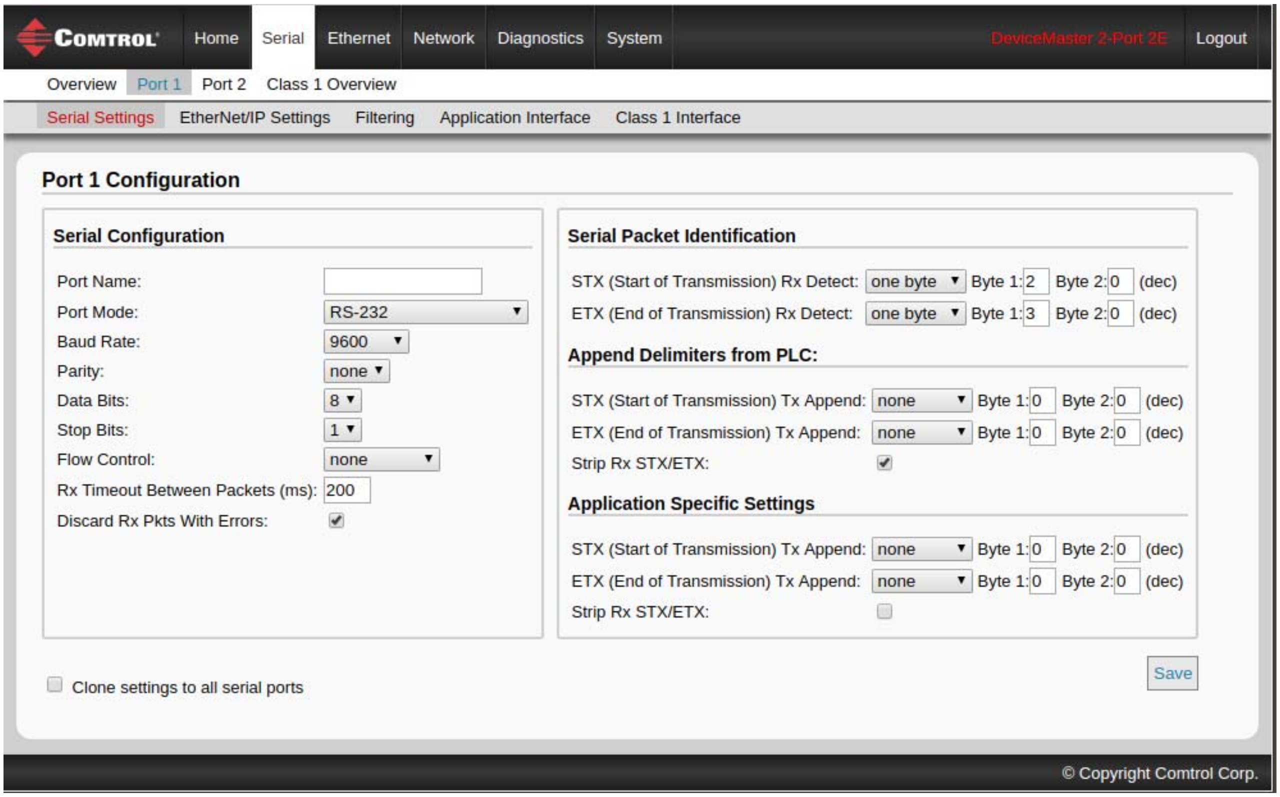
IP Config: Static
Address: 10.0.0.54
Netmask: 255.255.0.0
Gateway: 0.0.0.0

© Copyright Comtrol Corp.

4.1.1. Serial Device - Configuration Overview

a. Click the **Serial** menu.

b. Click the port number that you want to configure and the *Port Configuration* page appears.



Port 1 Configuration

Serial Configuration

Port Name:
Port Mode: RS-232
Baud Rate: 9600
Parity: none
Data Bits: 8
Stop Bits: 1
Flow Control: none
Rx Timeout Between Packets (ms): 200
Discard Rx Pkts With Errors: ☑

Serial Packet Identification

STX (Start of Transmission) Rx Detect: one byte Byte 1: 2 Byte 2: 0 (dec)
ETX (End of Transmission) Rx Detect: one byte Byte 1: 3 Byte 2: 0 (dec)

Append Delimiters from PLC:

STX (Start of Transmission) Tx Append: none Byte 1: 0 Byte 2: 0 (dec)
ETX (End of Transmission) Tx Append: none Byte 1: 0 Byte 2: 0 (dec)
Strip Rx STX/ETX: ☑

Application Specific Settings

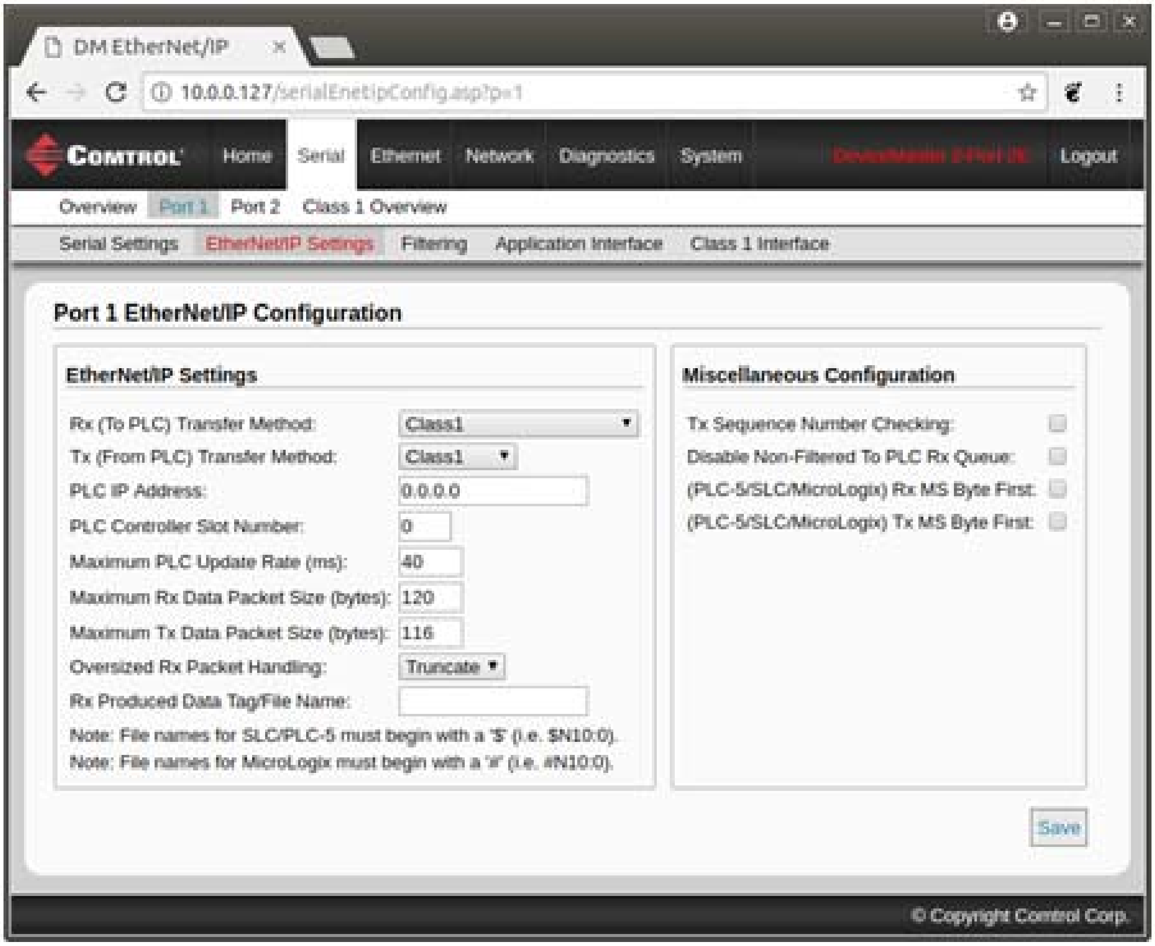
STX (Start of Transmission) Tx Append: none Byte 1: 0 Byte 2: 0 (dec)
ETX (End of Transmission) Tx Append: none Byte 1: 0 Byte 2: 0 (dec)
Strip Rx STX/ETX: □

□ Clone settings to all serial ports

c. Change the serial port configuration properties (Page 95) as required for your site.

d. Click the **Save** button.

3. Click the **Ethernet/IP Settings** option.
   a. Change the socket port configuration properties (Page 104) as required for your site.



Port 1 EtherNet/IP Configuration

EtherNet/IP Settings

Rx (To PLC) Transfer Method:	Class1 ▾
Tx (From PLC) Transfer Method:	Class1 ▾
PLC IP Address:	0.0.0.0
PLC Controller Slot Number:	0
Maximum PLC Update Rate (ms):	40
Maximum Rx Data Packet Size (bytes):	120
Maximum Tx Data Packet Size (bytes):	116
Oversized Rx Packet Handling:	Truncate ▾
Rx Produced Data Tag/File Name:	

Note: File names for SLC/PLC-5 must begin with a '$' (i.e. $N10:0).
Note: File names for MicroLogix must begin with a '#' (i.e. #N10:0).

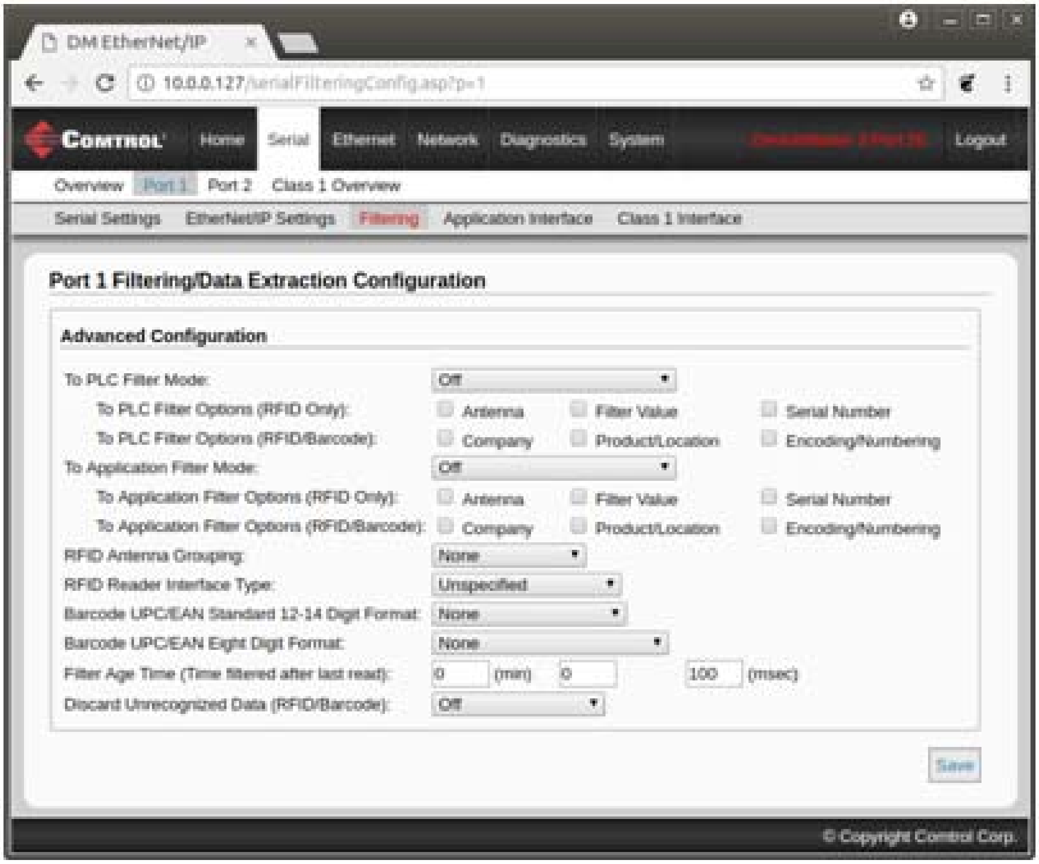
Miscellaneous Configuration

Tx Sequence Number Checking:	□
Disable Non-Filtered To PLC Rx Queue:	□
(PLC-5/SLC/MicroLogix) Rx MS Byte First:	□
(PLC-5/SLC/MicroLogix) Tx MS Byte First:	□

   b. Click the **Save** button.

4. If required for your site, click **Filtering**.

   a. Change the filtering configuration (Page 107) to meet your needs.



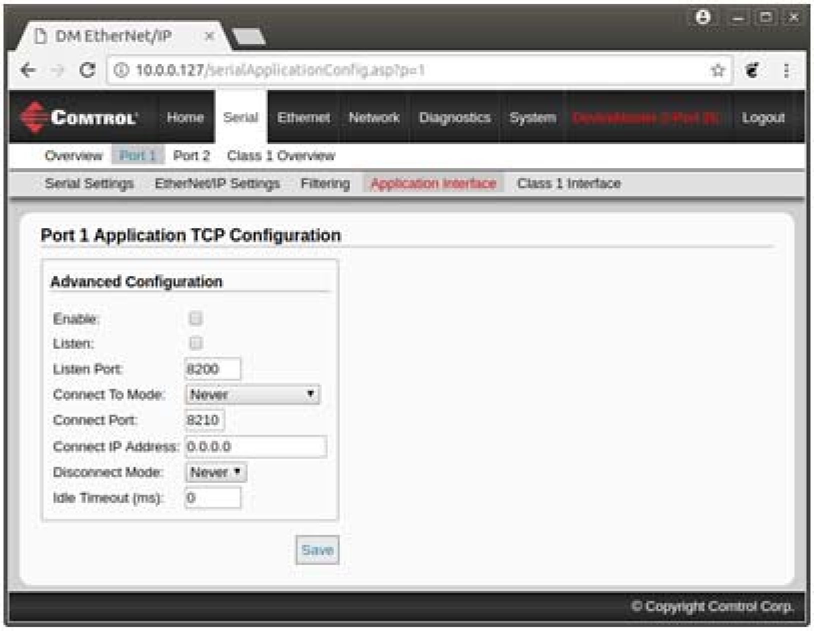
Port 1 Filtering/Data Extraction Configuration

Advanced Configuration

To PLC Filter Mode:	Off		
To PLC Filter Options (RFID Only):	□ Antenna	□ Filter Value	□ Serial Number
To PLC Filter Options (RFID/Barcode):	□ Company	□ Product/Location	□ Encoding/Numbering
To Application Filter Mode:	Off		
To Application Filter Options (RFID Only):	□ Antenna	□ Filter Value	□ Serial Number
To Application Filter Options (RFID/Barcode):	□ Company	□ Product/Location	□ Encoding/Numbering
RFID Antenna Grouping:	None		
RFID Reader Interface Type:	Unspecified		
Barcode UPC/EAN Standard 12-14 Digit Format:	None		
Barcode UPC/EAN Eight Digit Format:	None		
Filter Age Time (Time filtered after last read):	0 (min)	0	100 (msec)
Discard Unrecognized Data (RFID/Barcode):	Off		

   b. Click the **Save** button when you have completed the changes.

5. If required for your site, click **Application Interface**.
   a. Change the configuration (Page 111) to meet your needs.



Port 1 Application TCP Configuration

Advanced Configuration

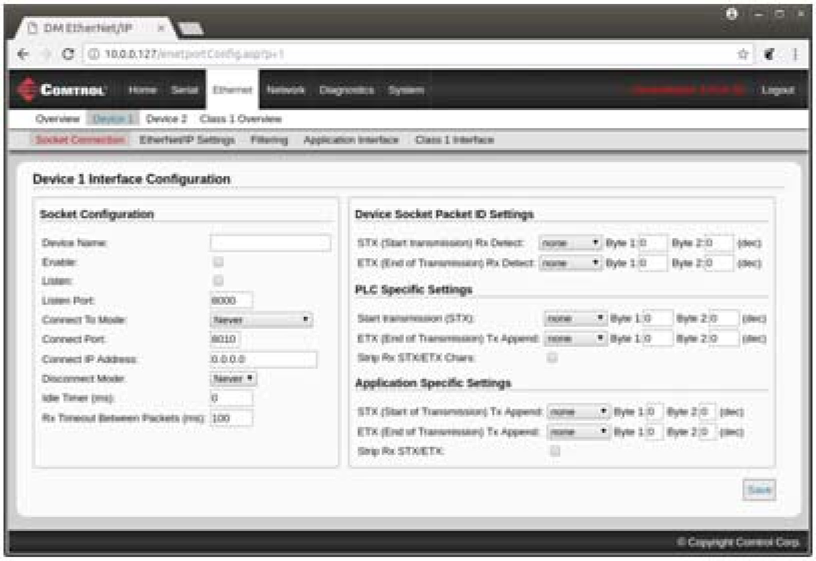
Enable: □
Listen: □
Listen Port: 8200
Connect To Mode: Never
Connect Port: 8210
Connect IP Address: 0.0.0.0
Disconnect Mode: Never
Idle Timeout (ms): 0

   b. Click the **Save** button when you have completed the changes.

4.1.2. Ethernet Device - Configuration Overview

Use the following overview to configure an Ethernet device.

1. Click the **Ethernet** menu.
2. Click the appropriate **Device** number to access the *Device Configuration* page for that port.
   a. Change the socket configuration properties (Page 100) as required for you site.



Device 1 Interface Configuration

Socket Configuration

Device Name:
Enable: □
Listen: □
Listen Port: 8000
Connect To Mode: Never
Connect Port: 8010
Connect IP Address: 0.0.0.0
Disconnect Mode: Never
Idle Timer (ms): 0
Rx Timeout Between Packets (ms): 100

Device Socket Packet ID Settings

STX (Start transmission) Rx Detect: none Byte 1: 0 Byte 2: 0 (dec)
ETX (End of Transmission) Rx Detect: none Byte 1: 0 Byte 2: 0 (dec)

PLC Specific Settings

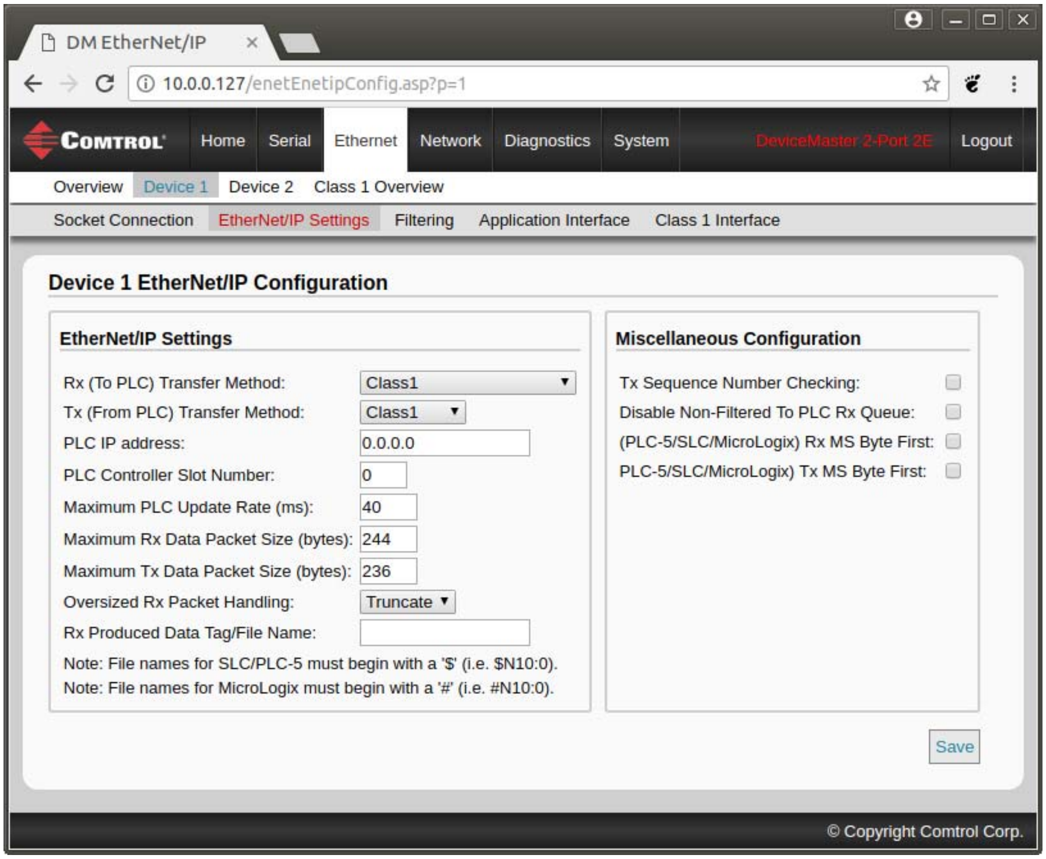
Start transmission (STX): none Byte 1: 0 Byte 2: 0 (dec)
ETX (End of Transmission) Tx Append: none Byte 1: 0 Byte 2: 0 (dec)
Strip Rx STX/ETX Chars: □

Application Specific Settings

STX (Start of Transmission) Tx Append: none Byte 1: 0 Byte 2: 0 (dec)
ETX (End of Transmission) Tx Append: none Byte 1: 0 Byte 2: 0 (dec)
Strip Rx STX/ETX: □

   b. Click the **Save** button after completing the changes.

3. Click the **EtherNet/IP Settings** option.



Device 1 EtherNet/IP Configuration

EtherNet/IP Settings

Rx (To PLC) Transfer Method:	Class1 ▾
Tx (From PLC) Transfer Method:	Class1 ▾
PLC IP address:	0.0.0.0
PLC Controller Slot Number:	0
Maximum PLC Update Rate (ms):	40
Maximum Rx Data Packet Size (bytes):	244
Maximum Tx Data Packet Size (bytes):	236
Oversized Rx Packet Handling:	Truncate ▾
Rx Produced Data Tag/File Name:	

Note: File names for SLC/PLC-5 must begin with a '$' (i.e. $N10:0).
Note: File names for MicroLogix must begin with a '#' (i.e. #N10:0).

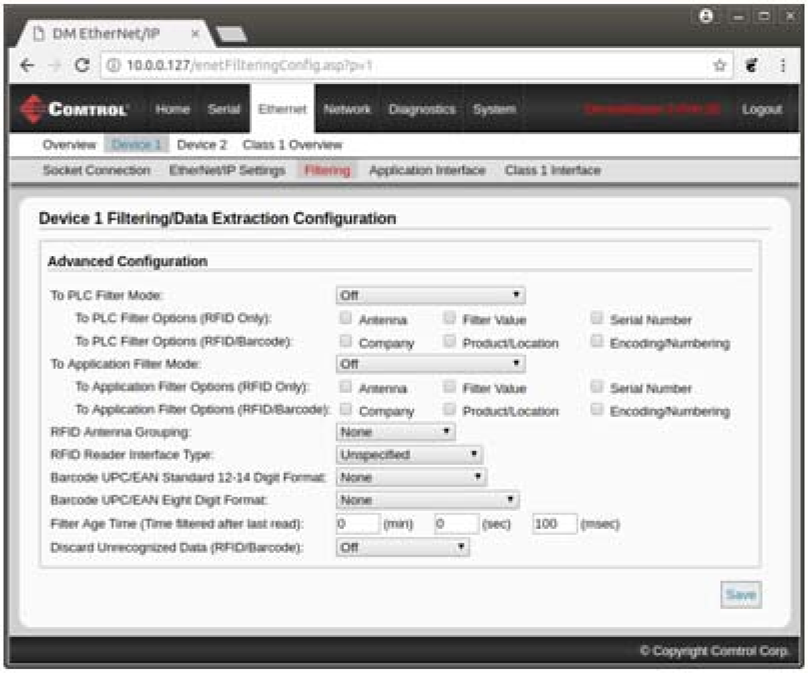
Miscellaneous Configuration

Tx Sequence Number Checking:	□
Disable Non-Filtered To PLC Rx Queue:	□
(PLC-5/SLC/MicroLogix) Rx MS Byte First:	□
PLC-5/SLC/MicroLogix) Tx MS Byte First:	□

a. Change the EtherNet/IP settings (Page 104) as required for your site.

b. Click the **Save** button after completing the changes.

4. If required for your site, click the **Filtering** option.
   a. Change the parameters (Page 107) to meet your needs.

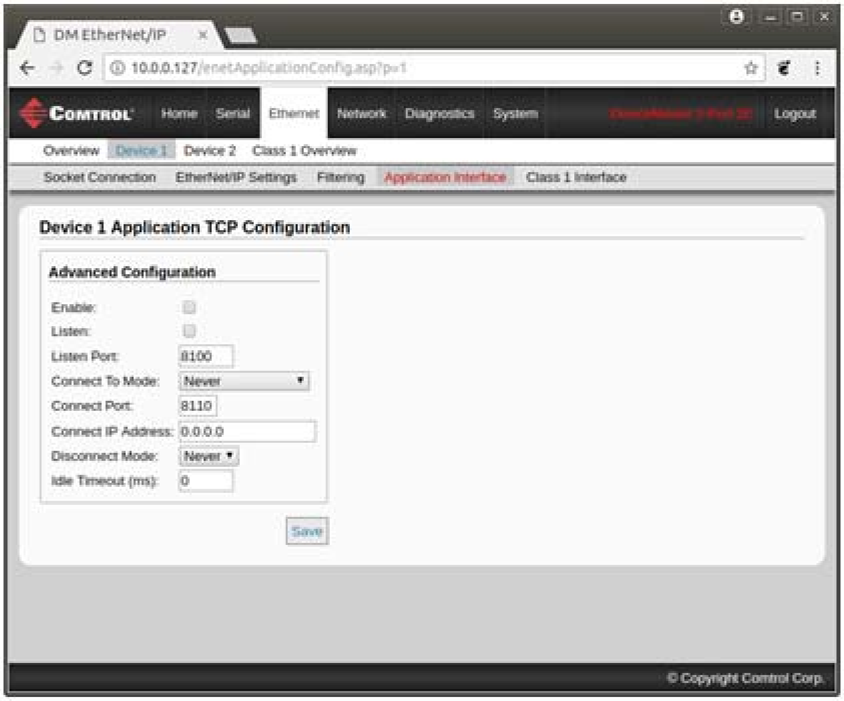


Device 1 Filtering/Data Extraction Configuration

Advanced Configuration

To PLC Filter Mode:	Off		
To PLC Filter Options (RFID Only):	□ Antenna	□ Filter Value	□ Serial Number
To PLC Filter Options (RFID/Barcode):	□ Company	□ Product/Location	□ Encoding/Numbering
To Application Filter Mode:	Off		
To Application Filter Options (RFID Only):	□ Antenna	□ Filter Value	□ Serial Number
To Application Filter Options (RFID/Barcode):	□ Company	□ Product/Location	□ Encoding/Numbering
RFID Antenna Grouping:	None		
RFID Reader Interface Type:	Unspecified		
Barcode UPC/EAN Standard 12-14 Digit Format:	None		
Barcode UPC/EAN Eight Digit Format:	None		
Filter Age Time (Time filtered after last read):	0 (min)	0 (sec)	100 (msec)
Discard Unrecognized Data (RFID/Barcode):	Off		

   b. Click the **Save** button.
5. If required, click the **Application Interface** option.
   a. Make the necessary changes (Page 111) for your site.



Device 1 Application TCP Configuration

Advanced Configuration

Enable:	□
Listen:	□
Listen Port:	8100
Connect To Mode:	Never
Connect Port:	8110
Connect IP Address:	0.0.0.0
Disconnect Mode:	Never
Idle Timeout (ms):	0

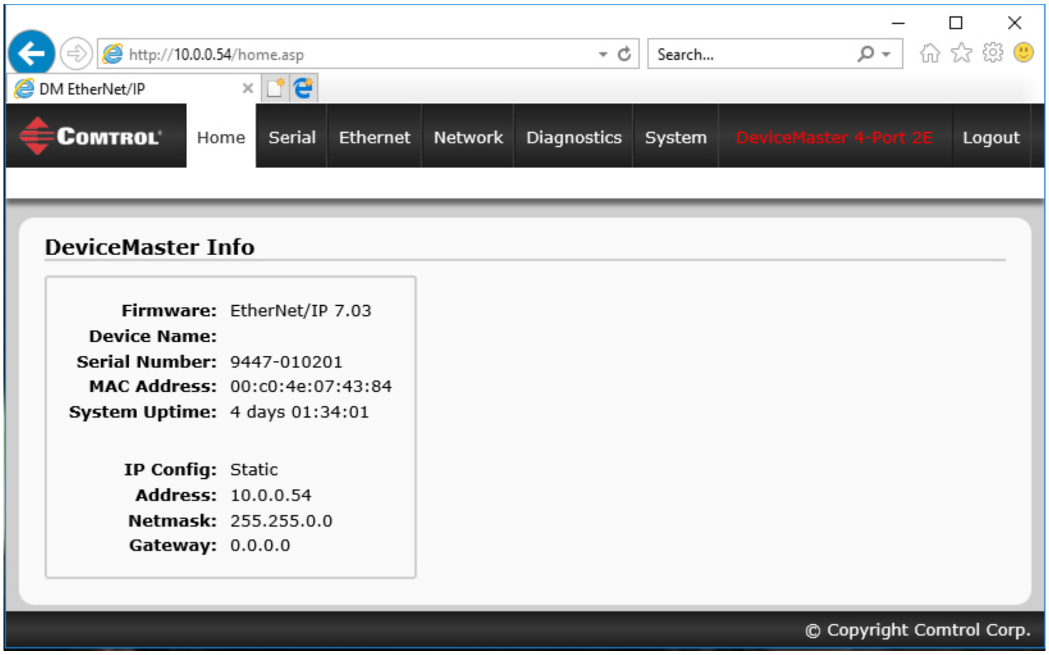
   b. Click the **Save** button.
6. Repeat the previous steps for each serial port and Ethernet device.

7. Use the appropriate procedure for your environment to complete the DeviceMaster installation.
   - **ControlLogix PLC**: *6.4. ControlLogix PLC Programming Example Instructions* on Page 153 describes how to use RSLogix 5000 to configure and run the DeviceMaster.
   - **SLC or MicroLogix PLC**: *6.5. SLC or MicroLogix PLC Programming Example Instructions* on Page 166 describes how to use RSLogix 500 to configure and run the DeviceMaster.
   - **PLC-5 PLC**: *6.6. PLC-5 PLC Programming Example Instructions* on Page 193 describes how to use RSLogix 5 to configure and run the DeviceMaster.

4.2. Embedded Web Pages Overview

The embedded web pages are the easiest way to configure serial and Ethernet device port settings for a DeviceMaster.

Access the main DeviceMaster web page from PortVision DX or enter the IP address of the DeviceMaster in the **Address** box of your web browser.



DeviceMaster Info

Firmware: EtherNet/IP 7.03
Device Name:
Serial Number: 9447-010201
MAC Address: 00:c0:4e:07:43:84
System Uptime: 4 days 01:34:01

IP Config: Static
Address: 10.0.0.54
Netmask: 255.255.0.0
Gateway: 0.0.0.0

© Copyright Comtrol Corp.

The *Home* page displays the DeviceMaster information including the firmware version, device name, serial number, MAC address and current network configuration..

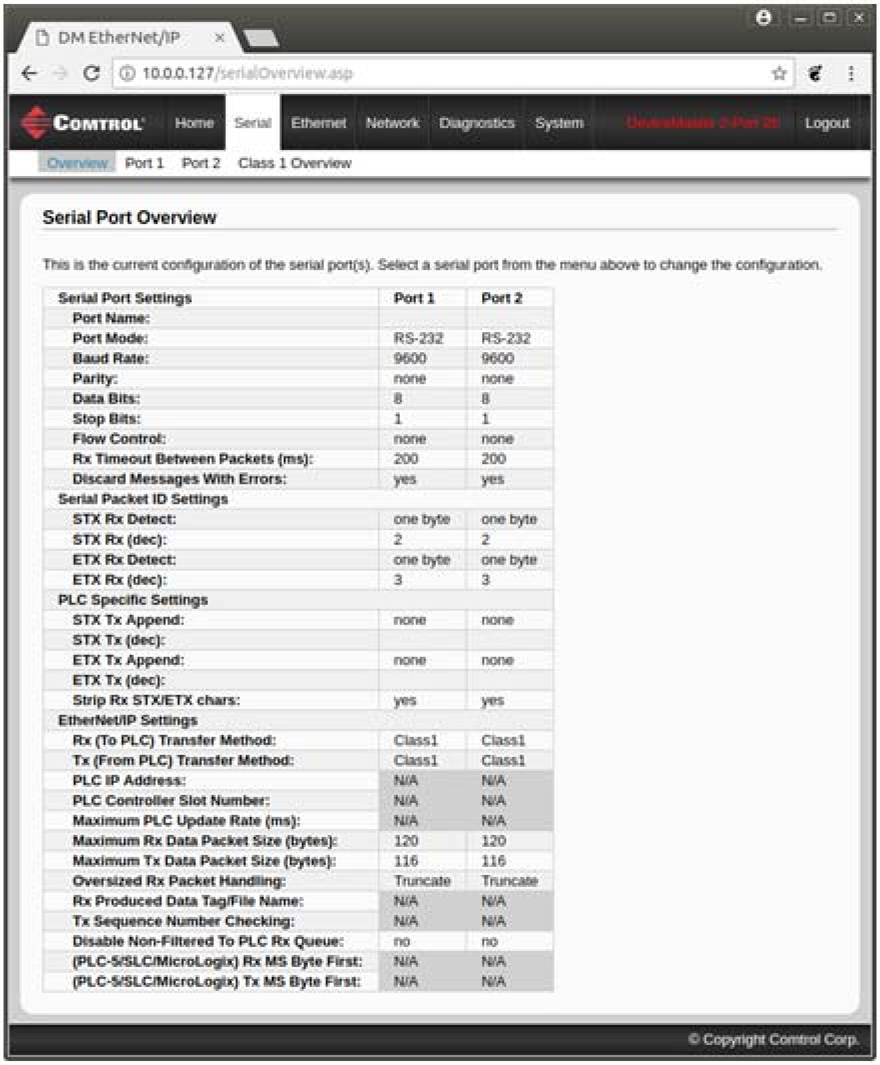
Home Page	
Firmware Version	EtherNet/IP firmware version currently running on the DeviceMaster.
Device Name	This is the device name of the DeviceMaster, which can be configured in the *Network Configuration* page.
Serial Number	DeviceMaster serial number.
MAC Address	The six byte MAC address of the DeviceMaster.
System Uptime	The time that has elapsed since the DeviceMaster has powered on. The format is ***n*** days and ***hh:mm:ss***.
IP Config	Type of IP configuration currently in use (static or DHCP).

Home Page (Continued)	
IP Address, IP Netmask, and IP Gateway	IP address, netmask, and gateway configured in the DeviceMaster.

4.3. Serial Device Configuration

Use these steps to change the settings for a port.

1. Click the **Serial** menu, which opens the *Serial Overview* page.



Serial Port Overview

This is the current configuration of the serial port(s). Select a serial port from the menu above to change the configuration.

Serial Port Settings	Port 1	Port 2
Port Name:		
Port Mode:	RS-232	RS-232
Baud Rate:	9600	9600
Parity:	none	none
Data Bits:	8	8
Stop Bits:	1	1
Flow Control:	none	none
Rx Timeout Between Packets (ms):	200	200
Discard Messages With Errors:	yes	yes
Serial Packet ID Settings		
STX Rx Detect:	one byte	one byte
STX Rx (dec):	2	2
ETX Rx Detect:	one byte	one byte
ETX Rx (dec):	3	3
PLC Specific Settings		
STX Tx Append:	none	none
STX Tx (dec):		
ETX Tx Append:	none	none
ETX Tx (dec):		
Strip Rx STX/ETX chars:	yes	yes
EtherNet/IP Settings		
Rx (To PLC) Transfer Method:	Class1	Class1
Tx (From PLC) Transfer Method:	Class1	Class1
PLC IP Address:	N/A	N/A
PLC Controller Slot Number:	N/A	N/A
Maximum PLC Update Rate (ms):	N/A	N/A
Maximum Rx Data Packet Size (bytes):	120	120
Maximum Tx Data Packet Size (bytes):	116	116
Oversized Rx Packet Handling:	Truncate	Truncate
Rx Produced Data Tag/File Name:	N/A	N/A
Tx Sequence Number Checking:	N/A	N/A
Disable Non-Filtered To PLC Rx Queue:	no	no
(PLC-5/SLC/MicroLogix) Rx MS Byte First:	N/A	N/A
(PLC-5/SLC/MicroLogix) Tx MS Byte First:	N/A	N/A

2. Click the **Port #** that you want to configure.

See *4.3.1. Serial Port Configuration* on Page 95 to locate information for each setting area.

4.3.1. Serial Port Configuration

Use the **Serial** menus to change a serial port's configuration parameters.

You can use the following procedure to access the *Serial | Serial Settings Port Configuration* page:

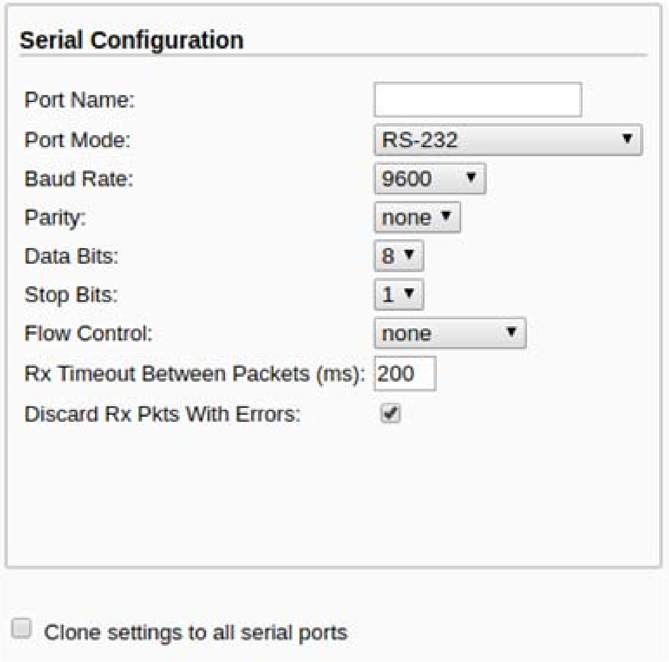
1. Click the **Serial** menu.
2. Click the appropriate port number menu option (for example, **Port 1**).

Selecting the Port # defaults to the *Serial Settings* page for that port. The next subsections discuss the *Serial Port* and *Serial Packet ID Settings* areas on the *Serial Settings* (Port Configuration) page. The remainder of the configuration pages are discussed in the following subsections, which are located under the *4.6. Common Configuration Areas (Serial or Ethernet Device)* section:

- *4.6.1. Ethernet/IP Settings* on Page 104
- *4.6.2. Filtering/Data Extraction Configuration* on Page 107
- *4.6.3. Application Interface | Application TCP Configuration* on Page 111
- *4.7. EtherNet/IP Class 1 Interface Pages* on Page 112

4.3.2. Serial Settings | Port | Serial Configuration

Use the *Serial Configuration* area of the *Serial Settings (Port Configuration)* page to configure serial port characteristics for the device that you plan on connecting to the port.



Serial Configuration

Port Name:

Port Mode: RS-232

Baud Rate: 9600

Parity: none

Data Bits: 8

Stop Bits: 1

Flow Control: none

Rx Timeout Between Packets (ms): 200

Discard Rx Pkts With Errors: ☑

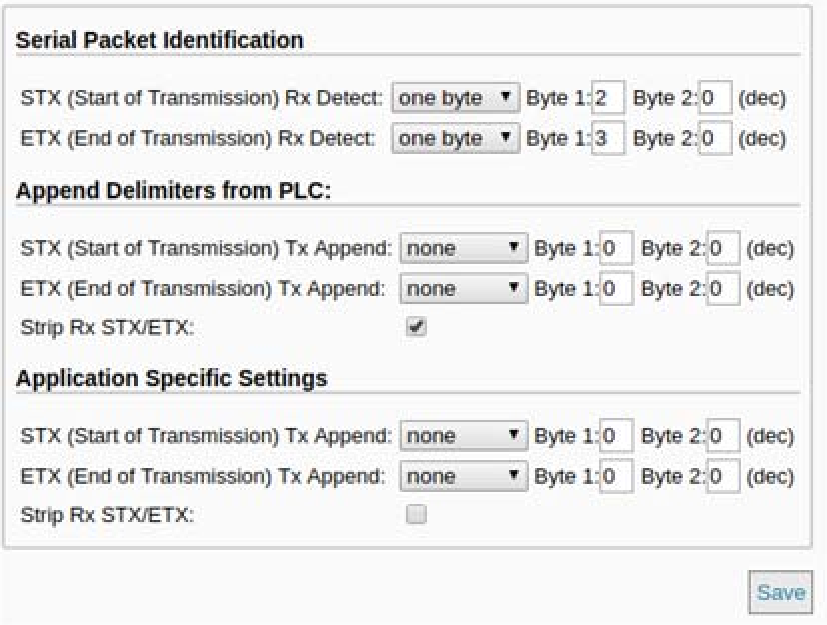
☐ Clone settings to all serial ports

Serial Settings | Port | Serial Configuration	
Port Name	Up to 80 character ASCII string. A user definable string used to describe the serial port. Valid characters include a-z, A-Z, 0-9, underscores, spaces and dashes. All other characters are discarded. The default name is blank.
Port Mode	Select the communications mode for the serial device that you are connecting to the port. The available modes are RS-232, RS-422, and RS-485.
Baud Rate	Select a baud rate from the list. The baud rate that you select determines how fast information is transferred through a port.

Serial Settings | Port | Serial Configuration (Continued)	
Parity	Select a method for error checking. • **None** - When the parity is set to none, there is no parity bit, and DeviceMaster does not perform parity checking. • **Odd** - Indicates that the sum of all the 1-bits in the byte plus the parity bit must be odd. When the total is odd, the parity bit is set to zero, when it is even, the parity bit is set to one. • **Even** - When the sum of all the 1-bits is even, the parity bit must be set to zero; when it is odd, the parity bit must be set to one.
Data Bits	Select the number of bits that make up the data. Choose from 5, 6, 7 or 8-bits.
Stop Bits	Select the number of bits to mark the end of data transmission.
Flow Control	Specifies the ability to start and stop the flow of data without the loss of bytes. Select a method for controlling the flow of data from the following list: • **None** - Indicates flow control is not in affect. • **RTS/CTS** - Request To Send (RTS) tells the receiving device that the sending device has data that is ready to send and Clear To Send (CTS) indicates the device is ready to accept data. • **XON/XOFF** - When selected, applies the standard method of controlling data flow between two modems. • **Half Duplex** - Transmits data in half-duplex mode.
DTR Mode	Select the state of Data Terminal Ready (DTR). • **on** - Enables DTR. • **off** - Disables DTR. • **WhenEnabled** - Select this option when enabling the serial port through the PLC.
Rx Timeout Between Packets (ms)	Specifies the following information, once the start of a packet is received: • How long the DeviceMaster should wait (in milliseconds) before timing-out, if the **ETX Rx Detect** length is one byte or two bytes and the ETX byte(s) are not received. • The time to wait in milliseconds between serial packets if the **ETX Rx Detect** length is set to **none**.
Discard Rx Pkts with Errors	By default, this box is checked and the DeviceMaster discards serial packets with errors. Clear the check box when you need to receive a serial packet with errors to troubleshoot an issue.
Clone settings to all ports	Selecting this option before selecting **Save** applies these settings to all of the serial ports.

4.3.3. Serial | Serial Settings | SerialPacket Identification

Use the *Serial Packet Identification* area of the *Serial Settings* page to configure the raw data serial port packet identification (ID) settings for a serial port.



Serial Packet Identification

STX (Start of Transmission) Rx Detect: one byte ▼ Byte 1: 2 Byte 2: 0 (dec)
ETX (End of Transmission) Rx Detect: one byte ▼ Byte 1: 3 Byte 2: 0 (dec)

Append Delimiters from PLC:

STX (Start of Transmission) Tx Append: none ▼ Byte 1: 0 Byte 2: 0 (dec)
ETX (End of Transmission) Tx Append: none ▼ Byte 1: 0 Byte 2: 0 (dec)
Strip Rx STX/ETX: ☑

Application Specific Settings

STX (Start of Transmission) Tx Append: none ▼ Byte 1: 0 Byte 2: 0 (dec)
ETX (End of Transmission) Tx Append: none ▼ Byte 1: 0 Byte 2: 0 (dec)
Strip Rx STX/ETX: □

For more information on serial packet ID settings, see *3.2.1. Serial Port Configuration Object Definition (70 Hex)* on Page 20.

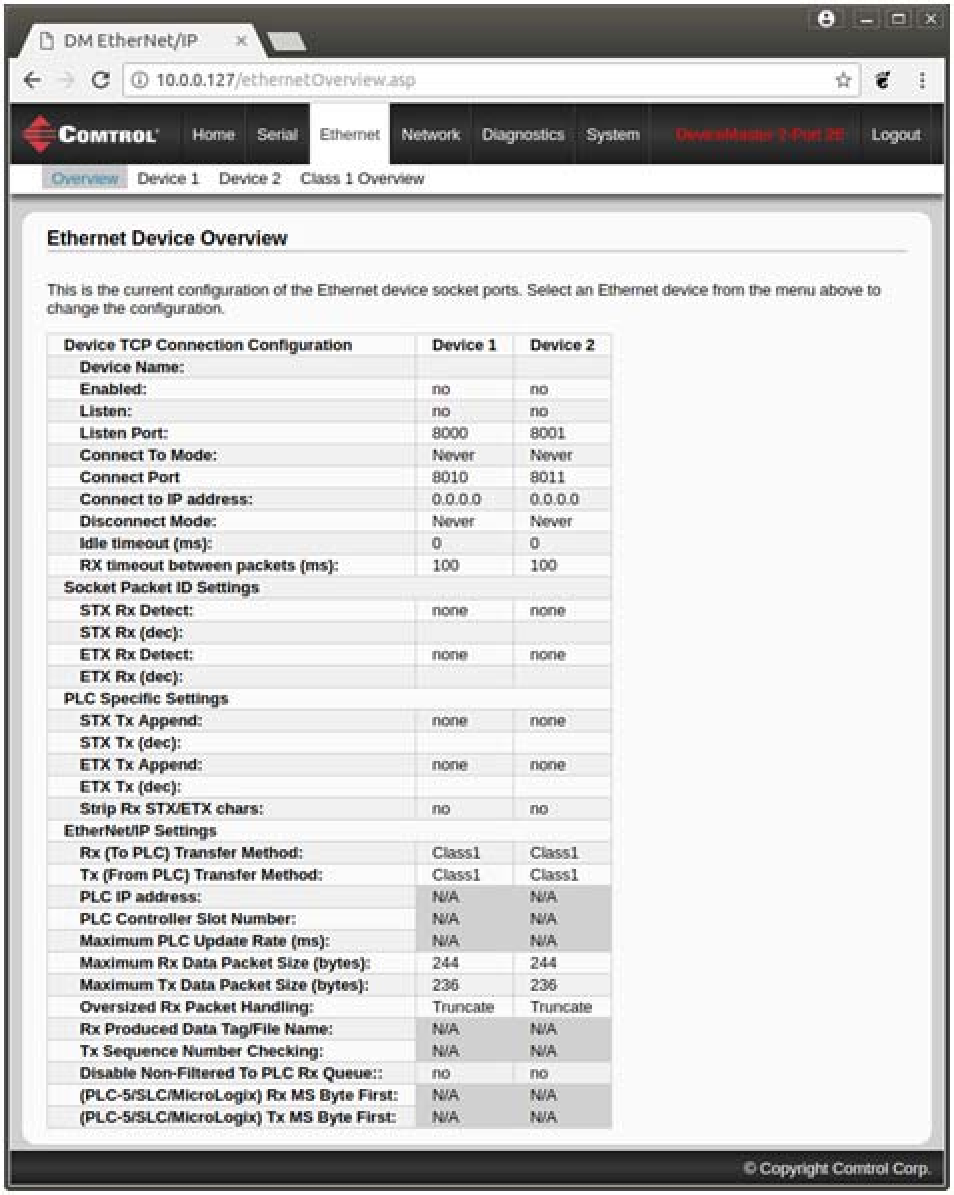
<table>
<tr><th colspan="2">Serial | Port | Serial Settings |Serial Packet Identification</th></tr>
<tr><td colspan="2"></td></tr>
<tr><td>STX (Start of Transmission) Rx Detect</td><td>When enabled, the DeviceMaster detects an STX (start of transmission) byte sequence which is configured as <b>one byte</b> or <b>two bytes</b> when it receives a serial packet.
The length indicates the number of STX bytes, valid values for length are:
• <b>none</b> - Disables this function and the DeviceMaster accepts the first byte received after the last ETX byte(s) as the start of the next data packet.
• <b>one byte</b> - Scans serial data for one STX byte and when the DeviceMaster finds an STX byte it collects the data. If the first byte is not the STX byte, it discards the byte. The DeviceMaster continues to discard the bytes until it finds an STX byte.
• <b>two bytes</b> - Scans serial data for two STX bytes and when the DeviceMaster finds two STX bytes it collects the data. If the STX bytes cannot be found, it discards the bytes. The DeviceMaster continues to discard the bytes until it finds the two STX bytes.
<b>Byte 1</b> - Specifies the character that represents the first STX byte. The DeviceMaster looks for this character in the first STX byte, if the length is <b>one byte</b> or <b>two bytes</b>. You can specify a value between 0 and 255 in decimal format.
<b>Byte 2</b> - Specifies the character that represents the second STX byte. The DeviceMaster looks for this character in the second STX byte, only if the length is <b>two bytes</b>. You can specify a value between 0 and 255 in decimal format.</td></tr>
</table>

Serial | Port | Serial Settings |Serial Packet Identification (Continued)	
ETX (End of Transmission) Rx Detect	When enabled, the DeviceMaster detects an ETX (end of transmission) byte sequence that is configured as **one byte** or **two bytes** marking the end of the serial packet. The length indicates the number of ETX bytes, valid values for length are: • **none** - Disables this function and the DeviceMaster uses the *Rx Timeout Between Packets* to indicate the end of data packet. • **one byte** - Scans serial data for one ETX byte and when the DeviceMaster finds the ETX byte, it identifies the data as a serial packet. • **two bytes** - Scans serial data for two ETX bytes and when the DeviceMaster finds the ETX bytes, it identifies the data as a serial packet. **Byte 1** - Specifies the character to scan for in the first ETX byte, if the length is **one byte** or **two bytes**. You can specify a value between 0 and 255 in decimal format. **Byte 2** - Specifies the character to scan for in the second ETX byte, if the length is **two bytes**. You can specify a value between 0 and 255 in decimal format.
Append Delimiters from PLC and Application Specific Settings Options	
STX Tx Append	When enabled, the DeviceMaster appends an STX (start of transmission) byte sequence which is configured as **one byte** or **two bytes** to the beginning of the serial packet before it is sent. The length indicates the number of STX bytes, values for length are: • **none** - Disables this function. • **one byte** - Inserts one STX byte before the data. • **two bytes** - Inserts two STX bytes before the data. **Byte 1** - Specifies the transmit character associated with the first STX byte, if the length is **one byte** or **two bytes**. You can specify a value between 0 and 255 in decimal format. **Byte 2** - Specifies the transmit character associated with the second STX byte, if the length is **two bytes**. You can specify a value between 0 and 255 in decimal format.
ETX Tx Append	When enabled, the DeviceMaster appends an ETX (end of transmission) byte sequence which is configured as **one byte** or **two bytes** to the end of the serial packet before it is sent. The length indicates the number of ETX bytes, valid values for length are: • **none** - Disables this function. • **one byte** - Inserts one ETX byte at the end of the data. • **two bytes** - Inserts two ETX bytes at the end of the data. **Byte 1** - Specifies the transmit character associated with the first ETX byte, if the length is set to **one byte** or **two bytes**. You can specify a value between 0 and 255 in decimal format. **Byte 2** - Specifies the transmit character associated with the second ETX byte, if the length is **two bytes**. You can specify a value between 0 and 255 in decimal format.
Strip Rx STX/ ETX	When you select this check box, the DeviceMaster strips STX/ETX characters from received serial packets. Clear the check box when you do not want the DeviceMaster to strip STX/ETX characters from received serial packets. Serial Packets sent from the PLC or application to the DeviceMaster (over Ethernet), and then sent out the serial port, are not checked for STX/ETX. No STX/ETX character stripping occurs in these serial packets, and framing/parity/ overrun error checking does not apply.

4.4. Ethernet Device Configuration

Change the configuration for an Ethernet device using the following steps,

1. Click the **Ethernet** menu, which displays the *Ethernet Device Overview* page.



Ethernet Device Overview

This is the current configuration of the Ethernet device socket ports. Select an Ethernet device from the menu above to change the configuration.

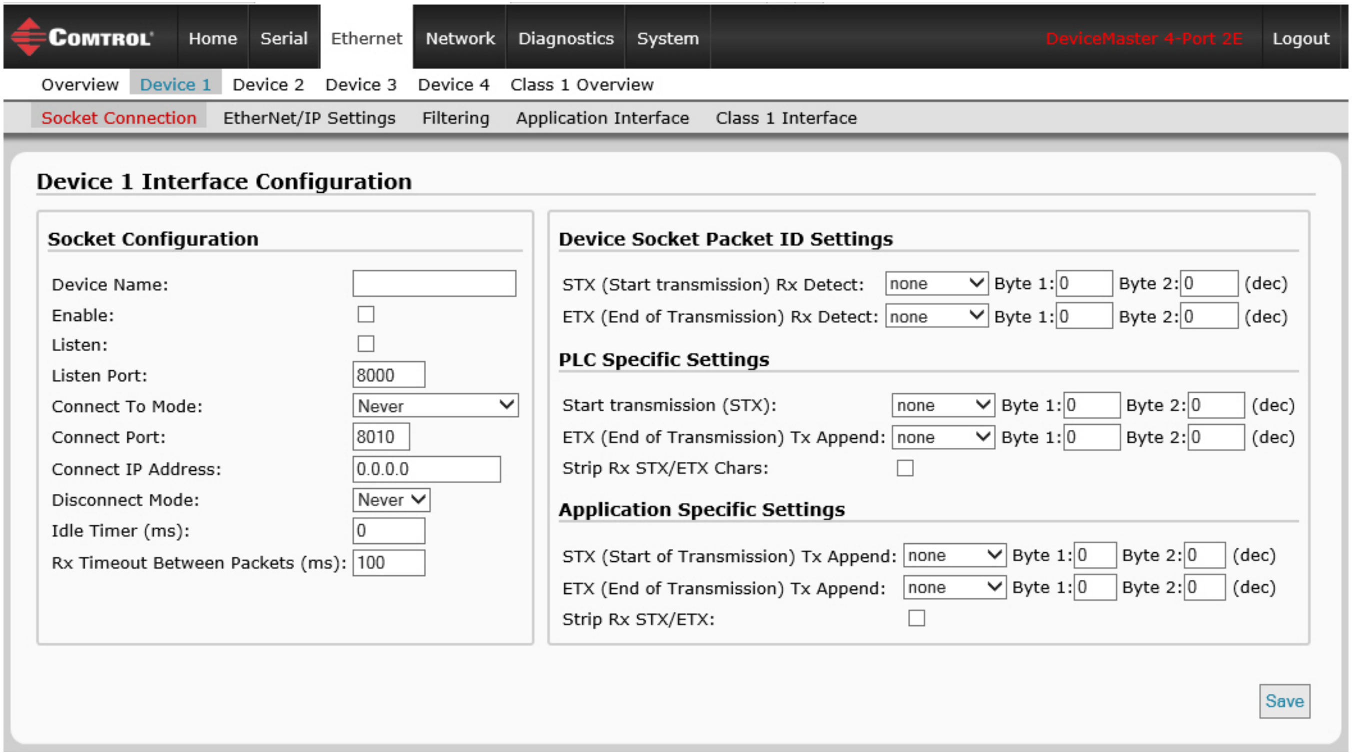
Device TCP Connection Configuration	Device 1	Device 2
Device Name:		
Enabled:	no	no
Listen:	no	no
Listen Port:	8000	8001
Connect To Mode:	Never	Never
Connect Port	8010	8011
Connect to IP address:	0.0.0.0	0.0.0.0
Disconnect Mode:	Never	Never
Idle timeout (ms):	0	0
RX timeout between packets (ms):	100	100
Socket Packet ID Settings		
STX Rx Detect:	none	none
STX Rx (dec):		
ETX Rx Detect:	none	none
ETX Rx (dec):		
PLC Specific Settings		
STX Tx Append:	none	none
STX Tx (dec):		
ETX Tx Append:	none	none
ETX Tx (dec):		
Strip Rx STX/ETX chars:	no	no
EtherNet/IP Settings		
Rx (To PLC) Transfer Method:	Class1	Class1
Tx (From PLC) Transfer Method:	Class1	Class1
PLC IP address:	N/A	N/A
PLC Controller Slot Number:	N/A	N/A
Maximum PLC Update Rate (ms):	N/A	N/A
Maximum Rx Data Packet Size (bytes):	244	244
Maximum Tx Data Packet Size (bytes):	236	236
Oversized Rx Packet Handling:	Truncate	Truncate
Rx Produced Data Tag/File Name:	N/A	N/A
Tx Sequence Number Checking:	N/A	N/A
Disable Non-Filtered To PLC Rx Queue::	no	no
(PLC-5/SLC/MicroLogix) Rx MS Byte First:	N/A	N/A
(PLC-5/SLC/MicroLogix) Tx MS Byte First:	N/A	N/A

2. Click the **Device #**, which by default opens the *Socket Connection* page for that port.

4.5. Ethernet | Device | Socket Connection Configuration

This section discusses the following area of the *Device Interface Configuration* page:

- *4.5.1. Socket Configuration*
- *4.5.2. Device Socket Packet ID Settings* on Page 102



Device 1 Interface Configuration

Socket Configuration

Device Name:	
Enable:	□
Listen:	□
Listen Port:	8000
Connect To Mode:	Never
Connect Port:	8010
Connect IP Address:	0.0.0.0
Disconnect Mode:	Never
Idle Timer (ms):	0
Rx Timeout Between Packets (ms):	100

Device Socket Packet ID Settings

STX (Start transmission) Rx Detect:	none	Byte 1:	0	Byte 2:	0	(dec)
ETX (End of Transmission) Rx Detect:	none	Byte 1:	0	Byte 2:	0	(dec)

PLC Specific Settings

Start transmission (STX):	none	Byte 1:	0	Byte 2:	0	(dec)
ETX (End of Transmission) Tx Append:	none	Byte 1:	0	Byte 2:	0	(dec)
Strip Rx STX/ETX Chars:	□					

Application Specific Settings

STX (Start of Transmission) Tx Append:	none	Byte 1:	0	Byte 2:	0	(dec)
ETX (End of Transmission) Tx Append:	none	Byte 1:	0	Byte 2:	0	(dec)
Strip Rx STX/ETX:	□					

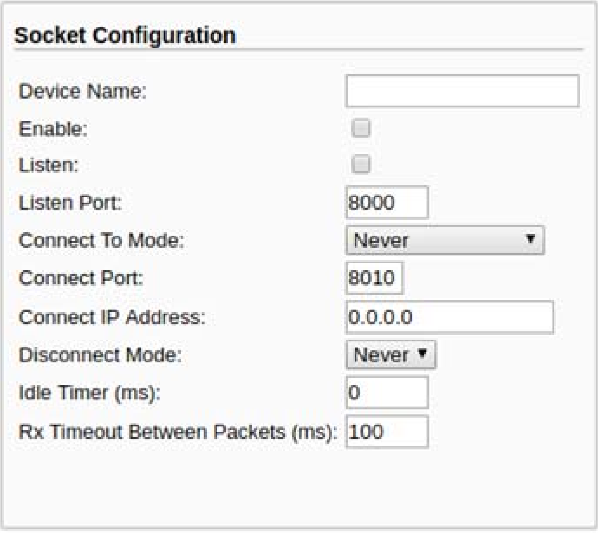
4.5.1. Socket Configuration

Access to the *Device Interface Configuration* page (*Socket Connection* submenu is the default menu displayed) is provided by selecting the corresponding *Device* number (for example, **Socket 1**) under the *Ethernet* menu.

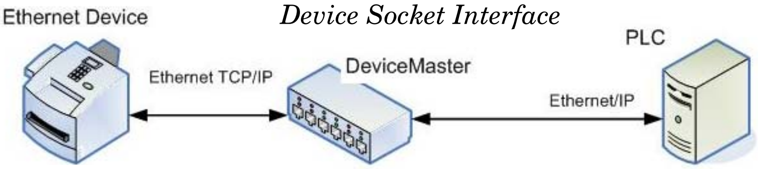
The other area of this page are discussed in the following subsections, which are located under the *4.6. Common Configuration Areas (Serial or Ethernet Device)* section:

- *4.6.1. Ethernet/IP Settings* on Page 104
- *4.6.2. Filtering/Data Extraction Configuration* on Page 107
- *4.6.3. Application Interface | Application TCP Configuration* on Page 111
- *4.7. EtherNet/IP Class 1 Interface Pages* on Page 112

The following table provides information about configuring the *Socket Configuration* area of the *Device Interface Configuration* page.


Socket Configuration

Device Name:	
Enable:	□
Listen:	□
Listen Port:	8000
Connect To Mode:	Never
Connect Port:	8010
Connect IP Address:	0.0.0.0
Disconnect Mode:	Never
Idle Timer (ms):	0
Rx Timeout Between Packets (ms):	100

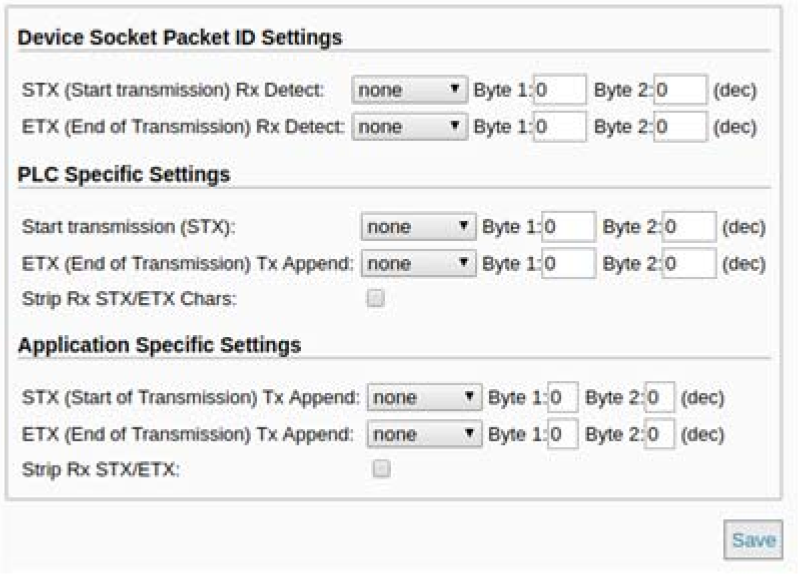
<table>
<tr><th colspan="2">Device Interface Configuration Page | Socket Configuration</th></tr>
<tr><td>Device Name</td><td>A user definable string used to describe the serial interface. Valid characters include a-z, A-Z, 0-9, underscores, spaces and dashes. All other characters are discarded. Up to 80 character ASCII string. The default is blank.</td></tr>
<tr><td>Enable</td><td>This setting enables/disables the *Device Ethernet Device*. Enabling this function allows an Ethernet TCP/IP device to be connected to a PLC and/or application. If both the PLC and application are connected to the device, both can transmit to and receive data from the device socket port. However, the PLC and application cannot communicate directly to each other.
Device Socket Interface


Ethernet Device
Ethernet TCP/IP
DeviceMaster
Ethernet/IP
PLC

</td></tr>
<tr><td>Listen</td><td>Enabling this setting allows the device to connect to the DeviceMaster via an Ethernet TCP/IP socket.
• **Not selected** - Disables listening; the DeviceMaster does not accept connection attempts.
• **Selected** - Enables listening; the DeviceMaster accepts connection attempts from the specified **Listen Port**.</td></tr>
<tr><td>Listen Port</td><td>This is the socket port number on the DeviceMaster the application connects to if the **Device Listen Enable** is selected.</td></tr>
<tr><td>Connect To Mode</td><td>This setting specifies if and how the DeviceMaster attempts to connect to the device at the specified **Connect IP Address** and **Connect Port**.
• **Never** - The DeviceMaster does not attempt to connect to the device.
• **Connect-Always** - The DeviceMaster attempts to connect to the device until a connection is made.
• **Connect-On-Data** - The DeviceMaster does not attempt to connect to the device until there is data to send to the device. Once data is received for the device, the DeviceMaster attempts to connect to the device until a connection is made.</td></tr>
<tr><td>Connect Port</td><td>The device socket port number the DeviceMaster connects to if the **Device Connect To Mode** is set to either **Connect-Always** or **Connect-On-Data**.</td></tr>
<tr><td>Connect IP Address</td><td>The device IP address the DeviceMaster connects to if the **Device Connect To Mode** is set to either **Connect-Always** or **Connect-On-Data**.</td></tr>
<tr><td>Disconnect Mode</td><td>This setting specifies if and how the DeviceMaster disconnects from the device.
• **Never** - The DeviceMaster does not disconnect from the device.
• **Idle** - The DeviceMaster disconnects when there has been no transmit or received data between the device and PLC/application for a specified Idle Timer period.</td></tr>
<tr><td>Idle Timer</td><td>The idle timeout period in milliseconds that is used if the **Device Disconnect Mode** is set to **Idle**.</td></tr>
<tr><td>Rx Timeout Between Packets</td><td>Specifies the following information, once the start of a packet is received:
• How long the DeviceMaster should wait (in milliseconds) before timing-out, if the ETX Rx Detect length is one byte or two bytes and the ETX byte(s) are not received.
• The time to wait in milliseconds between Ethernet packets if the ETX Rx Detect length is set to none.</td></tr>
</table>

4.5.2. Device Socket Packet ID Settings

This subsection discusses the *Socket Packet ID Settings* area of the *Ethernet | Device Interface Configuration* page.



Device Socket Packet ID Settings

STX (Start transmission) Rx Detect: none ▼ Byte 1: 0 Byte 2: 0 (dec)

ETX (End of Transmission) Rx Detect: none ▼ Byte 1: 0 Byte 2: 0 (dec)

PLC Specific Settings

Start transmission (STX): none ▼ Byte 1: 0 Byte 2: 0 (dec)

ETX (End of Transmission) Tx Append: none ▼ Byte 1: 0 Byte 2: 0 (dec)

Strip Rx STX/ETX Chars: □

Application Specific Settings

STX (Start of Transmission) Tx Append: none ▼ Byte 1: 0 Byte 2: 0 (dec)

ETX (End of Transmission) Tx Append: none ▼ Byte 1: 0 Byte 2: 0 (dec)

Strip Rx STX/ETX: □

Device Interface Configuration Page | Device Socket Packet ID Settings	
STX (Start of Transmission) Rx Detect	When enabled, the DeviceMaster detects an STX (start of transmission) byte sequence which is configured as **one byte** or **two bytes** when it receives an Ethernet packet. The length indicates the number of STX bytes, valid values for length are: • **none -** Disables this function and the DeviceMaster accepts the first byte received after the last ETX byte(s) as the start of the next Ethernet packet. • **one byte -** Scans Ethernet data for one STX byte and when the DeviceMaster finds an STX byte it collects the data. If the first byte is not the STX byte, it discards the byte. The DeviceMaster continues to discard the bytes until it finds an STX byte. • **two bytes -** Scans Ethernet data for two STX bytes and when the DeviceMaster finds two STX bytes it collects the data. If the STX bytes cannot be found, it discards the bytes. The DeviceMaster continues to discard the bytes until it finds the two STX bytes. **Byte 1 -** Specifies the character that represents the first STX byte. The DeviceMaster looks for this character in the first STX byte, if the length is **one byte** or **two bytes**. You can specify a value between 0 and 255 in decimal format. **Byte 2 -** Specifies the character that represents the second STX byte. The DeviceMaster looks for this character in the second STX byte, only if the length is two bytes. You can specify a value between 0 and 255 in decimal format.

Device Interface Configuration Page | Device Socket Packet ID Settings (Continued)	
ETX (End of Transmission) Rx Detect	When enabled, the DeviceMaster detects an ETX (end of transmission) byte sequence that is configured as one byte or two bytes marking the end of the Ethernet packet. The length indicates the number of ETX bytes, valid values for length are: • **none** - Disables this function and the DeviceMaster uses the Rx Timeout Between Packets to indicate the end of data packet. • **one byte** - Scans Ethernet data for one ETX byte and when the DeviceMaster finds the ETX byte, it identifies the data as an Ethernet packet. • **two bytes** - Scans Ethernet data for two ETX bytes and when the DeviceMaster finds the ETX bytes, it identifies the data as an Ethernet packet. **Byte 1** - Specifies the character to scan for in the first ETX byte, if the length is one byte or two bytes. You can specify a value between 0 and 255 in decimal format. **Byte 2** - Specifies the character to scan for in the second ETX byte, if the length is two bytes. You can specify a value between 0 and 255 in decimal format.
PLC Specific Settings and Application Specific Settings	
STX Tx Append	When enabled, the DeviceMaster appends an STX (start of transmission) byte sequence which is configured as one byte or two bytes to the beginning of the Ethernet packet before it is sent. The length indicates the number of STX bytes, values for length are: • **none** - Disables this function. • **one byte** - Inserts one STX byte before the data. • **two bytes** - Inserts two STX bytes before the data. **Byte 1** - Specifies the transmit character associated with the first STX byte, if the length is **one byte** or **two bytes**. You can specify a value between 0 and 255 in decimal format. **Byte 2** - Specifies the transmit character associated with the second STX byte, if the length is **two bytes**. You can specify a value between 0 and 255 in decimal format.
ETX Tx Append	When enabled, the DeviceMaster appends an ETX (end of transmission) byte sequence which is configured as **one byte** or **two bytes** to the end of the Ethernet packet before it is sent. The length indicates the number of ETX bytes, valid values for length are: • **none** - Disables this function. • **one byte** - Inserts one ETX byte at the end of the data. • **two bytes** - Inserts two ETX bytes at the end of the data. **Byte 1** - Specifies the transmit character associated with the first ETX byte, if the length is set to **one byte** or **two bytes**. You can specify a value between 0 and 255 in decimal format. **Byte 2** - Specifies the transmit character associated with the second ETX byte, if the length is **two bytes**. You can specify a value between 0 and 255 in decimal format.
Strip Rx STX/ ETX	When you select this check box, the DeviceMaster strips STX/ETX characters from received Ethernet packets. Clear the check box when you do not want the DeviceMaster to strip STX/ETX characters from received Ethernet packets. Packets sent from the PLC to the DeviceMaster (over Ethernet), and then sent out the Ethernet port, are not checked for STX/ETX. No STX/ETX character stripping occurs in these Ethernet packets.

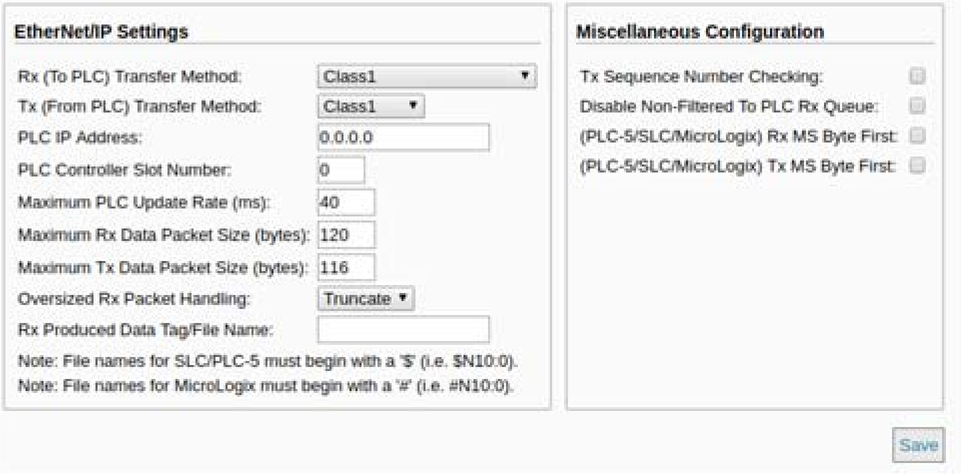
4.6. Common Configuration Areas (Serial or Ethernet Device)

The *Serial* and *Ethernet Configuration* pages have the following areas in common:

- *4.6.1. Ethernet/IP Settings* on Page 104
- *4.6.2. Filtering/Data Extraction Configuration* on Page 107
- *4.6.3. Application Interface | Application TCP Configuration* on Page 111

4.6.1. Ethernet/IP Settings

Use this area to configure the EtherNet/IP settings for the *Serial | Port | EtherNet Settings* page or *Ethernet | Device | EtherNet/IP Settings* page.



EtherNet/IP Settings

Rx (To PLC) Transfer Method:	Class1 ▾
Tx (From PLC) Transfer Method:	Class1 ▾
PLC IP Address:	0.0.0.0
PLC Controller Slot Number:	0
Maximum PLC Update Rate (ms):	40
Maximum Rx Data Packet Size (bytes):	120
Maximum Tx Data Packet Size (bytes):	116
Oversized Rx Packet Handling:	Truncate ▾
Rx Produced Data Tag/File Name:	

Note: File names for SLC/PLC-5 must begin with a '$' (i.e. $N10:0).
Note: File names for MicroLogix must begin with a '#' (i.e. #N10:0).

Miscellaneous Configuration

Tx Sequence Number Checking:	☐
Disable Non-Filtered To PLC Rx Queue:	☐
(PLC-5/SLC/MicroLogix) Rx MS Byte First:	☐
(PLC-5/SLC/MicroLogix) Tx MS Byte First:	☐

For more information on EtherNet/IP settings, see *3.2.1. Serial Port Configuration Object Definition (70 Hex)* on Page 20.

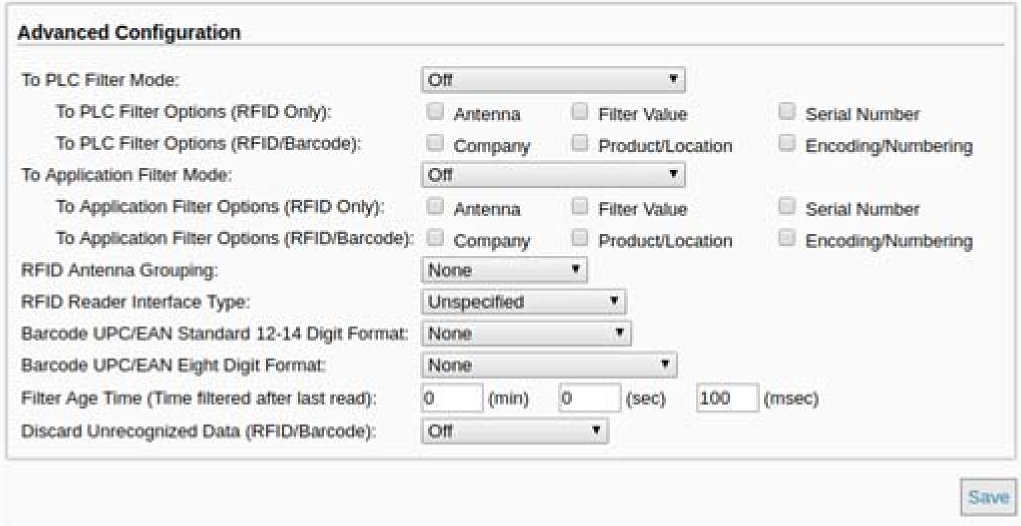
Ethernet/IP Settings (Serial or Socket Port)	
Rx (To PLC) Transfer Method	Specifies the Receive data transfer method used by the DeviceMaster. There are four methods that the DeviceMaster can use to transfer data received from a serial or Ethernet device to the PLC. These methods are: • *Write-to-Tag/File* - The DeviceMaster writes the data directly into a tag or file on the PLC. However, it should only be used if the PLC can scan and consume the data faster than the device can produce it. • *Write-to-Tag/File-Synced* - The DeviceMaster writes the data into a tag or file on the PLC and provides a mechanism to synchronize the data flow between the PLC and the DeviceMaster. Use this method when you want to ensure that the tag or file is not overwritten before the PLC can consume the data. • *Polling* - The PLC requests data on a periodic basis. It provides the ability to control the received data flow. However, it does require periodic data requests and the request rate must be fast enough to ensure that the receive queues on the DeviceMaster do not overflow. • *Class1*- The DeviceMaster sends cycle data to the PLC over a UDP connection that was requested by the PLC. The received data is formatted the same way as for the other methods with new data being indicated by an incremented sequence number.

Ethernet/IP Settings (Serial or Socket Port)	
Tx (from PLC) Transfer Method	Specifies the Transmit data transfer method used by the DeviceMaster. There are two methods that the DeviceMaster can use to transfer data received from a PLC to a serial or Ethernet device. These methods are: • *Write-Msg* – Using a MSG instruction, the PLC sends a formatted write message to the DeviceMaster. • *Class1-* The PLC sends cycle data to the DeviceMaster over a UDP connection that was requested by the PLC. The data to transmit is formatted the same way as for the other methods with new data being indicated by an incremented sequence number.
PLC IP Address	Specifies the IP address for the PLC EtherNet/IP card. ***Note:*** *The Polling and Class1 methods do not use this setting.*
PLC Controller Slot Number *(ControlLogix Family)*	Specifies the slot number on the PLC where the controller resides. The slot numbers typically start at zero for the first slot. This is only required for the ControlLogix family of PLCs. ***Note:*** *The Polling and Class1 methods do not use this setting.*
Maximum PLC Update Rate (Write-To-Tag/File and Class1)	The maximum rate (or minimum time interval) in milliseconds that received messages are: • Sent to the PLC tag in the *Write-To-Tag/File receive* method. • Updated in the Class1 receive method This setting configures the DeviceMaster to space the messages to the PLC in order to prevent overrunning of data before the PLC can process it.
Maximum Rx Data Packet Size	Specifies the maximum acceptable size of a received serial or Ethernet packet. The default is dependent on the number of serial ports on the DeviceMaster.
Maximum Tx Data Packet Size (applies only to Class1)	Specifies the maximum acceptable size of a transmit serial or Ethernet packet. The default is dependent on the number of serial ports on the DeviceMaster. This setting applies only to the Class1 transmit method.
Oversize Rx Packet Handling	Specifies how to process oversized received packets. **Truncate** – truncate the packet to the **Maximum Rx Data Packet Size**. **Drop** – drop the packet. Default = Truncate
Rx (To PLC) Produced Data Tag/ File Name	Specifies the PLC tag or file name. It indicates where to write received data while operating in the *Write-to-Tag/File* or *Write-to-Tag/File-Synced receive* method. This column supports a name containing up to 40 characters. ***Note:*** *The Polling method does not use this attribute.* The maximum length for this tag name is 40 characters. File names for the PLC-5/ SLC PLCs must begin with a **$** (i.e. **$N10:0**).File names for MicroLogix PLCs must begin with a # (i.e. **#10:0**).

Ethernet/IP Settings (Serial or Socket Port)	
Tx Sequence Number Checking	When you select this check box, the DeviceMaster checks the transmit sequence number and performs the following tasks: • Transmit messages with the expected sequence number (last sequence number plus one). • Rejects messages with duplicate sequence numbers (that is, the same sequence number as the previous transmit data message) and increments the *Duplicate Transmit Sequence Error Count*. • Transmits messages with unexpected transmit sequence numbers (that is, sequence numbers that are not the same as or are not equal to the previous sequence number plus one) and increments the *Unexpected Transmit Sequence Error Count*. This check box is clear by default. Clear the check box when you do not want DeviceMaster to check the transmit sequence number.
Disable Non-Filtered TO PLC Rx Queue	If filtering is disabled, only the last message received is sent to the PLC. This box is clear by default.
(PLC-5/SLC) Rx MS Byte First	When you select this check box, the DeviceMaster receives the Most Significant (MS) byte of a 16-bit integer first. This check box is clear by default. Clear the check box when you need to receive the Least Significant (LS) byte of a 16-bit integer first. This applies only to SLC/PLC/MicroLogix PLC interface.
(PLC-5/SLC) Tx MS Byte First	When you select this check box, DeviceMaster transmits the Most Significant (MS) byte of a 16-bit integer first. This check box is clear by default. Clear the check box when you need to transmit the Least Significant (LS) byte of a 16-bit integer first. This applies only to SLC/PLC/MicroLogix PLC interface

4.6.2. Filtering/Data Extraction Configuration

Use this area to configure filtering or data extraction settings for a serial or socket port using the *Serial* | *Port* | *Filter* or *Ethernet* | *Device* | *Filter* pages.



Advanced Configuration

To PLC Filter Mode: Off

To PLC Filter Options (RFID Only): □ Antenna □ Filter Value □ Serial Number

To PLC Filter Options (RFID/Barcode): □ Company □ Product/Location □ Encoding/Numbering

To Application Filter Mode: Off

To Application Filter Options (RFID Only): □ Antenna □ Filter Value □ Serial Number

To Application Filter Options (RFID/Barcode): □ Company □ Product/Location □ Encoding/Numbering

RFID Antenna Grouping: None

RFID Reader Interface Type: Unspecified

Barcode UPC/EAN Standard 12-14 Digit Format: None

Barcode UPC/EAN Eight Digit Format: None

Filter Age Time (Time filtered after last read): 0 (min) 0 (sec) 100 (msec)

Discard Unrecognized Data (RFID/Barcode): Off

Filtering (Serial or Socket Port)	
To PLC Filter Mode	Defines the filter/data extraction mode to be employed on data to be sent to the PLC. • **Off** • **String (128 char max)** - Raw/ASCII data is filtered up to 128 characters (or bytes) in length. • **RFID (EPCglobal formats)** - RFID data in any of the EPCglobal formats is filtered, the associated parameters are extracted, and the extracted data and RFID tag are sent to the PLC in a specified format. • **Barcode (UPC/EAN formats)** - Barcode data in specified UPC/EAN formats is filtered, the associated parameters are extracted, and the extracted data and barcode are sent to the PLC in a specified format. See the barcode format definitions in Attribute 41 on Page 32.
To PLC Filtering Options (RFID Only)	Defines the RFID filtering criteria to the PLC. If an option is enabled, it is used to decide when an RFID tag can be filtered or sent to the PLC. • **Antenna** - Include the antenna number in the filtering criteria. This is data from the RFID reader and not from the RFID tag itself. • **Filter Value** - Include the filter value in the filtering criteria, which is part of the RFID tag data. • **Serial Number** - Include the serial number in the filtering criteria, which is part of the RFID tag data.

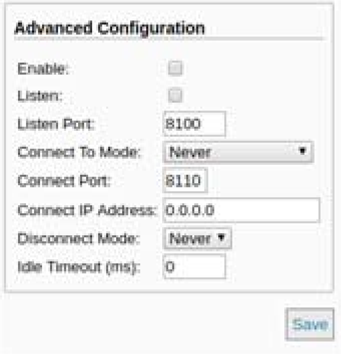
Filtering (Serial or Socket Port) (Continued)	
To PLC Filtering Options (RFID/Barcode)	Defines the RFID filtering criteria and the barcode filtering criteria to the application. If an option is enabled, it is used to decide when a valid RFID tag or barcode can be filtered or sent to the PLC. • **Company** - Include the company code in the filtering criteria, which is part of the RFID tag or barcode data. • **Product/Location** - Include the product/location code in the filtering criteria, which is part of the RFID tag or barcode data. • **Encoding/Numbering** - Include the encoding/numbering code in the filtering criteria, which is part of the RFID tag or barcode data.
To Application Filter Mode	The filter/data extraction mode to be employed on data to be sent to the application. • **Off** • **String (128 char max)** - Raw/ASCII data is filtered up to 128 characters (or bytes) in length. • **RFID (EPCglobal formats)** - RFID data in any of the EPCglobal formats are filtered, the associated parameters are extracted, and the extracted data and RFID tag are sent to the application in a specified format. • **Barcode (UPC/EAN formats)** - Barcode data in specified UPC/EAN formats is filtered, the associated parameters are extracted, and the extracted data and barcode are sent to the application in a specified format. See the barcode format definitions in Attribute 41 on Page 32. ***Note:*** *The application filter mode can be set independently of the PLC filtering mode. The only exceptions are:* • *If the PLC filter mode is set to **RFID**, the application filter mode cannot be set to **Barcode**.* • *If the PLC filter mode is set to **Barcode**, the application filter mode cannot be set to **RFID**.*
To Application Filtering Options (RFID Only)	Defines the RFID filtering criteria to the application. If an option is enabled, it is used to decide when an RFID tag can be filtered or sent to the PLC. • **Antenna** - Include the antenna number in the filtering criteria. This is data from the RFID reader and not part of the RFID tag. • **Filter Value** - Include the filter value in the filtering criteria, which is part of the RFID tag data. • **Serial Number** - Include the serial number in the filtering criteria, which is part of the RFID tag data.
To Application Filtering Options (RFID/Barcode)	Defines the barcode filtering criteria and part of the RFID filtering criteria to the application. If an option is enabled, it is used to decide when a valid RFID tag or barcode can be filtered or sent to the application. • **Company** - Include the company code in the filtering criteria, which is part of the RFID tag or barcode data. • **Product/Location** - Include the product/location code in the filtering criteria, which is part of the RFID tag or barcode data. • **Encoding/Numbering** - Include the encoding/numbering code in the filtering criteria, which is part of the RFID tag or barcode data.

<table>
<tr><th colspan="2">Filtering (Serial or Socket Port) (Continued)</th></tr>
<tr><td>RFID Antenna Grouping</td><td>This setting is applicable only to RFID filtering and only if the antenna filtering option is enabled. It allows the DeviceMaster to filter RFID tags based on antenna groupings. The possible groupings are:
<table>
<tr><th>Setting</th><th>Group 1 Antennas</th><th>Group 2 Antennas</th><th>Group 3 Antennas</th><th>Group N Antennas</th></tr>
<tr><td>None</td><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td>Groups of Twos</td><td>1,2</td><td>3,4</td><td>5,6</td><td>Etc.</td></tr>
<tr><td>Groups of Threes</td><td>1,2,3</td><td>4,5,6</td><td>7,8,9</td><td>Etc.</td></tr>
<tr><td>Groups of Fours</td><td>1,2,3,4</td><td>5,6,7,8</td><td>9,10,11,12</td><td>Etc.</td></tr>
<tr><td>First Two Only</td><td>1,2</td><td>3</td><td>4</td><td>N+1</td></tr>
<tr><td>First Three Only</td><td>1,2,3</td><td>4</td><td>5</td><td>N+2</td></tr>
</table></td></tr>
<tr><td>RFID Reader Interface Type</td><td>Defines the expected RFID data format to be used while operating in the RFID filtering mode. Each Reader Interface Type is unique and pertains to the RFID reader manufacturer. If a different RFID reader is to be used and it provides a similar format to any of the RFID readers listed below, it can also be used in the RFID filtering mode.
• <b>Unspecified</b> - The DeviceMaster assumes a HEX ASCII format and attempts to locate the antenna number.
• <b>Alien (Text Mode)</b> - Specifies the Alien RFID reader <b>Text Mode</b>.
• <b>Alien (Terse Mode)</b> - Specifies the Alien RFID reader <b>Terse Mode</b>.
• <b>Intermec (Hex ASCII Mode)</b> - Specifies the Intermec reader returning data in the <b>Hex ASCII Mode</b>.
See the <i>DeviceMaster UP Filtering and Data Extraction Reference Guide</i> for further details.</td></tr>
<tr><td>Barcode Formats:
UPC/EAN Standard 12-14 Digit
UPD/EAN Eight Digit</td><td>Defines barcode format to be used for both standard and eight digit UPC labels. The term <i>standard</i> refers to UPC-A, EAN-13, JAN, and EAN-14 barcodes which all have ten company/product digits.
The standard and eight digit formats are selected independently and each operates independently. Barcode filtering/data extraction does not function if no format is selected.
<table>
<tr><th>Format</th><th>Numbering Digits</th><th>Company Digits</th><th>Product Digits</th><th>Check Digit</th></tr>
<tr><td><b><i>Standard Formats</i></b></td><td></td><td></td><td></td><td></td></tr>
<tr><td>None</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr>
<tr><td>Company-5/ Product-5</td><td>1-3</td><td>5</td><td>5</td><td>1</td></tr>
<tr><td>Company-6/ Product-4</td><td>1-3</td><td>6</td><td>4</td><td>1</td></tr>
<tr><td>Company-7/ Product-3</td><td>1-3</td><td>7</td><td>3</td><td>1</td></tr>
<tr><td>Company-8/ Product-2</td><td>1-3</td><td>8</td><td>2</td><td>1</td></tr>
<tr><td>Company-9/ Product-1</td><td>1-3</td><td>9</td><td>1</td><td>1</td></tr>
<tr><td><b><i>Eight Digit Formats</i></b></td><td></td><td></td><td></td><td></td></tr>
<tr><td>EAN-8 Number-2/Product 5</td><td>2</td><td>0</td><td>5</td><td>1</td></tr>
<tr><td>EAN-8 Number-3/Product 4</td><td>3</td><td>0</td><td>4</td><td>1</td></tr>
<tr><td>UPC-E</td><td></td><td></td><td></td><td></td></tr>
</table>
See the <i>DeviceMaster UP Filtering and Data Extraction Reference Guide</i> for further details.</td></tr>
<tr><td>Filter Age Time (Time filtered after last read)</td><td>Defines the time a filter string, RFID tag, or barcode continues to be filtered after the last time it was received.
If an entry is received before the <b>Filter Age Time</b> has passed, the entry is filtered and the data is not sent to the PLC and/or application. However, if the <b>Filter Age Time</b> has passed, it passes filtering and be sent to the PLC and/or application.</td></tr>
</table>

Filtering (Serial or Socket Port) (Continued)	
Discard Unrecognized Data (RFID/Barcode)	Specifies what to do with unrecognized RFID or barcode data. • **Off -** Sends unrecognized data to the PLC and/or application. • **To-PLC -** Discards unrecognized data to the PLC. Allows sending of unrecognized data to the application. • **To-Application -** Discards unrecognized data to the application. Allows sending of unrecognized data to the PLC. • **To-PLC/Application -** Discards unrecognized data to both the PLC and application.

4.6.3. Application Interface | Application TCP Configuration

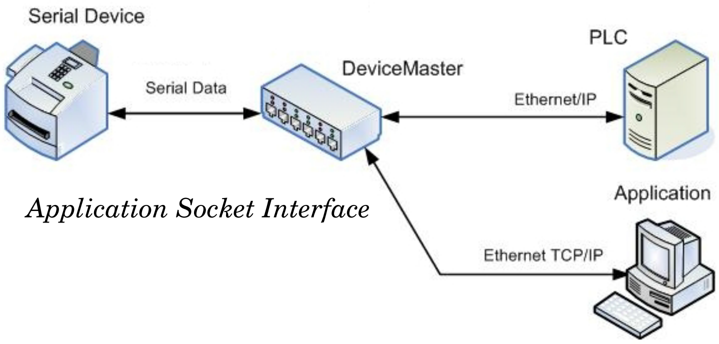
Use this area to configure application TCP connection settings for a serial or socket port using the appropriate *Application TCP Configuration* page.



Advanced Configuration

Enable: ☐
Listen: ☐
Listen Port: 8100
Connect To Mode: Never ▾
Connect Port: 8110
Connect IP Address: 0.0.0.0
Disconnect Mode: Never ▾
Idle Timeout (ms): 0

Save

Application TCP Configuration (Serial or Socket Port)	
Enable	Enables/disables the *Application Socket Interface*. Enabling this function allows an application to be connected to the device serial/socket port.  Serial Device Serial Data DeviceMaster PLC Ethernet/IP Application Ethernet TCP/IP *Application Socket Interface* If both the PLC and application are connected to the device serial/socket port, both can transmit to and receive data from the serial/socket port. However, the PLC and application cannot communicate directly to each other.
Listen	Enabling this setting allows the application to connect to the DeviceMaster via an Ethernet TCP/IP socket. • **Not selected -** Disables listening and the DeviceMaster does not accept connection attempts. • **Selected -** Enables listening and the DeviceMaster accepts connection attempts from the specified **Listen Port**.
Listen Port	The socket port number on the DeviceMaster the application connects to if the **Application Listen Enable** is selected.
Connect To Mode	Specifies if and how the DeviceMaster attempts to connect to the application at the specified **Connect IP Address** and **Connect Port**. • **Never -** The DeviceMaster does not attempt to connect to the application. • **Connect-Always -** The DeviceMaster attempts to connect to the application until a connection is made. • **Connect-On-Data** – The DeviceMaster does not attempt to connect to the application until there is data to send to the application. Once data is received from the serial/socket device, the DeviceMaster attempts to connect to the application until a connection is made.
Connect Port	The application socket port number the DeviceMaster connects to if the **Application Connect To Mode** is set to either **Connect-Always** or **Connect-On-Data**.

Application TCP Configuration (Serial or Socket Port) (Continued)	
Connect IP Address	The application IP address the DeviceMaster connects to if the **Application Connect To Mode** is set to either **Connect-Always** or **Connect-On-Data**.
Disconnect Mode	Controls if and how the DeviceMaster disconnects from an application. • **Never** – The DeviceMaster does not disconnect from the application. • **Idle** - The DeviceMaster disconnects when there has been no transmit or received data between the serial/socket device and application for a specified **Idle Timer** period.
Idle Timer	The idle timeout period in milliseconds that is used if the application **Disconnect Mode** is set to **Idle**.

4.7. EtherNet/IP Class 1 Interface Pages

The DeviceMaster provides an array of highly informative Class1 interface web pages designed to:

- Provide both Class1 overview pages, which display all Class1 interfaces, as well as serial and/or Ethernet device interface specific pages.
- Provide easily understood information to aid the PLC programmer.
  - Instance numbers
  - Data offsets for each serial and/or Ethernet device interface
  - Instance and connection lengths
  - The interface
- Provide the ability to easily reconfigure the Class1 interface for:
  - Evenly sized instances for all serial port and Ethernet device interfaces
  - Evenly sized instances for only serial port interfaces
  - Evenly sized instances for only Ethernet device interfaces

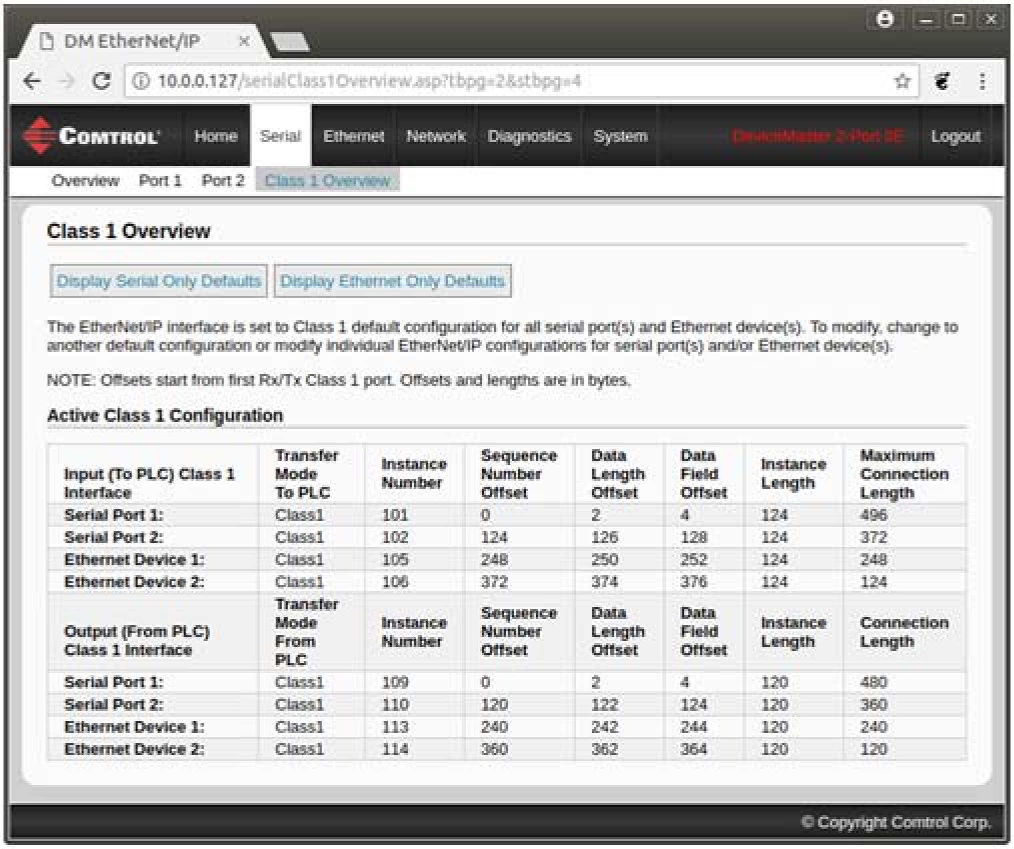
4.7.1. Class1 Overview Pages

These pages display the active configuration s well as possible default configurations that can be displayed and selected.

4.7.1.1. Active Class1 Configuration

This page displays the current active Class1 configuration. Depending on the active configuration, various buttons are displayed on the top of the page that provide available display and configuration options.

4.7.1.1.1. Active Class1 Configuration - All Interfaces Set to Defaults


DM EtherNet/IP
10.0.0.127/serialClass1Overview.asp?tbpg=2&stbpg=4
COMTROL
Home
Serial
Ethernet
Network
Diagnostics
System
DeviceMaster 2-Port 2E
Logout
Overview
Port 1
Port 2
Class 1 Overview
Display Serial Only Defaults
Display Ethernet Only Defaults
© Copyright Comtrol Corp.

Class 1 Overview

The EtherNet/IP interface is set to Class 1 default configuration for all serial port(s) and Ethernet device(s). To modify, change to another default configuration or modify individual EtherNet/IP configurations for serial port(s) and/or Ethernet device(s).

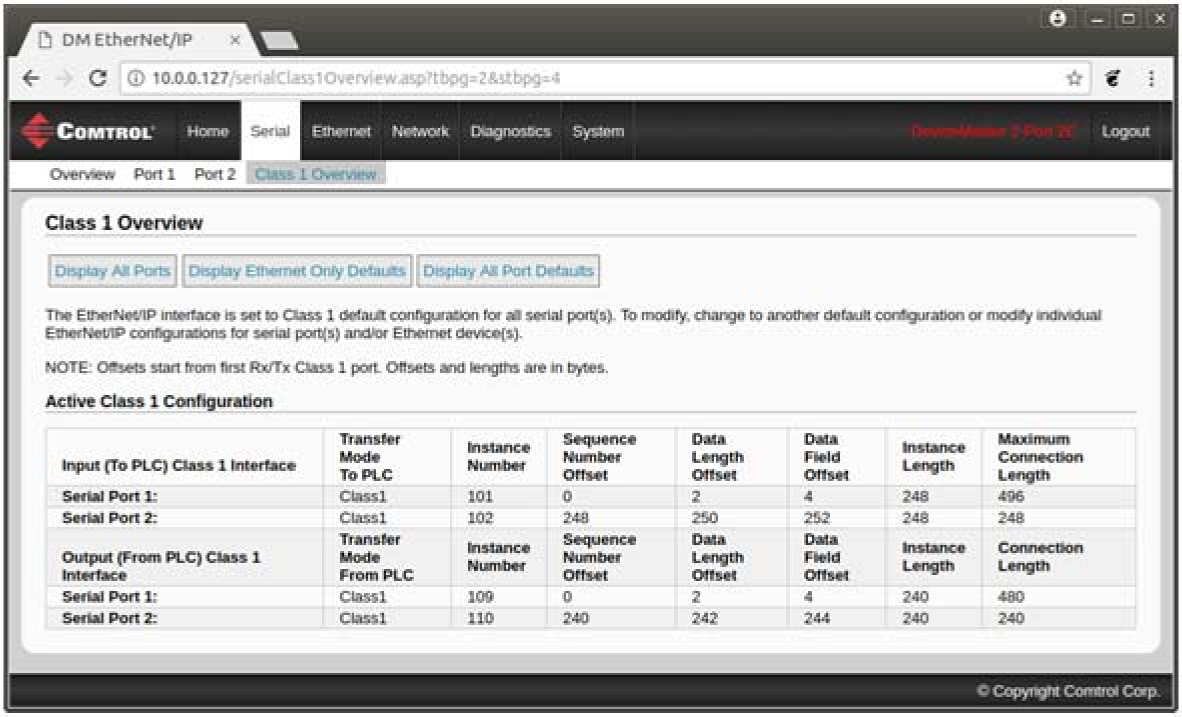
NOTE: Offsets start from first Rx/Tx Class 1 port. Offsets and lengths are in bytes.

Active Class 1 Configuration

Input (To PLC) Class 1 Interface	Transfer Mode To PLC	Instance Number	Sequence Number Offset	Data Length Offset	Data Field Offset	Instance Length	Maximum Connection Length
Serial Port 1:	Class1	101	0	2	4	124	496
Serial Port 2:	Class1	102	124	126	128	124	372
Ethernet Device 1:	Class1	105	248	250	252	124	248
Ethernet Device 2:	Class1	106	372	374	376	124	124
Output (From PLC) Class 1 Interface	**Transfer Mode From PLC**	**Instance Number**	**Sequence Number Offset**	**Data Length Offset**	**Data Field Offset**	**Instance Length**	**Connection Length**
Serial Port 1:	Class1	109	0	2	4	120	480
Serial Port 2:	Class1	110	120	122	124	120	360
Ethernet Device 1:	Class1	113	240	242	244	120	240
Ethernet Device 2:	Class1	114	360	362	364	120	120

4.7.1.1.2. Active Class1 Configuration – Only Serial Ports Set to Defaults

Display Only Class1 Ports:



Class 1 Overview

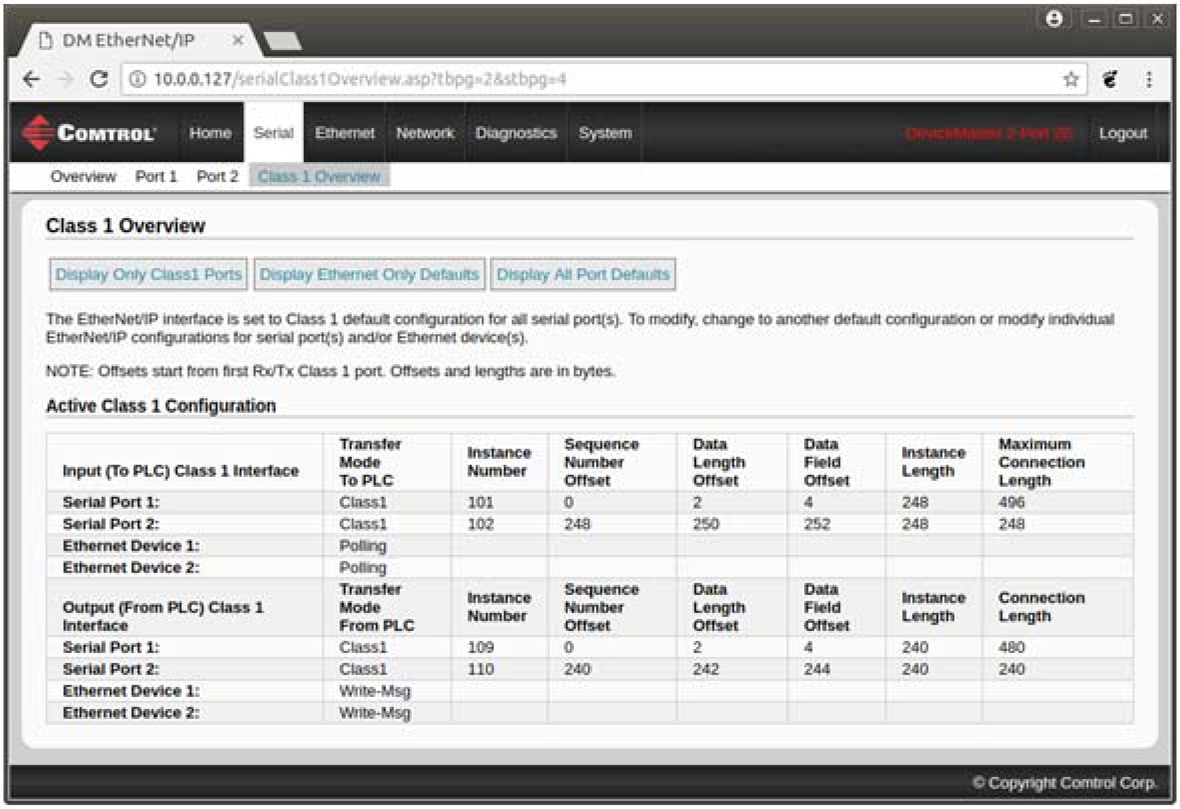
The EtherNet/IP interface is set to Class 1 default configuration for all serial port(s). To modify, change to another default configuration or modify individual EtherNet/IP configurations for serial port(s) and/or Ethernet device(s).

NOTE: Offsets start from first Rx/Tx Class 1 port. Offsets and lengths are in bytes.

Active Class 1 Configuration

Input (To PLC) Class 1 Interface	Transfer Mode To PLC	Instance Number	Sequence Number Offset	Data Length Offset	Data Field Offset	Instance Length	Maximum Connection Length
Serial Port 1:	Class1	101	0	2	4	248	496
Serial Port 2:	Class1	102	248	250	252	248	248
Output (From PLC) Class 1 Interface	**Transfer Mode From PLC**	**Instance Number**	**Sequence Number Offset**	**Data Length Offset**	**Data Field Offset**	**Instance Length**	**Connection Length**
Serial Port 1:	Class1	109	0	2	4	240	480
Serial Port 2:	Class1	110	240	242	244	240	240

Display All Ports:


DM EtherNet/IP
10.0.0.127/serialClass1Overview.asp?tbpg=2&stbpg=4
Comtrol
Home
Serial
Ethernet
Network
Diagnostics
System
DeviceMaster 2-Port 2E
Logout
Overview
Port 1
Port 2
Class 1 Overview
Display Only Class1 Ports
Display Ethernet Only Defaults
Display All Port Defaults
© Copyright Comtrol Corp.

Class 1 Overview

The EtherNet/IP interface is set to Class 1 default configuration for all serial port(s). To modify, change to another default configuration or modify individual EtherNet/IP configurations for serial port(s) and/or Ethernet device(s).

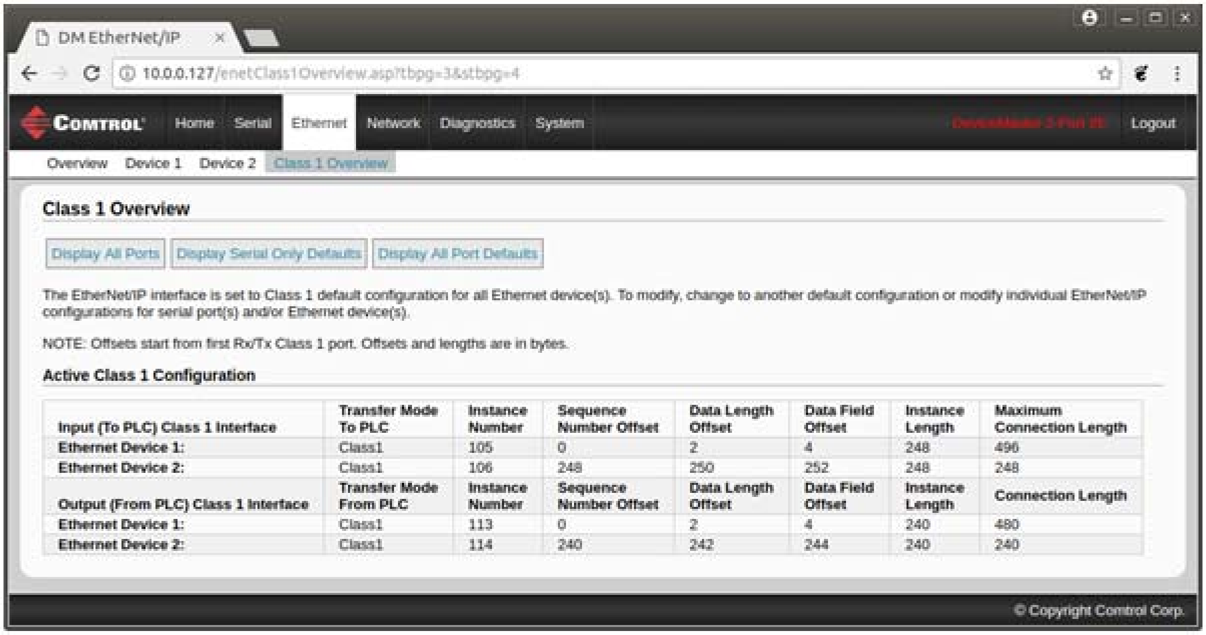
NOTE: Offsets start from first Rx/Tx Class 1 port. Offsets and lengths are in bytes.

Active Class 1 Configuration

Input (To PLC) Class 1 Interface	Transfer Mode To PLC	Instance Number	Sequence Number Offset	Data Length Offset	Data Field Offset	Instance Length	Maximum Connection Length
Serial Port 1:	Class1	101	0	2	4	248	496
Serial Port 2:	Class1	102	248	250	252	248	248
Ethernet Device 1:	Polling						
Ethernet Device 2:	Polling						
Output (From PLC) Class 1 Interface	**Transfer Mode From PLC**	**Instance Number**	**Sequence Number Offset**	**Data Length Offset**	**Data Field Offset**	**Instance Length**	**Connection Length**
Serial Port 1:	Class1	109	0	2	4	240	480
Serial Port 2:	Class1	110	240	242	244	240	240
Ethernet Device 1:	Write-Msg						
Ethernet Device 2:	Write-Msg						

4.7.1.1.3. Active Class1 Configuration – Only TCP/IP Sockets Set to Defaults

Display Only Class1 Ports:



Class 1 Overview

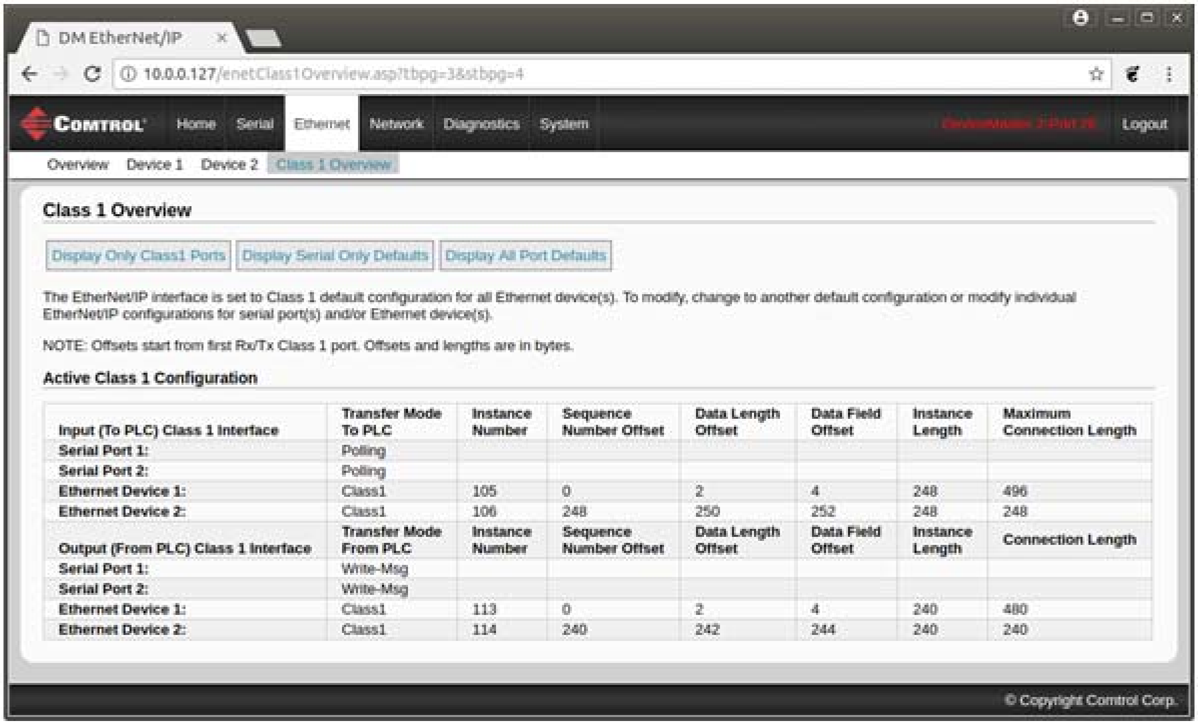
The EtherNet/IP interface is set to Class 1 default configuration for all Ethernet device(s). To modify, change to another default configuration or modify individual EtherNet/IP configurations for serial port(s) and/or Ethernet device(s).

NOTE: Offsets start from first Rx/Tx Class 1 port. Offsets and lengths are in bytes.

Active Class 1 Configuration

Input (To PLC) Class 1 Interface	Transfer Mode To PLC	Instance Number	Sequence Number Offset	Data Length Offset	Data Field Offset	Instance Length	Maximum Connection Length
Ethernet Device 1:	Class1	105	0	2	4	248	496
Ethernet Device 2:	Class1	106	248	250	252	248	248
Output (From PLC) Class 1 Interface	**Transfer Mode From PLC**	**Instance Number**	**Sequence Number Offset**	**Data Length Offset**	**Data Field Offset**	**Instance Length**	**Connection Length**
Ethernet Device 1:	Class1	113	0	2	4	240	480
Ethernet Device 2:	Class1	114	240	242	244	240	240

Display All Ports:


DM EtherNet/IP
10.0.0.127/enetClass1Overview.asp?tbpg=3&stbpg=4
COMTROL
Home
Serial
Ethernet
Network
Diagnostics
System
DeviceMaster 2-Port 2E
Logout
Overview
Device 1
Device 2
Class 1 Overview
Display Only Class1 Ports
Display Serial Only Defaults
Display All Port Defaults
© Copyright Comtrol Corp.

Class 1 Overview

The EtherNet/IP interface is set to Class 1 default configuration for all Ethernet device(s). To modify, change to another default configuration or modify individual EtherNet/IP configurations for serial port(s) and/or Ethernet device(s).

NOTE: Offsets start from first Rx/Tx Class 1 port. Offsets and lengths are in bytes.

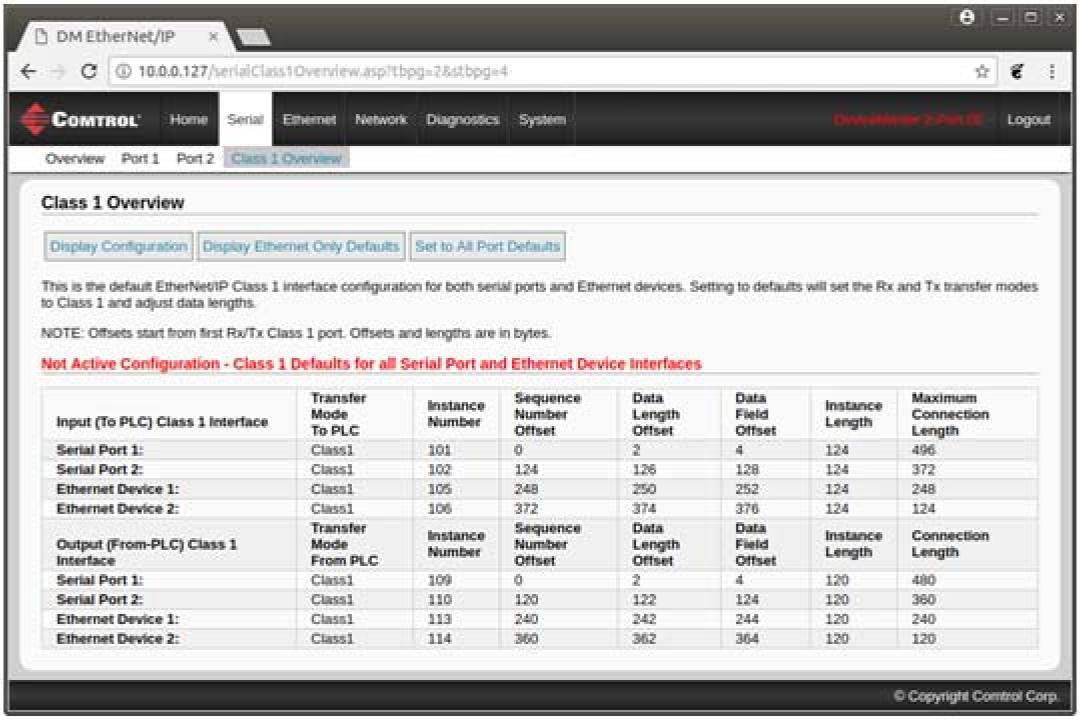
Active Class 1 Configuration

Input (To PLC) Class 1 Interface	Transfer Mode To PLC	Instance Number	Sequence Number Offset	Data Length Offset	Data Field Offset	Instance Length	Maximum Connection Length
Serial Port 1:	Polling						
Serial Port 2:	Polling						
Ethernet Device 1:	Class1	105	0	2	4	248	496
Ethernet Device 2:	Class1	106	248	250	252	248	248
Output (From PLC) Class 1 Interface	**Transfer Mode From PLC**	**Instance Number**	**Sequence Number Offset**	**Data Length Offset**	**Data Field Offset**	**Instance Length**	**Connection Length**
Serial Port 1:	Write-Msg						
Serial Port 2:	Write-Msg						
Ethernet Device 1:	Class1	113	0	2	4	240	480
Ethernet Device 2:	Class1	114	240	242	244	240	240

4.7.1.2. Default Class1 Configurations

These pages display the available default Class1 configurations. If a default configuration is currently active, it will not be available for viewing as a default.

4.7.1.2.1. All Serial Port and TCP/IP Socket Interface Default



Class 1 Overview

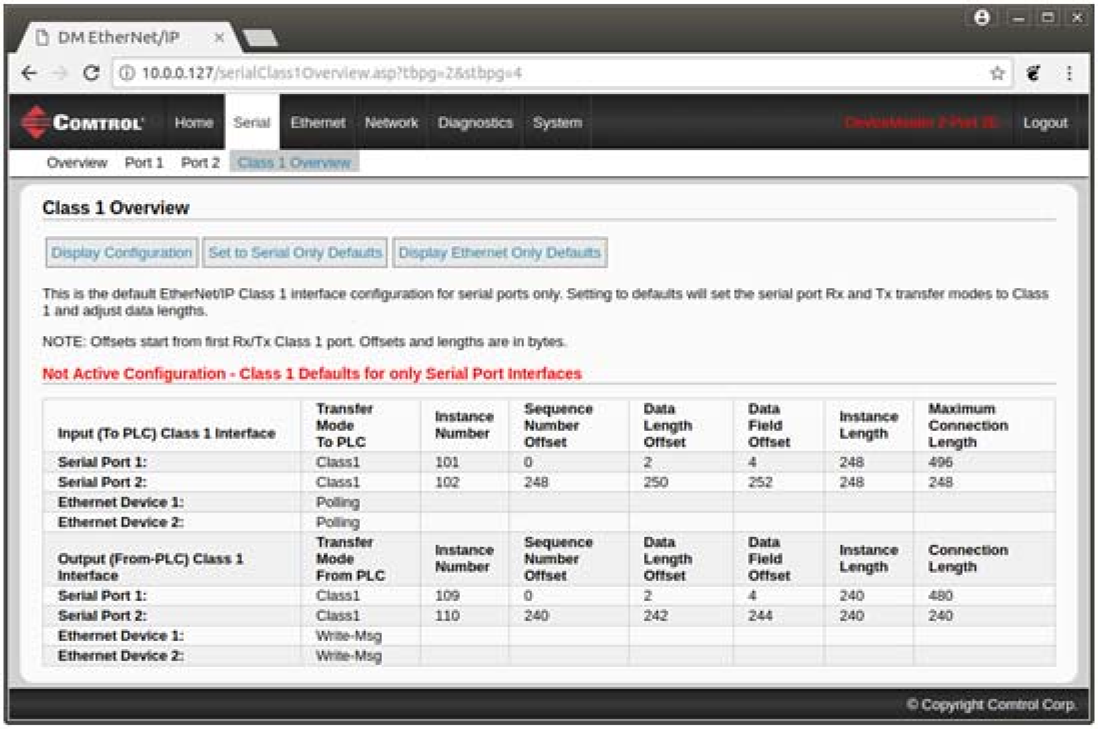
This is the default EtherNet/IP Class 1 interface configuration for both serial ports and Ethernet devices. Setting to defaults will set the Rx and Tx transfer modes to Class 1 and adjust data lengths.

NOTE: Offsets start from first Rx/Tx Class 1 port. Offsets and lengths are in bytes.

Not Active Configuration - Class 1 Defaults for all Serial Port and Ethernet Device Interfaces

Input (To PLC) Class 1 Interface	Transfer Mode To PLC	Instance Number	Sequence Number Offset	Data Length Offset	Data Field Offset	Instance Length	Maximum Connection Length
Serial Port 1:	Class1	101	0	2	4	124	496
Serial Port 2:	Class1	102	124	126	128	124	372
Ethernet Device 1:	Class1	105	248	250	252	124	248
Ethernet Device 2:	Class1	106	372	374	376	124	124
Output (From-PLC) Class 1 Interface	**Transfer Mode From PLC**	**Instance Number**	**Sequence Number Offset**	**Data Length Offset**	**Data Field Offset**	**Instance Length**	**Connection Length**
Serial Port 1:	Class1	109	0	2	4	120	480
Serial Port 2:	Class1	110	120	122	124	120	360
Ethernet Device 1:	Class1	113	240	242	244	120	240
Ethernet Device 2:	Class1	114	360	362	364	120	120

4.7.1.2.2. Only Serial Port(s) Default


DM EtherNet/IP
10.0.0.127/serialClass1Overview.asp?tbpg=2&stbpg=4
Comtrol
Home
Serial
Ethernet
Network
Diagnostics
System
Logout
Overview
Port 1
Port 2
Class 1 Overview
Display Configuration
Set to Serial Only Defaults
Display Ethernet Only Defaults
© Copyright Comtrol Corp.

Class 1 Overview

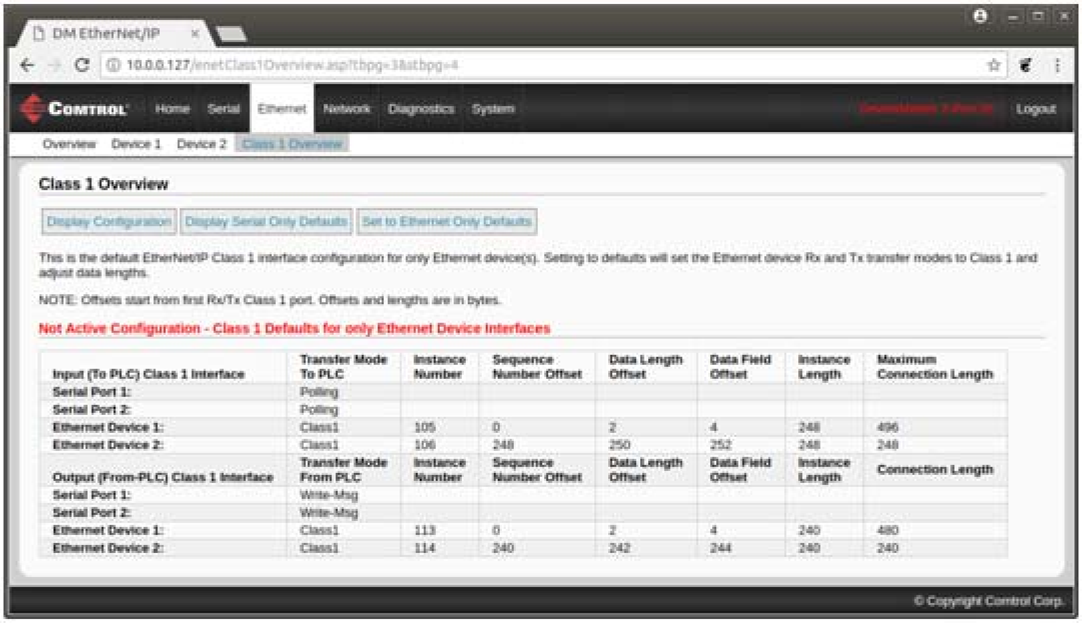
This is the default EtherNet/IP Class 1 interface configuration for serial ports only. Setting to defaults will set the serial port Rx and Tx transfer modes to Class 1 and adjust data lengths.

NOTE: Offsets start from first Rx/Tx Class 1 port. Offsets and lengths are in bytes.

Not Active Configuration - Class 1 Defaults for only Serial Port Interfaces

Input (To PLC) Class 1 Interface	Transfer Mode To PLC	Instance Number	Sequence Number Offset	Data Length Offset	Data Field Offset	Instance Length	Maximum Connection Length
Serial Port 1:	Class1	101	0	2	4	248	496
Serial Port 2:	Class1	102	248	250	252	248	248
Ethernet Device 1:	Polling						
Ethernet Device 2:	Polling						
Output (From-PLC) Class 1 Interface	**Transfer Mode From PLC**	**Instance Number**	**Sequence Number Offset**	**Data Length Offset**	**Data Field Offset**	**Instance Length**	**Connection Length**
Serial Port 1:	Class1	109	0	2	4	240	480
Serial Port 2:	Class1	110	240	242	244	240	240
Ethernet Device 1:	Write-Msg						
Ethernet Device 2:	Write-Msg						

4.7.1.2.3. Only TCP/IP Socket(s) Default


DM EtherNet/IP
10.0.0.127/enetClass1Overview.asp?tbpg=3&stbpg=4
Comtrol
Home
Serial
Ethernet
Network
Diagnostics
System
Logout
Overview
Device 1
Device 2
Class 1 Overview
Display Configuration
Display Serial Only Defaults
Set to Ethernet Only Defaults
© Copyright Comtrol Corp.

Class 1 Overview

This is the default EtherNet/IP Class 1 interface configuration for only Ethernet device(s). Setting to defaults will set the Ethernet device Rx and Tx transfer modes to Class 1 and adjust data lengths.

NOTE: Offsets start from first Rx/Tx Class 1 port. Offsets and lengths are in bytes.

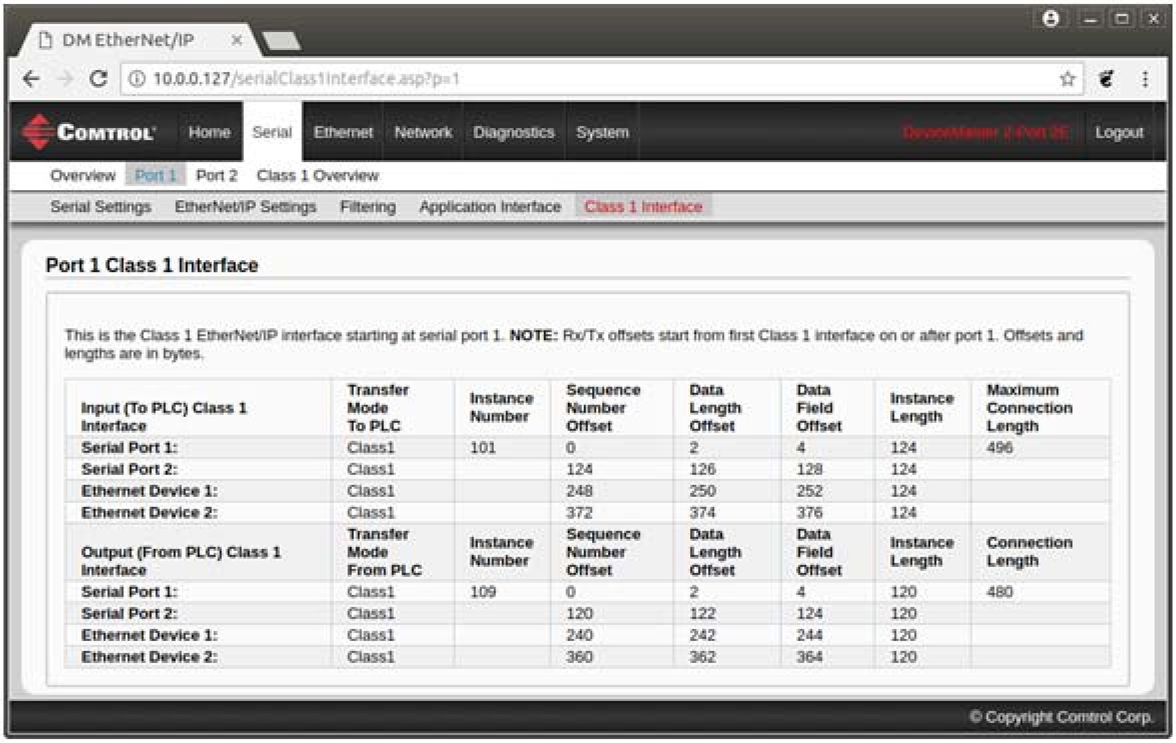
Not Active Configuration - Class 1 Defaults for only Ethernet Device Interfaces

Input (To PLC) Class 1 Interface	Transfer Mode To PLC	Instance Number	Sequence Number Offset	Data Length Offset	Data Field Offset	Instance Length	Maximum Connection Length
Serial Port 1:	Polling						
Serial Port 2:	Polling						
Ethernet Device 1:	Class1	105	0	2	4	248	496
Ethernet Device 2:	Class1	106	248	250	252	248	248
Output (From-PLC) Class 1 Interface	**Transfer Mode From PLC**	**Instance Number**	**Sequence Number Offset**	**Data Length Offset**	**Data Field Offset**	**Instance Length**	**Connection Length**
Serial Port 1:	Write-Msg						
Serial Port 2:	Write-Msg						
Ethernet Device 1:	Class1	113	0	2	4	240	480
Ethernet Device 2:	Class1	114	240	242	244	240	240

4.8. Class1 Interface Specific Pages

The Class1 interface pages are designed to specify the Class1 interface related to that serial port or TCP/IP socket.

4.8.1. Class1 Interface Starting at Serial Port 1 (Example)

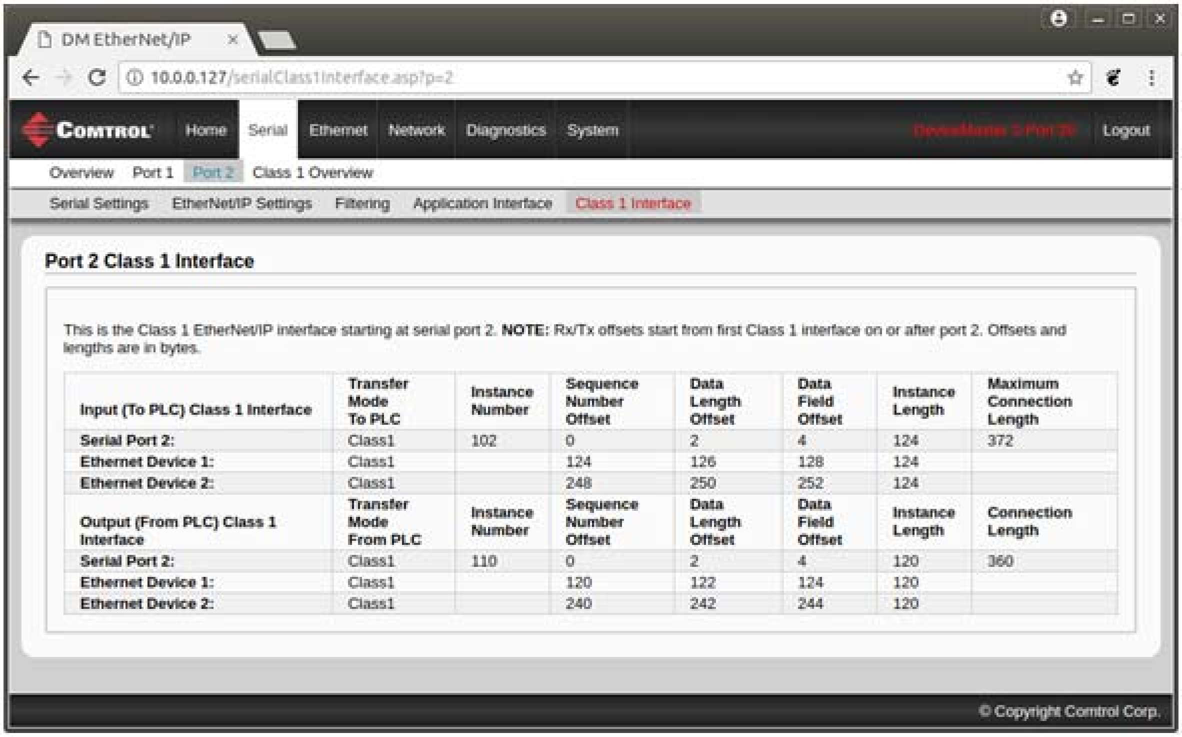


Port 1 Class 1 Interface

This is the Class 1 EtherNet/IP interface starting at serial port 1. **NOTE:** Rx/Tx offsets start from first Class 1 interface on or after port 1. Offsets and lengths are in bytes.

Input (To PLC) Class 1 Interface	**Transfer Mode To PLC**	**Instance Number**	**Sequence Number Offset**	**Data Length Offset**	**Data Field Offset**	**Instance Length**	**Maximum Connection Length**
Serial Port 1:	Class1	101	0	2	4	124	496
Serial Port 2:	Class1		124	126	128	124	
Ethernet Device 1:	Class1		248	250	252	124	
Ethernet Device 2:	Class1		372	374	376	124	
Output (From PLC) Class 1 Interface	**Transfer Mode From PLC**	**Instance Number**	**Sequence Number Offset**	**Data Length Offset**	**Data Field Offset**	**Instance Length**	**Connection Length**
Serial Port 1:	Class1	109	0	2	4	120	480
Serial Port 2:	Class1		120	122	124	120	
Ethernet Device 1:	Class1		240	242	244	120	
Ethernet Device 2:	Class1		360	362	364	120	

4.8.2. Class Interface Starting at Serial Port 2 (Example)

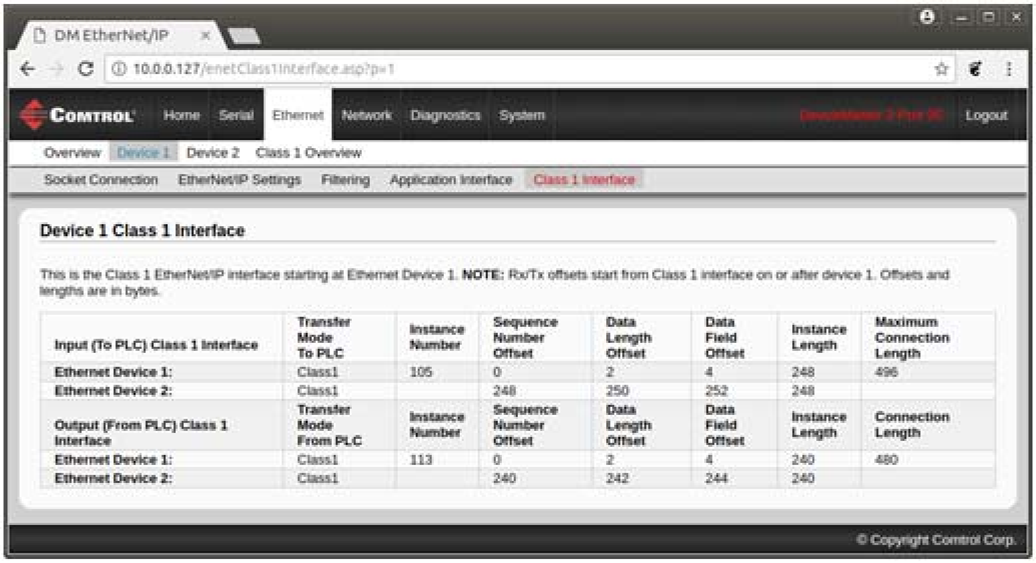


Port 2 Class 1 Interface

This is the Class 1 EtherNet/IP interface starting at serial port 2. **NOTE:** Rx/Tx offsets start from first Class 1 interface on or after port 2. Offsets and lengths are in bytes.

Input (To PLC) Class 1 Interface	Transfer Mode To PLC	Instance Number	Sequence Number Offset	Data Length Offset	Data Field Offset	Instance Length	Maximum Connection Length
Serial Port 2:	Class1	102	0	2	4	124	372
Ethernet Device 1:	Class1		124	126	128	124	
Ethernet Device 2:	Class1		248	250	252	124	
Output (From PLC) Class 1 Interface	**Transfer Mode From PLC**	**Instance Number**	**Sequence Number Offset**	**Data Length Offset**	**Data Field Offset**	**Instance Length**	**Connection Length**
Serial Port 2:	Class1	110	0	2	4	120	360
Ethernet Device 1:	Class1		120	122	124	120	
Ethernet Device 2:	Class1		240	242	244	120	

4.8.3. Class 1 Interface Starting at TCP/IP Device 1 (Example)

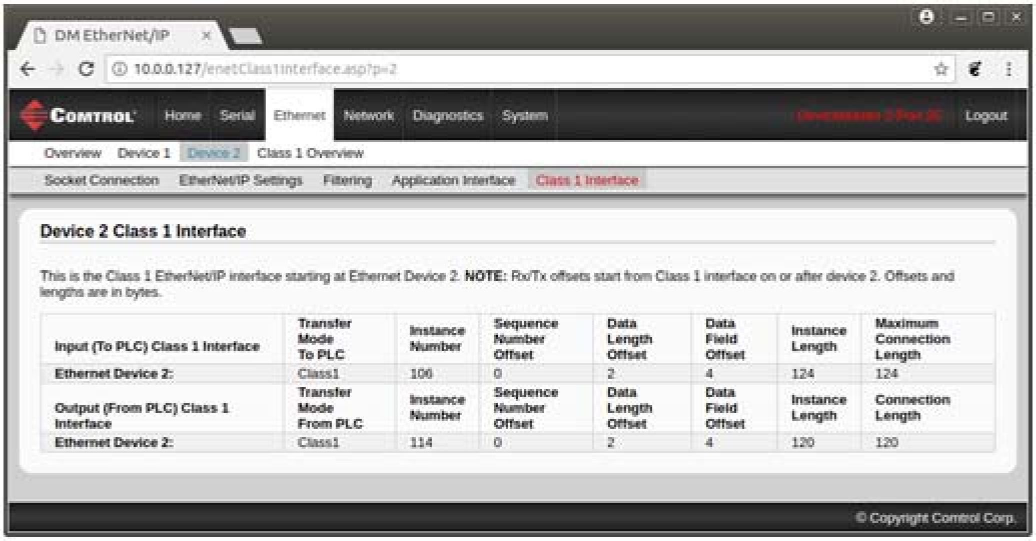


Device 1 Class 1 Interface

This is the Class 1 EtherNet/IP interface starting at Ethernet Device 1. **NOTE:** Rx/Tx offsets start from Class 1 interface on or after device 1. Offsets and lengths are in bytes.

Input (To PLC) Class 1 Interface	Transfer Mode To PLC	Instance Number	Sequence Number Offset	Data Length Offset	Data Field Offset	Instance Length	Maximum Connection Length
Ethernet Device 1:	Class1	105	0	2	4	248	496
Ethernet Device 2:	Class1		248	250	252	248	
Output (From PLC) Class 1 Interface	**Transfer Mode From PLC**	**Instance Number**	**Sequence Number Offset**	**Data Length Offset**	**Data Field Offset**	**Instance Length**	**Connection Length**
Ethernet Device 1:	Class1	113	0	2	4	240	480
Ethernet Device 2:	Class1		240	242	244	240	

4.8.4. Class 1 Interface Starting at TCP/IP Devcie 2 (Example)



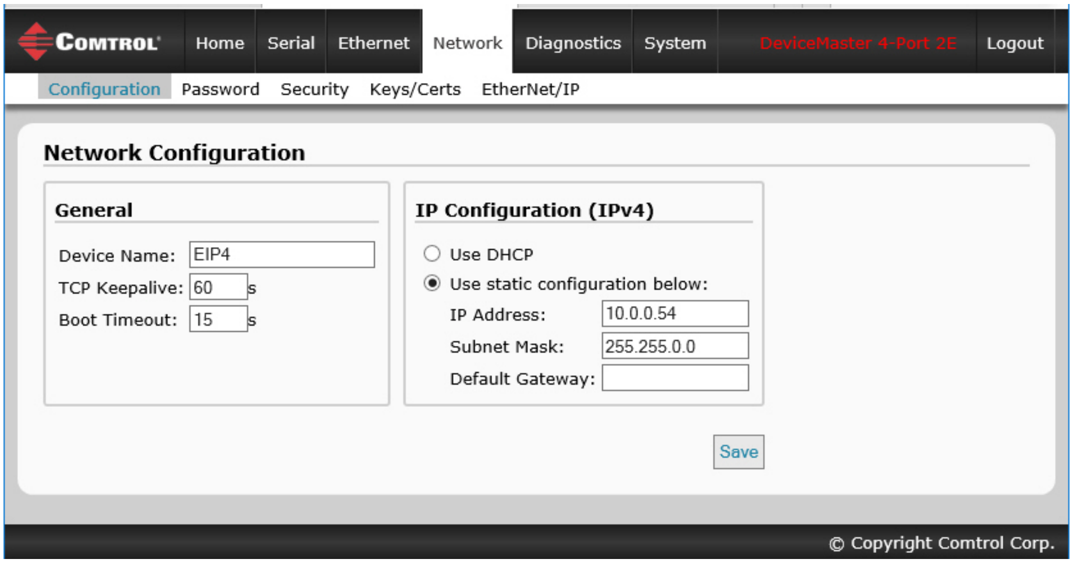
Device 2 Class 1 Interface

This is the Class 1 EtherNet/IP interface starting at Ethernet Device 2. **NOTE:** Rx/Tx offsets start from Class 1 interface on or after device 2. Offsets and lengths are in bytes.

Input (To PLC) Class 1 Interface	Transfer Mode To PLC	Instance Number	Sequence Number Offset	Data Length Offset	Data Field Offset	Instance Length	Maximum Connection Length
Ethernet Device 2:	Class1	106	0	2	4	124	124
Output (From PLC) Class 1 Interface	**Transfer Mode From PLC**	**Instance Number**	**Sequence Number Offset**	**Data Length Offset**	**Data Field Offset**	**Instance Length**	**Connection Length**
Ethernet Device 2:	Class1	114	0	2	4	120	120

4.9. Network Configuration Page

You can use the *Network Configuration* page to change the DeviceMaster network configuration after using PortVision DX for initial network configuration.



Network Configuration

General

Device Name: EIP4

TCP Keepalive: 60 s

Boot Timeout: 15 s

IP Configuration (IPv4)

○ Use DHCP

● Use static configuration below:

IP Address: 10.0.0.54

Subnet Mask: 255.255.0.0

Default Gateway:

4.9.1. Network Configuration Page Options

The *Network Configuration* page provides the following options.

Network Configuration Page	
Device Name	You can enter a 16-character **Host Name** to identify this DeviceMaster on the **Home** page.
TCP Keepalive **Default = 60 seconds**	This option allows you to set the amount of time in seconds that the DeviceMaster waits until it closes this connection and frees all the ports associated with it. The TCP protocol has an optional keepalive feature where the two network stacks periodically ping each other to make sure the connection is still up. In the UNIX world, this feature is usually known as the SOKEEPALIVE socket option. By default, this keepalive feature is not enabled, and the only time you know a connection is down is when you try to write something and the other end does not acknowledge you. The potential issue with the DeviceMaster occurs when data flow was mostly one-way, and the receiver of the data was configured to initiate the TCP connection. If the sender of the data was rebooted, it would wait for the initiation a connection, discarding data. The receiver would wait forever for data and never realize that the other end wasn't connected any more and that the TCP connection needed to be re-established. Enabling the TCP keepalive feature for a connection solves this problem: the TCP stack periodically pings the other end. If the connection has gone away, the DeviceMaster is notified, so that it can attempt to re-establish the TCP connection.
Boot Timeout **Default = 15 seconds**	Allows you to change the bootloader time-out value before the EtherNet/IP firmware loads. You may need to increase this time-out value to 45 for compatibility with spanning tree devices (normally switches). If you change the time-out value to 0, this prevents EtherNet/IP from loading.

Network Configuration Page	
Use DHCP	Configures the DeviceMaster to use DHCPv4 mode. If you select Use DHCP, the IP Address field below is disabled and set to 0.0.0.0. See your System Administrator to acquire a unique reserved IPv4 address if you are using DHCP. They will need the MAC address of the unit to provide you with an IPv4 address.
Use Static Config Below	Configures the DeviceMaster to use DHCPv4 mode. Configures the DeviceMaster with the static IPv4 address information that you provide in the IP Address, Subnet Netmask, and Default Gateway fields below. The DeviceMaster is shipped from the factory with these default IP network settings: • IP address = 192.168.250.250 • Subnet Netmask = 255.255.0.0 • Default Gateway address = 192.168.250.1

4.9.2. Changing the Network Configuration

You can change the IP address configuration after initial network configuration.

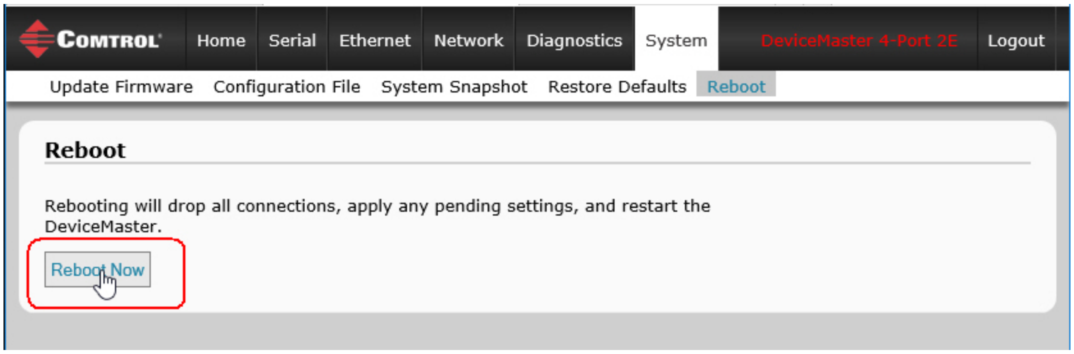
Note: *You must use PortVision DX or Redboot to initially configure an IP address for your network. Refer to the Hardware Installation and Configuration for procedures.*

Use the following procedure to change the network configuration.

1. Click the **Network** menu, which by default opens the *Network Configuration* page.
2. If desired, enter a **Device Name** for this DeviceMaster.
3. If necessary, change the **TCP Keepalive** value.
4. Leave the default **Boot Timeout** unless you need to change it for compatibility with spanning tree devices (switches).
5. Select the IP configuration type (**DHCP** or **Static**).
6. If you select **Static**, enter a valid IP address, subnet mask, and IP gateway for your network. The network information is programmed into the DeviceMaster after applying the changes and rebooting the device. If necessary, see your network administrator for a valid IP address.

   Note: *The DeviceMaster family default IP address is 192.168.250.250, default subnet mask is 255.255.0.0, and the default IP gateway is 192.168.250.1.*

7. Click **System | Reboot** and then click the **Reboot Now** button.



Reboot

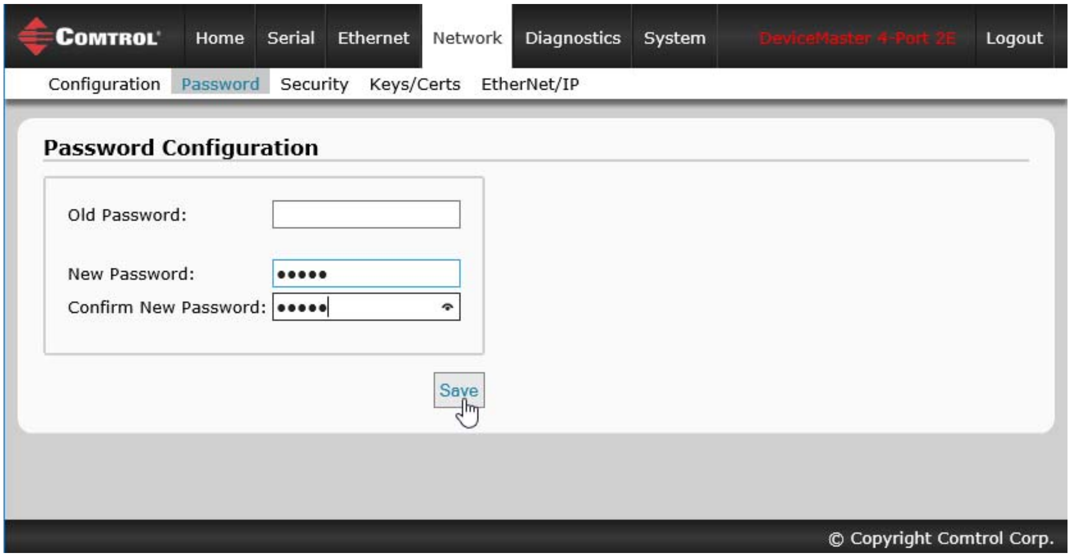
Rebooting will drop all connections, apply any pending settings, and restart the DeviceMaster.

Note: *Changed network settings do not take effect until the DeviceMaster is rebooted.*

4.10. Password Configuration Page

You can easily set up a password to secure the DeviceMaster.

Note: *There is no password set from the factory.*



Password Configuration

Old Password:

New Password: •••••

Confirm New Password: •••••

Use the following information to configure a password for this DeviceMaster.

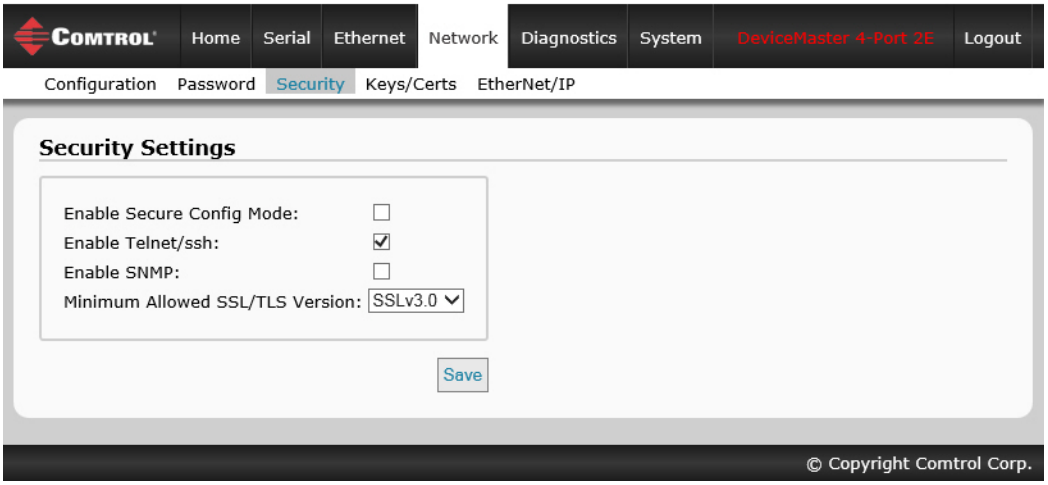
1. If necessary, click **Network | Password**.
2. If changing an existing password, enter that password in the **Old Password** field.
3. Enter a new password.
4. Enter the password in the **Confirm New Password** field.
5. Click the **Save** button.

To log into the DeviceMaster, you must enter the following:

- **admin** for the user name
- The configured password for the password.

4.11. Security Settings Page

You can use the *Network |Security Settings* page to configure security on the DeviceMaster.



Security Settings

Enable Secure Config Mode: □
Enable Telnet/ssh: ☑
Enable SNMP: □
Minimum Allowed SSL/TLS Version: SSLv3.0

Network | Security Settings Page	
Enable Secure Data Mode (Default = Disabled)	If **Secure Data Mode** is enabled, TCP connections that carry data to/from the serial ports are encrypted using SSL or TLS security protocols. This includes the following: • TCP connections to the per-serial-port TCP ports (default is 8000, 8001, 8002, ...) are encrypted using SSL/TLS. • TCP connections to TCP port 4606 on which the DeviceMaster implements the Comtrol proprietary protocol are encrypted using SSL/TLS. • In addition to encrypting the data streams, it is possible to configure the DeviceMaster so that only authorized client applications can connect using SSL/TLS. See *4.11.1. Client Authentication* on Page 125 for more information.
Enable Telnet/ssh (Default = Enabled)	This option enables or disables the telnet security feature after you click **Save** and the DeviceMaster has been rebooted.
Enable SNMP (Default = Disabled)	This option enables or disables the SNMP security feature after you click **Save** and the DeviceMaster has been rebooted.
Minimum Allowed SSL/TLS Version	You can select the appropriate version for your environment. • SSLv3.0 • TLSv1.0 (default) • TLSv1.1 • TLSv1.2

4.11.1. Client Authentication

If desired, controlled access to SSL/TLS protected features can be configured by uploading a client authentication certificate to the DeviceMaster. By default, the DeviceMaster is shipped without a CA (Certificate Authority) and therefore allows connections from any SSL/TLS client.

If a CA certificate is uploaded, the DeviceMaster only allows SSL/TLS connections from client applications that provide to the DeviceMaster an identity certificate that has been signed by the CA certificate that was uploaded to the DeviceMaster.

This uploaded CA certificate that is used to validate a client's identity is sometimes referred to as a *trusted root certificate*, a *trusted authority certificate*, or a *trusted CA certificate*. This CA certificate might be that of a trusted commercial certificate authority or it may be a privately generated certificate that an organization creates internally to provide a mechanism to control access to resources that are protected by the SSL/TLS protocols.

To control access to the DeviceMaster's SSL/TLS protected resources you should create your own custom CA certificate and then configure authorized client applications with identity certificates signed by the custom CA certificate.

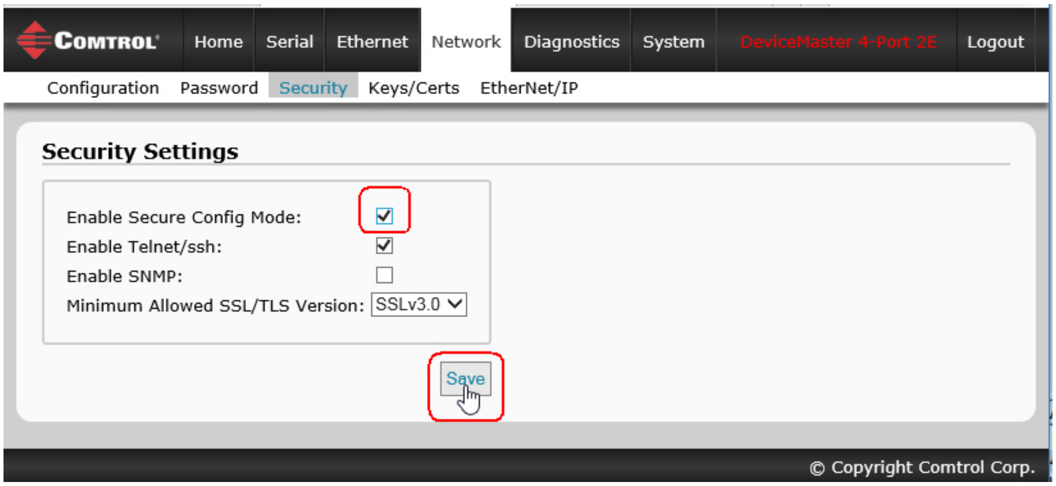
4.11.2. Configuring DeviceMaster Security

Use the following procedure to configure DeviceMaster security settings.

Note: *All DeviceMaster units are shipped from the factory with identical configurations. They all have the identical, self-signed, Comtrol Server RSA Certificates, Server RSA Keys, Server DH Keys, and no Client Authentication Certificates.*

For maximum data and access security, you should configure all DeviceMaster units with custom certificates and keys.

1. If necessary, access the *Network | Security Settings* page by entering the DeviceMaster IP address in your web browser or by highlighting the DeviceMaster in PortVision DX and clicking **Webpage.**
2. Enable the **Secure Config Mode** option.



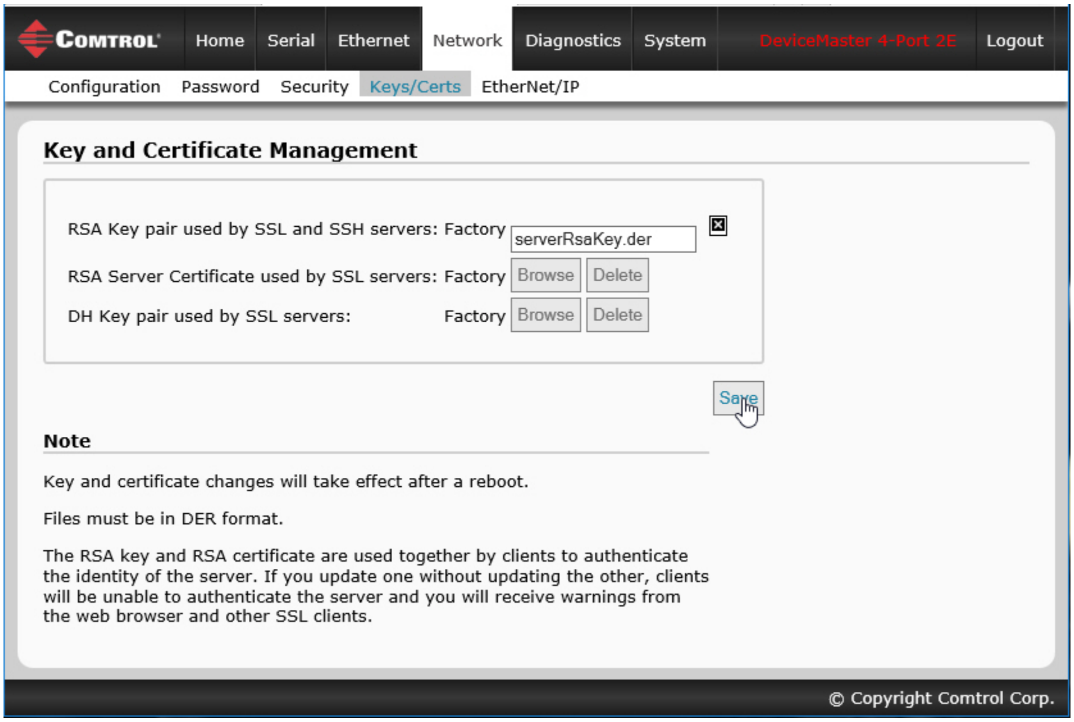
Security Settings

Enable Secure Config Mode: ☑
Enable Telnet/ssh: ☑
Enable SNMP: □
Minimum Allowed SSL/TLS Version: SSLv3.0

3. If desired, disable the **Telnet/ssh** option.
4. If desired, enable the **Enable SNMP** option.
5. If necessary, change the **Minimum Allowed SSL/TLS Version**.
6. Click the **Save** button.
7. Go to the next subsection to manage security keys and certificates.

4.12. Keys and Certificate Management Page

Use the *Network* | *Key and Certificate Management* page to configure your security keys and certificates.



Key and Certificate Management

RSA Key pair used by SSL and SSH servers: Factory serverRsaKey.der ☒

RSA Server Certificate used by SSL servers: Factory

DH Key pair used by SSL servers: Factory

Note

Key and certificate changes will take effect after a reboot.

Files must be in DER format.

The RSA key and RSA certificate are used together by clients to authenticate the identity of the server. If you update one without updating the other, clients will be unable to authenticate the server and you will receive warnings from the web browser and other SSL clients.

Key and Certificate Management Page	
RSA Key pair used by SSL and SSH servers	This is a private/public key pair that is used for two purposes: • It is used by some cipher suites to encrypt the SSL/TLS handshaking messages. Possession of the private portion of this key pair allows an eavesdropper to both decrypt traffic on SSL/TLS connections that use RSA encryption during handshaking. • It is used to sign the Server RSA Certificate in order to verify that the DeviceMaster is authorized to use the server RSA identity certificate. ***Note:*** *Possession of the private portion of this key pair allows somebody to pose as the* DeviceMaster. If the Server RSA Key is to be replaced, a corresponding RSA identity certificate must also be generated and uploaded or clients are not able to verify the identity certificate.
RSA Server Certificate used by SSL servers	This is the RSA identity certificate that the DeviceMaster uses during SSL/TLS handshaking to identify itself. It is used most frequently by SSL server code in the DeviceMaster when clients open connections to the DeviceMaster's secure web server or other secure TCP ports. If a DeviceMaster serial port configuration is set up to open (as a client) a TCP connection to another server device, the DeviceMaster also uses this certificate to identify itself as an SSL client if requested by the server. In order to function properly, this certificate must be signed using the Server RSA Key. This means that the server RSA certificate and server RSA key must be replaced as a pair.

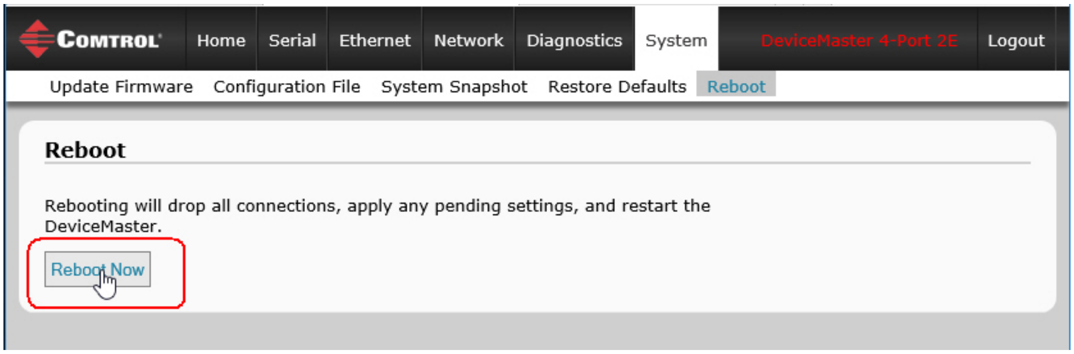
Key and Certificate Management Page	
DH Key pair used by SSL servers	This is a private/public key pair that is used by some cipher suites to encrypt the SSL/TLS handshaking messages. ***Note:*** *Possession of the private portion of the key pair allows an eavesdropper to decrypt traffic on SSL/TLS connections that use DH encryption during handshaking.*

Use the following procedure to configure your security keys and certificates.

1. Click **Network | Keys/Certs**.
2. Click **Browse** to locate the key or certificate file, highlight the file, and click **Open**.

   Note: *The key or certificate notation changes from factory or none to User when the DeviceMaster is secure.*
3. Click the **Save** button.

   Note: *Changes do not take effect until the DeviceMaster is rebooted.*
4. Click **System | Reboot** and then click the **Reboot Now** button.

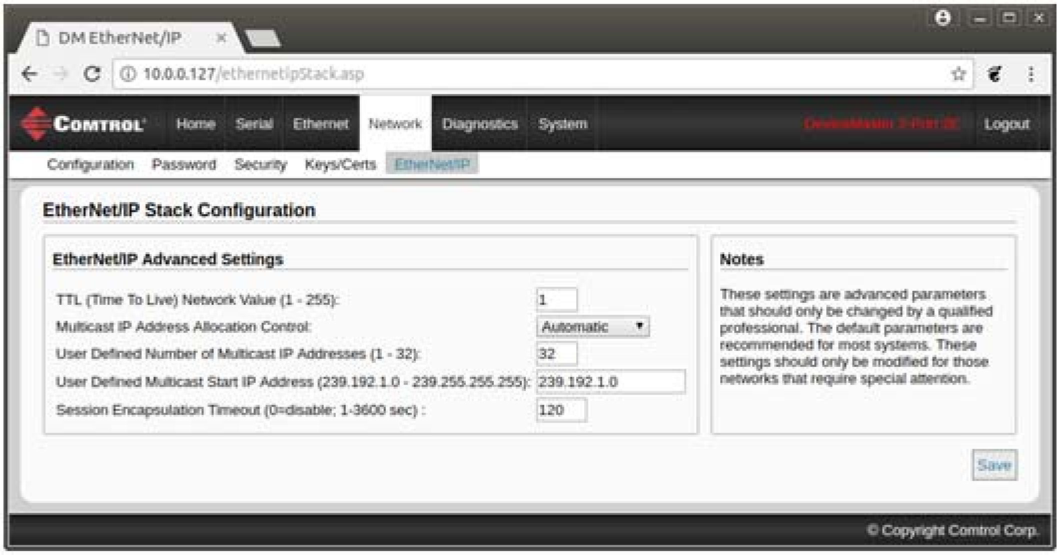


Reboot

Rebooting will drop all connections, apply any pending settings, and restart the DeviceMaster.

4.13. EtherNet/IP Stack Configuration

The EtherNet/IP stack settings are very advanced settings that should only be changed if required and should only be done by a qualified professional.



EtherNet/IP Stack Configuration

EtherNet/IP Advanced Settings

TTL (Time To Live) Network Value (1 - 255): 1

Multicast IP Address Allocation Control: Automatic

User Defined Number of Multicast IP Addresses (1 - 32): 32

User Defined Multicast Start IP Address (239.192.1.0 - 239.255.255.255): 239.192.1.0

Session Encapsulation Timeout (0=disable; 1-3600 sec) : 120

Notes

These settings are advanced parameters that should only be changed by a qualified professional. The default parameters are recommended for most systems. These settings should only be modified for those networks that require special attention.

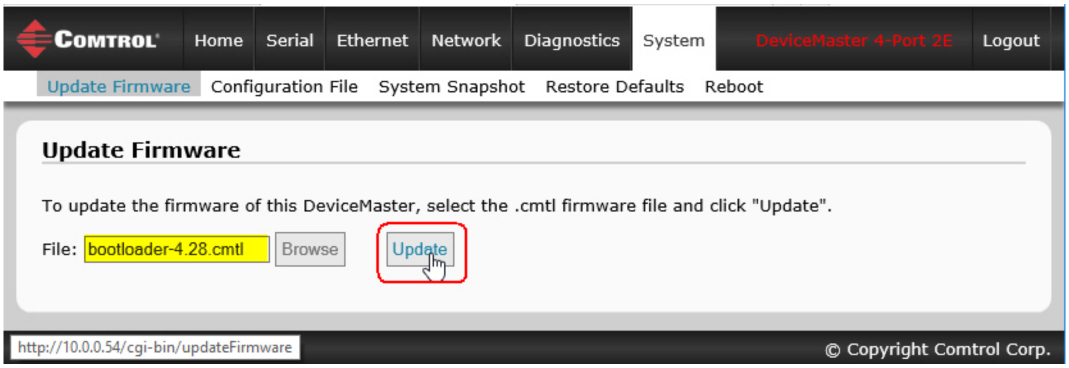
EtherNet/IP Stack Configuration Page	
TTL (Time To Live) Network Value	This is the network routing parameter that allows EtherNet/IP messages to be routed between different subnets. A TTL of 1, the default, allows the messages to be routed over one network “hop”. A TTL allows two hops and so on. Default = 1. This setting generally allows messages to be routed on the same subnet.
Multicast IP Address Allocation Control	This defines how the multi-cast addresses, which are used in Class1 communications, will be allocated. • **Automatic** – This instructs the DeviceMaster gateway to use the standard automatic multicast address mechanism. • **User Defined** – This instructs the DeviceMaster gateway to allocate the multicast addresses based on the user settings. Default: Automatic
User Defined Number of Multicast IP Addresses	If Multicast IP address allocation control is set to User Defined, this setting instructs the DeviceMaster gateway to use the configured number of multicast IP addresses.
User Defined Multicast Start IP Address	If Multicast IP address allocation control is set to User Defined, this setting instructs the DeviceMaster gateway where to start its multicast IP address range.
Session Encapsulation Timeout	The session encapsulation timeout instructs the DeviceMaster gateway how long to wait with no activity before timing out a session.
Save	Make sure that you click the **Save** button before leaving the page if you want the changes saved.

4.14. Update Firmware Page

You can upload firmware (EtherNet/IP or Bootloader) using the *System | Update Firmware* page.

Use the following procedure to upload the latest firmware onto the DeviceMaster.

1. If necessary, download the firmware from the Comtrol download site.
2. Click the System menu, which defaults to the *Update Firmware* page.
3. Click the **Browse** button, navigate to the file and select it.
4. Click the **Update** button.



Update Firmware

To update the firmware of this DeviceMaster, select the .cmtl firmware file and click "Update".

File: bootloader-4.28.cmtl

The web page provides a status screen and will reboot the DeviceMaster.

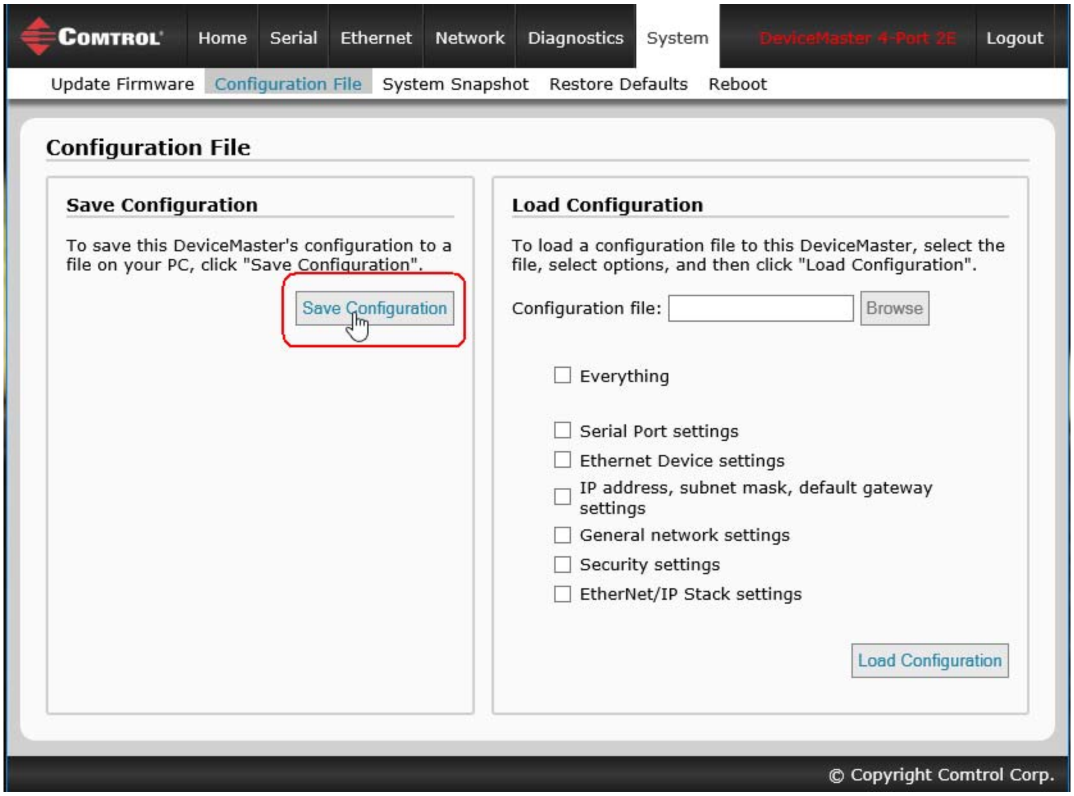
4.15. Configuration File Page

You can use the *Save Configuration* option to save a DeviceMaster configuration file for recovery purposes or to quickly configure other DeviceMasters that require the same configuration using the **Load Configuration** option.

4.15.1. Saving a Configuration File

You can use this procedure to save a DeviceMaster configuration file.

1. Click **System | Configuration File**.
2. Click the **Save Configuration** button.
3. Save the configuration file using the method provided by your browser.



Configuration File

Save Configuration

To save this DeviceMaster's configuration to a file on your PC, click "Save Configuration".

Load Configuration

To load a configuration file to this DeviceMaster, select the file, select options, and then click "Load Configuration".

Configuration file:

□ Everything

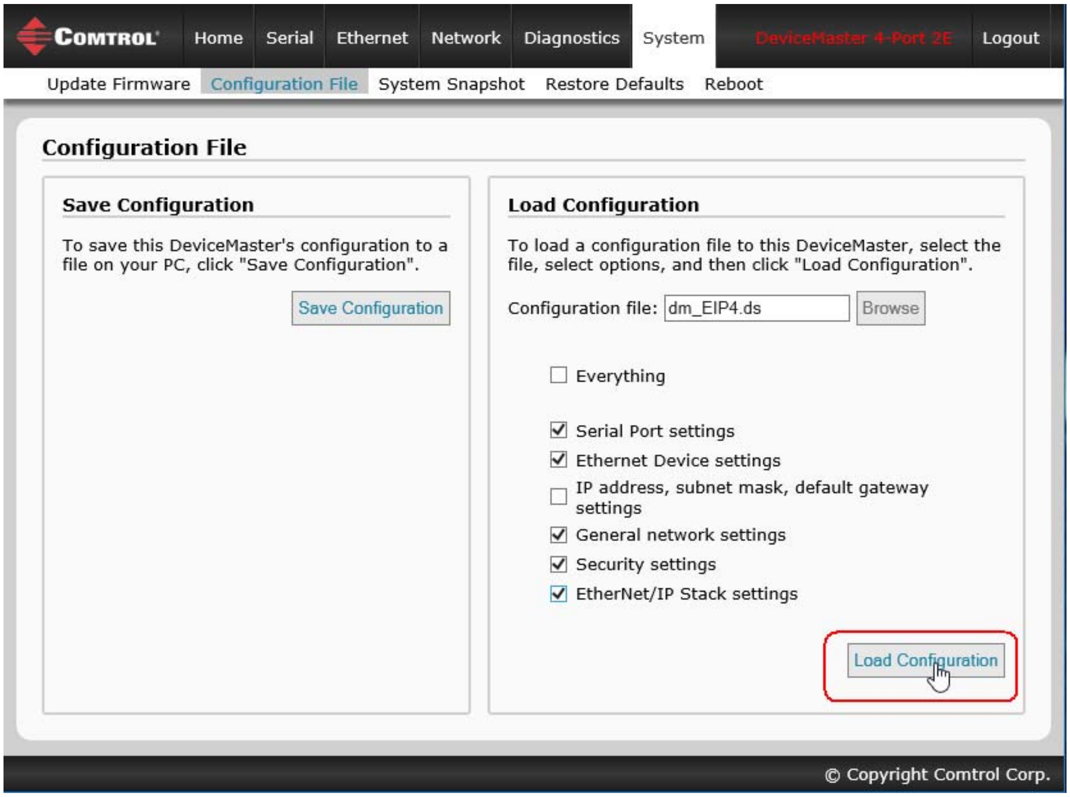
□ Serial Port settings
□ Ethernet Device settings
□ IP address, subnet mask, default gateway settings
□ General network settings
□ Security settings
□ EtherNet/IP Stack settings

© Copyright Comtrol Corp.

4.15.2. Loading a Configuration File

You can use this procedure to load a previously saved DeviceMaster configuration file.

1. Click **System | Configuration File**.
2. Select the items that you want to load.
3. Click the **Browse** button, highlight the configuration file you want to load, and click the **Open** button.
4. Click the **Load Configuration** button.



Configuration File

Save Configuration

To save this DeviceMaster's configuration to a file on your PC, click "Save Configuration".

Load Configuration

To load a configuration file to this DeviceMaster, select the file, select options, and then click "Load Configuration".

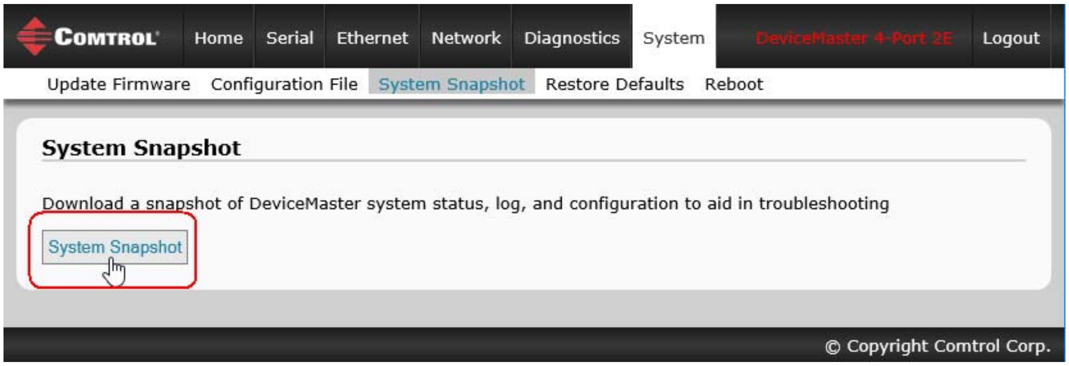
Configuration file: dm_EIP4.ds

☐ Everything

☑ Serial Port settings
☑ Ethernet Device settings
☐ IP address, subnet mask, default gateway settings
☑ General network settings
☑ Security settings
☑ EtherNet/IP Stack settings

4.16. System Snapshot Page

Use the *System | System Snapshot* page to download a snapshot of the DM status, log and DeviceMaster configuration. This information may be requested by Technical Support to provide configuration and status data for troubleshooting an issue.



System Snapshot

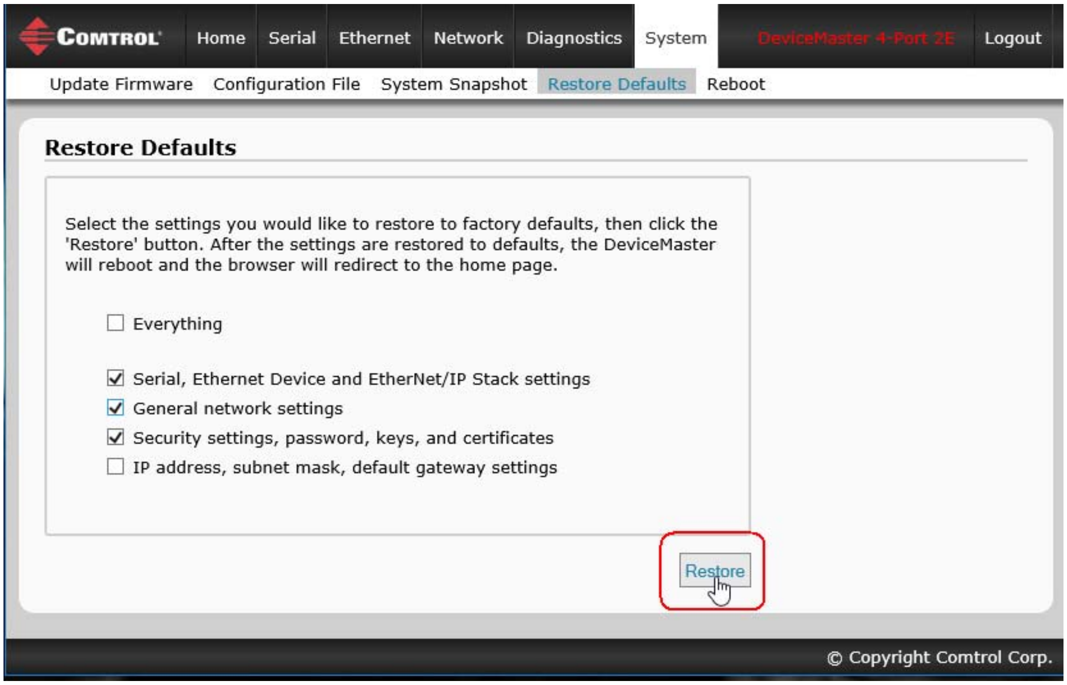
Download a snapshot of DeviceMaster system status, log, and configuration to aid in troubleshooting

4.17. Restore Defaults Page

You can use the *System | Restore Defaults* page to restore all settings or a combination of the following:

- Everything
- Serial, Ethernet Device and EtherNet/IP Stack settings
- General network settings
- Security settings, password, keys, and certificates
- IP address, subnet mask, default gateway settings

To restore the defaults, click the settings that you want to reset and click the **Restore** button.



Restore Defaults

Select the settings you would like to restore to factory defaults, then click the 'Restore' button. After the settings are restored to defaults, the DeviceMaster will reboot and the browser will redirect to the home page.

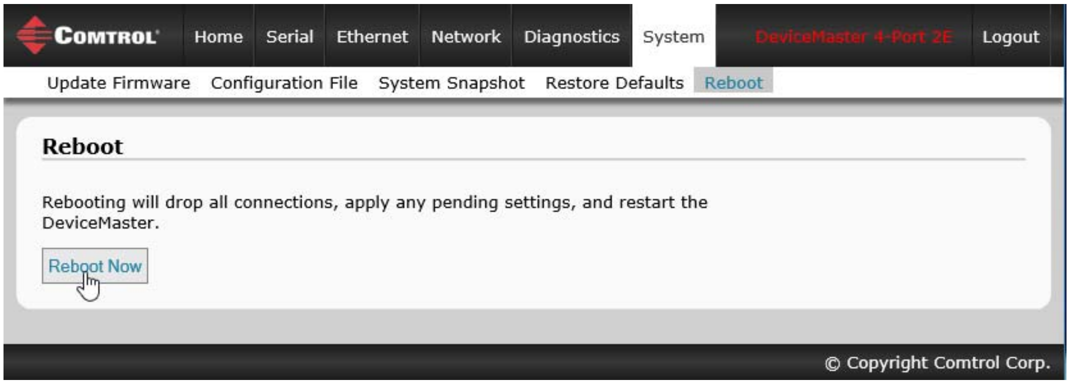
☐ Everything

☑ Serial, Ethernet Device and EtherNet/IP Stack settings
☑ General network settings
☑ Security settings, password, keys, and certificates
☐ IP address, subnet mask, default gateway settings

4.18. Reboot Page

You can use PortVision DX or the Reboot web page to reboot your DeviceMaster.

1. Click **System | Reboot**.
2. Click the **Reboot Now** button to reboot the DeviceMaster.



Reboot

Rebooting will drop all connections, apply any pending settings, and restart the DeviceMaster.

Chapter 5. Diagnostic and Statistics Pages

You can access the following diagnostic and statistics pages from the *Diagnostics* menu:

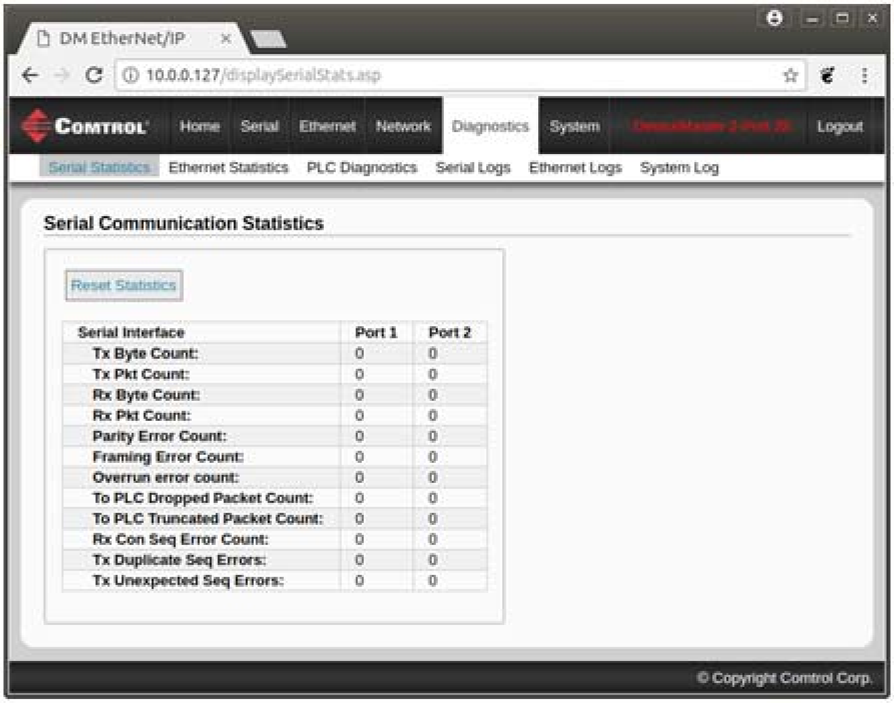
This section discusses the following pages:

- *5.1. Serial Communication Statistics Page* on Page 135
- *5.2. Ethernet Device Statistics Page* on Page 138
- *5.3. PLC Diagnostics (EtherNet/IP Interface Statistics) Page* on Page 140
- *5.4. Serial Interface Logs* on Page 143
- *5.5. Ethernet Device Logs* on Page 144
- *5.6. System Log* on Page 144

5.1. Serial Communication Statistics Page

The default *Diagnostics* menu page is the *Serial Communication Statistics* page.

The displayed counters are the same as those specified in *3.2.3. Serial Port Statistics Object Definition (72 Hex)* on Page 35. Refer to the following table for definitions of the fields.



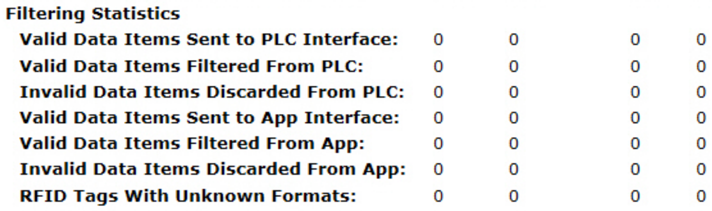
Serial Communication Statistics

Serial Interface	Port 1	Port 2
Tx Byte Count:	0	0
Tx Pkt Count:	0	0
Rx Byte Count:	0	0
Rx Pkt Count:	0	0
Parity Error Count:	0	0
Framing Error Count:	0	0
Overrun error count:	0	0
To PLC Dropped Packet Count:	0	0
To PLC Truncated Packet Count:	0	0
Rx Con Seq Error Count:	0	0
Tx Duplicate Seq Errors:	0	0
Tx Unexpected Seq Errors:	0	0

Serial Communications Statistics	
Reset Statistics	This button clears the serial port statistics, which sets the value to 0 for all ports.
Tx Byte Count	Displays the number of bytes sent out of the serial port.
Tx Pkt Count	Displays the number of serial packets sent out of the serial port.
Rx Byte Count	Displays the number of bytes received over the serial port.
Rx Pkt Count	Displays the number of packets received over the serial port.
Parity Error Count	Displays the number of received serial packets dropped due to parity errors.
Framing Error Count	Displays the number of received serial packets dropped due to framing errors.
Overrun Error Count	Displays the number of received serial packets dropped due to overrun error incidents.
To PLC Dropped Packet Count	Displays the number of received serial packets intended for the PLC dropped: • No STX byte(s) found • No ETX byte(s) found • Time-outs • Packet to large • Receive buffer queue overflows
To PLC Truncated Packet Count	Displays the number of received packets that were truncated before being sent to the PLC.
Rx Con Seq Errors Count	Displays the number of received consumed sequenced number errors. The DeviceMasteronly increments this number when all of the following statements are true: • You selected the Write-to-Tag-Sync method of receive communication. • DeviceMaster receives a serial packet. • The Consumed Sequence number is out of sync. (It is not equal to the Produced Sequence number or equal to the Produced Sequence number minus one.)
Tx Duplicate Seq Errors	Displays the number of *Duplicate Transmit Sequence Number* errors. The DeviceMaster increments this number when the following statements are true: • You enabled the *Transmit Sequence Number Checking configuration* option. (See *Attribute 16 - Serial Port Transfer Options* on Page 27 for additional information.) • DeviceMaster receives a transmit message with a sequence number equal to the previous sequence number. (The DeviceMaster expects this sequence number to be incremented by one from the sequence number in the previous transmit message.)
Tx Unexpected Seq Errors	Displays the number of *Unexpected Transmit Sequence Number* errors. The DeviceMaster increments this number when the following statements are true: • You enabled the *Transmit Sequence Number Checking configuration* option. (See *Attribute 16 - Serial Port Transfer Options* on Page 27 for additional information.) • DeviceMaster receives a transmit message with a sequence number that is not equal to either the previous transmit sequence number or the previous transmit sequence number plus one. (The DeviceMaster expects this sequence number to be incremented by one with each new transmit message.)

Note: *Filtering Statistics are displayed only if filtering is enabled on or more serial ports.*

Filtering Statistics (Serial)

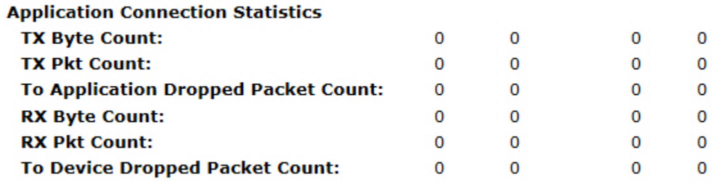


```
Filtering Statistics
  Valid Data Items Sent to PLC Interface:   0     0        0     0
  Valid Data Items Filtered From PLC:       0     0        0     0
  Invalid Data Items Discarded From PLC:    0     0        0     0
  Valid Data Items Sent to App Interface:   0     0        0     0
  Valid Data Items Filtered From App:       0     0        0     0
  Invalid Data Items Discarded From App:    0     0        0     0
  RFID Tags With Unknown Formats:           0     0        0     0
```

Valid Data Items Sent To PLC Interface	Displays the number of valid string, RFID, or barcode data sent to the PLC. Applies when filtering is enabled.
Valid Data Items Filtered From PLC	Displays the number of valid string, RFID, or barcode data filtered from (not sent) to the PLC. Applies when filtering is enabled.
Invalid Data Items Discarded From PLC	Displays the number of invalid RFID or barcode data not sent to the PLC. Applies when RFID or barcode filtering is enabled.
Valid Data Items Sent To App Interface	Displays the number of valid string, RFID, or barcode data sent to the application. Applies when filtering is enabled.
Valid Data Items Filtered From App	Displays the number of valid string, RFID, or barcode data filtered from (not sent) to the application. Applies when filtering is enabled.
Invalid Data Items Discarded From Application	Displays the number of invalid RFID or barcode data not sent to the PLC. Applies when RFID or barcode filtering is enabled.
RFID Tags With Unknown Formats	Data received that was in the general form of 64 or 96 bit RFID tags, but was not in any of the EPCglobal formats. Applies only when RFID filtering is enabled.

Note: *Connection Statistics are shown only if the Application connection is enabled on one or more serial ports.*

Application Connection Statistics (Serial)

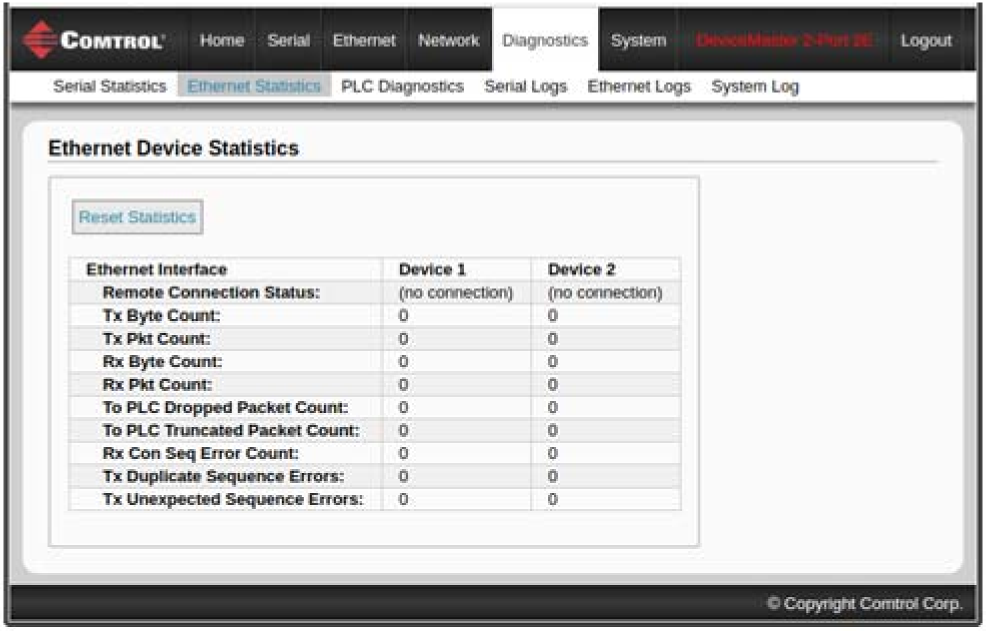


```
Application Connection Statistics
  TX Byte Count:                          0     0        0     0
  TX Pkt Count:                           0     0        0     0
  To Application Dropped Packet Count:    0     0        0     0
  RX Byte Count:                          0     0        0     0
  RX Pkt Count:                           0     0        0     0
  To Device Dropped Packet Count:         0     0        0     0
```

TX Byte Count	Displays the number of bytes sent out the application socket port.
TX Pkt Count	Displays the number of packets sent out the application socket port.
To Application Dropped Packet Count	Displays the number of received serial or Ethernet device packets intended for the application dropped: • No STX byte(s) found • No ETX byte(s) found • Time-outs • Packet to large • Receive buffer queue overflows • Application connection is offline
RX Byte Count	Displays the number of bytes received over the application socket port.
RX Pkt Count	Displays the number of packets received over the application socket port.
To Device Dropped Packet Count	Displays the number of dropped packets that were intended for the device.

5.2. Ethernet Device Statistics Page

Access the *Ethernet Device Statistics* page by clicking **Diagnostics | Ethernet Statistics**.



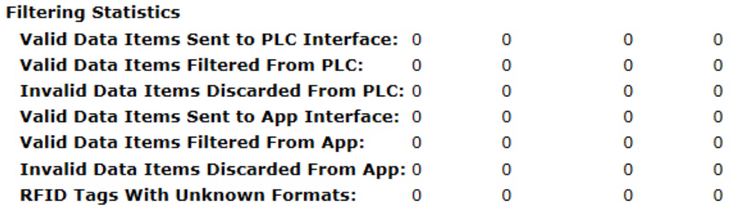
Ethernet Device Statistics

Ethernet Interface	Device 1	Device 2
Remote Connection Status:	(no connection)	(no connection)
Tx Byte Count:	0	0
Tx Pkt Count:	0	0
Rx Byte Count:	0	0
Rx Pkt Count:	0	0
To PLC Dropped Packet Count:	0	0
To PLC Truncated Packet Count:	0	0
Rx Con Seq Error Count:	0	0
Tx Duplicate Sequence Errors:	0	0
Tx Unexpected Sequence Errors:	0	0

Ethernet Device Statistics	
Reset Statistics	This button clears the socket port statistics, which sets the value to 0 for all ports.
Tx Byte Count	Displays the number of bytes sent out the device socket port.
Tx Pkt Count	Displays the number of packets sent out the device socket port.
Rx Byte Count	Displays the number of bytes received over the device socket port.
Rx Pkt Count	Displays the number of packets received over the device socket port.
To PLC Dropped Packet Count	Displays the number of dropped packets that were intended for the PLC.
To PLC Truncated Packet Count	Displays the number of received packets that were truncated before being sent to the PLC.
Rx Con Sequence Error Count	Same as the serial port statistics, except the packet was received over a socket port (Page 136).
Tx Duplicate Sequence Errors	Same as the serial port statistics (Page 136).
Tx Unexpected Sequence Errors	Same as the serial port statistics (Page 136).

Filtering Statistics are displayed only if filtering is enabled on or more Ethernet device interfaces.

Filtering Statistics (Ethernet Device Interface Statistics)

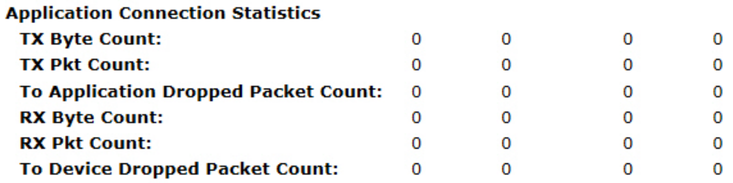


```
Filtering Statistics
  Valid Data Items Sent to PLC Interface:  0      0      0      0
  Valid Data Items Filtered From PLC:      0      0      0      0
  Invalid Data Items Discarded From PLC:   0      0      0      0
  Valid Data Items Sent to App Interface:  0      0      0      0
  Valid Data Items Filtered From App:      0      0      0      0
  Invalid Data Items Discarded From App:   0      0      0      0
  RFID Tags With Unknown Formats:          0      0      0      0
```

Valid Data Items Sent To PLC Interface	Displays the number of valid string, RFID, or barcode data sent to the PLC. Applies when filtering is enabled.
Valid Data Items Filtered From PLC	Displays the number of valid string, RFID, or barcode data filtered from (not sent) to the PLC. Applies when filtering is enabled.
Invalid Data Items Discarded From PLC	Displays the number of invalid RFID or barcode data not sent to the PLC. Applies when RFID or barcode filtering is enabled.
Valid Data Items Sent To App Interface	Displays the number of valid string, RFID, or barcode data sent to the application. Applies when filtering is enabled.
Valid Data Items Filtered From App	Displays the number of valid string, RFID, or barcode data filtered from (not sent) to the application. Applies when filtering is enabled.
Invalid Data Items Discarded From Application	Displays the number of invalid RFID or barcode data not sent to the PLC. Applies when RFID or barcode filtering is enabled.
RFID Tags With Unknown Formats	Data received that was in the general form of 64 or 96 bit RFID tags, but was not in any of the EPCglobal formats. Applies only when RFID filtering is enabled.

Application Connection Statistics are displayed only if an Application connection is enabled on or more Ethernet device interfaces.

Application Connection Statistics (Ethernet Device Interface Statistics)

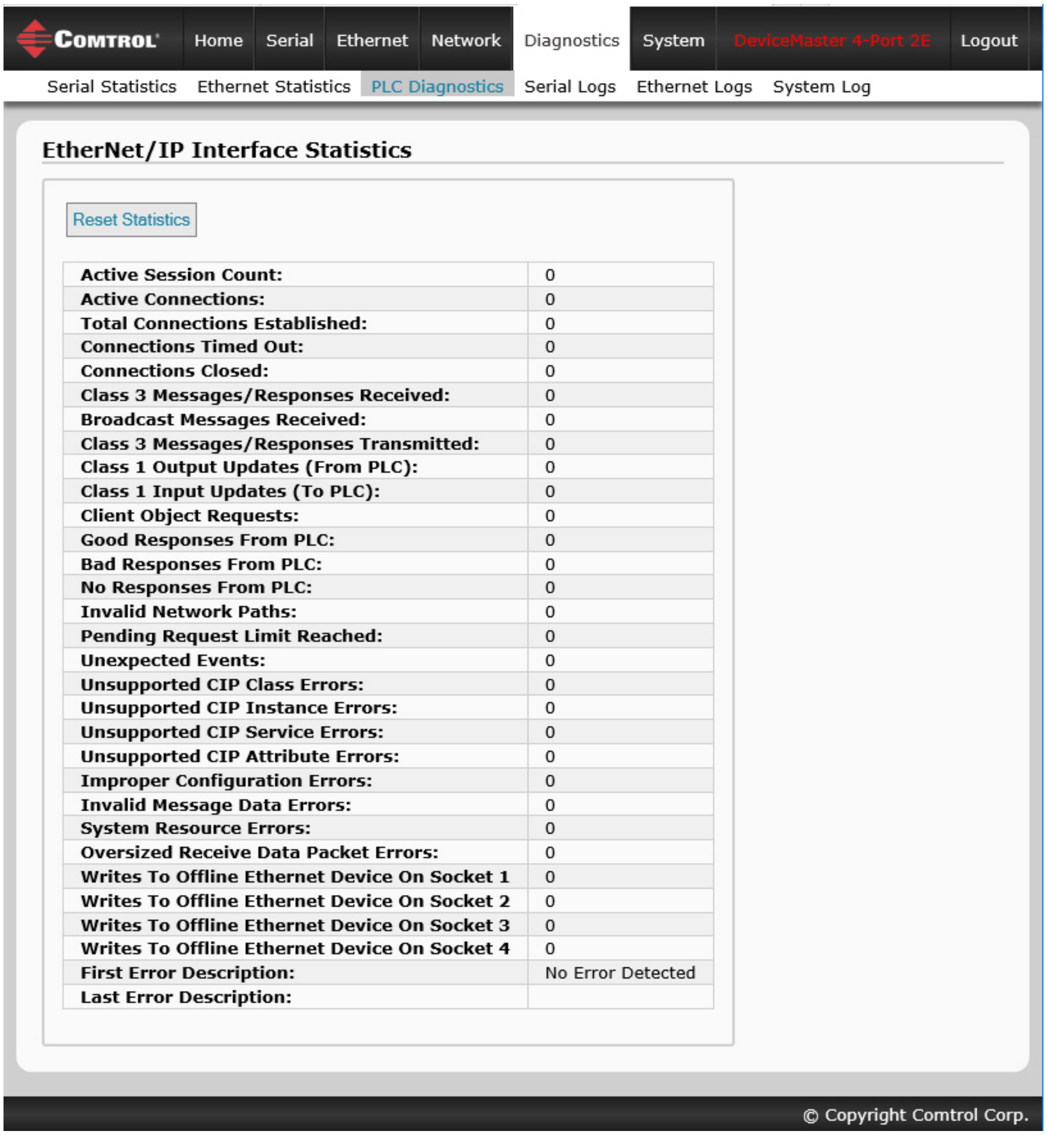


```
Application Connection Statistics
  TX Byte Count:                         0      0      0      0
  TX Pkt Count:                          0      0      0      0
  To Application Dropped Packet Count:   0      0      0      0
  RX Byte Count:                         0      0      0      0
  RX Pkt Count:                          0      0      0      0
  To Device Dropped Packet Count:        0      0      0      0
```

TX Byte Count	Displays the number of bytes sent out the application socket port.
TX Pkt Count	Displays the number of packets sent out the application socket port.
To Application Dropped Packet Count	Displays the number of received serial or Ethernet device packets intended for the application dropped: • No STX byte(s) found • No ETX byte(s) found • Time-outs • Packet to large • Receive buffer queue overflows • Application connection is offline
RX Byte Count	Displays the number of bytes received over the application socket port.
RX Pkt Count	Displays the number of packets received over the application socket port.
To Device Dropped Packet Count	Displays the number of dropped packets that were intended for the device.

5.3. PLC Diagnostics (EtherNet/IP Interface Statistics) Page

Access the *EtherNet/IP Interface Statistics* page by clicking **Diagnostics | PLC Diagnostics**.



EtherNet/IP Interface Statistics

Active Session Count:	0
Active Connections:	0
Total Connections Established:	0
Connections Timed Out:	0
Connections Closed:	0
Class 3 Messages/Responses Received:	0
Broadcast Messages Received:	0
Class 3 Messages/Responses Transmitted:	0
Class 1 Output Updates (From PLC):	0
Class 1 Input Updates (To PLC):	0
Client Object Requests:	0
Good Responses From PLC:	0
Bad Responses From PLC:	0
No Responses From PLC:	0
Invalid Network Paths:	0
Pending Request Limit Reached:	0
Unexpected Events:	0
Unsupported CIP Class Errors:	0
Unsupported CIP Instance Errors:	0
Unsupported CIP Service Errors:	0
Unsupported CIP Attribute Errors:	0
Improper Configuration Errors:	0
Invalid Message Data Errors:	0
System Resource Errors:	0
Oversized Receive Data Packet Errors:	0
Writes To Offline Ethernet Device On Socket 1	0
Writes To Offline Ethernet Device On Socket 2	0
Writes To Offline Ethernet Device On Socket 3	0
Writes To Offline Ethernet Device On Socket 4	0
First Error Description:	No Error Detected
Last Error Description:	

EtherNet/IP Interface Statistics	
Active Session Count	The number of active Ethernet/IP sessions. A session can: • Support both Class 1 I/O and Class 3 Messages • Can be initiated by either the PLC or the GW EIP/ASCII • Can be terminated by either the PLC or the GW EIP/ASCII
Active Connections	The current number of active connections (both Class 1 and 3).
Total Connections Established	The total number of connections that have been established.
Connection Timed Out	The number of connections that have closed due to timing out.

EtherNet/IP Interface Statistics	
Connections Closed	The number connections that have closed due to a standard processes.
Class3 Messages/ Responses Received	Displays the number of Class3 messages and responses received from the PLC(s).
Broadcasts Messages Received	Displays the number of broadcast messages received from the PLC(s).
Class 3 Messages/ Responses Transmitted	Displays the number of messages and responses sent to the PLC(s).
Class 1 Output Updates (From PLC)	The number of Class 1 output data updates received from the PLC or PLCs.
Class 1 Input Updates (To PLC)	The number of Class 1 input data updates sent to the PLC or PLCs.
Client Objects Requests	Displays the number of Class3 request messages received from the PLC(s).
Good Responses From PLC	The number of good responses received from the PLC.
Bad Responses From PLC	Displays the number of bad responses from messages sent to the PLC(s). Bad responses are typically returned for such errors as: • Incorrect tag or file names • Incorrect tag or file data types • Incorrect tag or file data sizes • PLC is overloaded and cannot handle the amount of Ethernet traffic • PLC malfunction
No Responses From PLC	Displays the number of no responses from messages sent to the PLC(s). No responses are typically returned for such errors as: • Incorrect IP address • Incorrect PLC configuration • PLC malfunction • PLC is overloaded and cannot handle the amount of Ethernet traffic
Invalid Network Paths	Displays the number of network path errors on messages sent to the PLC(s). These are typically caused by incorrect IP address settings.
Pending Request Limit Reached	Displays the number of pending request limit errors. These errors occur when the PLC is sending a continuous stream of messages to the DeviceMaster faster than the DeviceMaster can process them.
Unexpected Events	Displays the number of unexpected event errors. Unexpected event errors occur when the DeviceMaster receives an unexpected message from the PLC such as an unexpected response or unknown message.
Unsupported CIP Class Errors	Displays the number of unsupported CIP request instance errors. These errors occur when a message with an invalid class is sent to the DeviceMaster.
Unsupported CIP Instance Errors	Displays the number of unsupported CIP request instance errors. These errors occur when a message with an invalid instance is sent to the DeviceMaster.
Unsupported CIP Service Errors	Displays the number of unsupported CIP request instance errors. These errors occur when a message with an invalid service is sent to the DeviceMaster.
Unsupported CIP Attribute Errors	Displays the number of unsupported CIP request instance errors. These errors occur when a message with an invalid attribute is sent to the DeviceMaster.
Improper Configuration Errors	Displays the number of improper configuration errors. These errors occur when the DeviceMaster receives a message that cannot be performed due to an invalid configuration.

EtherNet/IP Interface Statistics	
Invalid Message Data Errors	Displays the number of invalid message data errors. These errors occur when the DeviceMaster receives a message that cannot be performed due to invalid data.
System Resource Errors	Displays the number of system resource errors. These errors indicate a system error on the DeviceMaster such as an inoperable serial port or a full transmit queue. These errors typically occur when the PLC(s) are sending data to the DeviceMaster faster than the DeviceMaster can process it.
Oversized Receive Data Packet Errors	Displays the number of received Ethernet data packets that were larger than the configured maximum receive data packet.
Writes To Offline Ethernet Device On Socket *N*	Displays the number of write attempts by a PLC to the Ethernet device when the device was offline.
First Error Description	Text description of the first error that occurred.
Last Error Description	Text description of the last or most recent error that occurred.

5.4. Serial Interface Logs

Access the *Serial Interface Logs* page by clicking **Diagnostics | Serial Logs**.

The *Serial Interface Logs* page provides a log of received and transmitted serial port messages. Up to 128 bytes per message and up to 128 messages are logged. It is intended to help with debugging serial connectivity problems, determining the proper start and end of transmission bytes, and diagnosing device problems.

The format is as follows:

```
Pkt(n): ddd:hh:mm:ss.ms Tx/Rx:<Data>
```

Where:

ddd – days since last system restart

hh – hours since last system restart

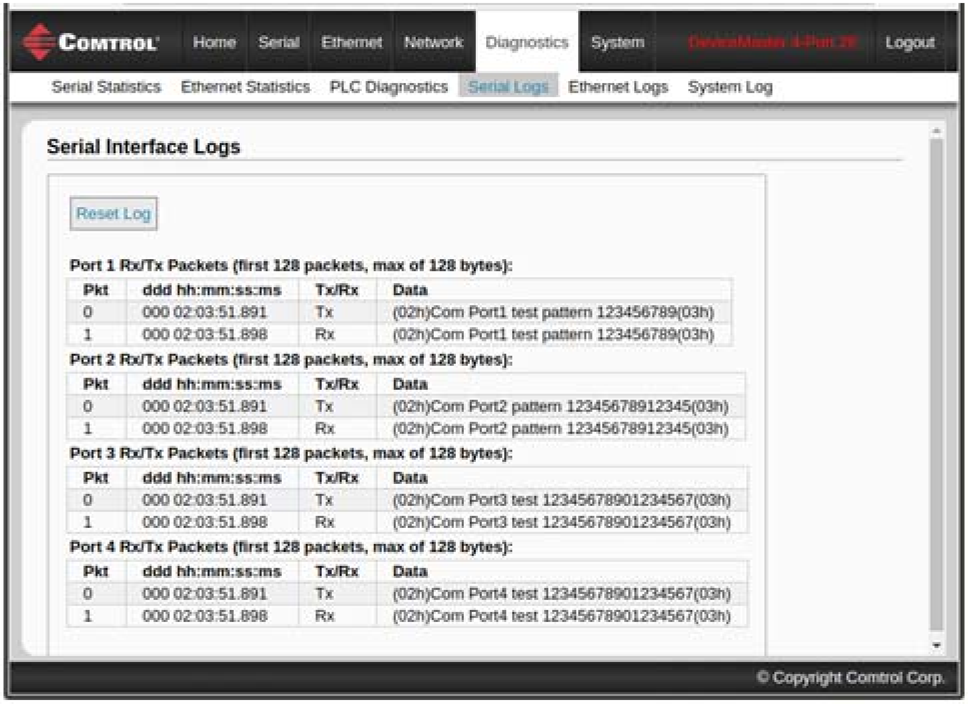
ms – minutes since last system restart

ss – seconds since last system restart

mmm – milliseconds since last system restart

<Data> - Data packet received.

- ASCII characters displayed as characters
- Non-ASCII displayed in hex (xxh) format



Serial Interface Logs

Port 1 Rx/Tx Packets (first 128 packets, max of 128 bytes):

Pkt	ddd hh:mm:ss:ms	Tx/Rx	Data
0	000 02:03:51.891	Tx	(02h)Com Port1 test pattern 123456789(03h)
1	000 02:03:51.898	Rx	(02h)Com Port1 test pattern 123456789(03h)

Port 2 Rx/Tx Packets (first 128 packets, max of 128 bytes):

Pkt	ddd hh:mm:ss:ms	Tx/Rx	Data
0	000 02:03:51.891	Tx	(02h)Com Port2 pattern 12345678912345(03h)
1	000 02:03:51.898	Rx	(02h)Com Port2 pattern 12345678912345(03h)

Port 3 Rx/Tx Packets (first 128 packets, max of 128 bytes):

Pkt	ddd hh:mm:ss:ms	Tx/Rx	Data
0	000 02:03:51.891	Tx	(02h)Com Port3 test 12345678901234567(03h)
1	000 02:03:51.898	Rx	(02h)Com Port3 test 12345678901234567(03h)

Port 4 Rx/Tx Packets (first 128 packets, max of 128 bytes):

Pkt	ddd hh:mm:ss:ms	Tx/Rx	Data
0	000 02:03:51.891	Tx	(02h)Com Port4 test 12345678901234567(03h)
1	000 02:03:51.898	Rx	(02h)Com Port4 test 12345678901234567(03h)

5.5. Ethernet Device Logs

The *Ethernet Device Interface Logs* page is accessed using the Display Ethernet Device Logs option, which provides a log of received and transmitted Ethernet device messages. Up to 128 bytes per message and up to 128 messages are logged. It is intended to help with debugging Ethernet connectivity problems, determining the proper start and end of transmission bytes, and diagnosing device problems.

The format is as follows: **Pkt(n)**: ddd:hh:mm:ss.ms Tx/Rx:<Data>

Where:

ddd – days since last system restart

hh – hours since last system restart

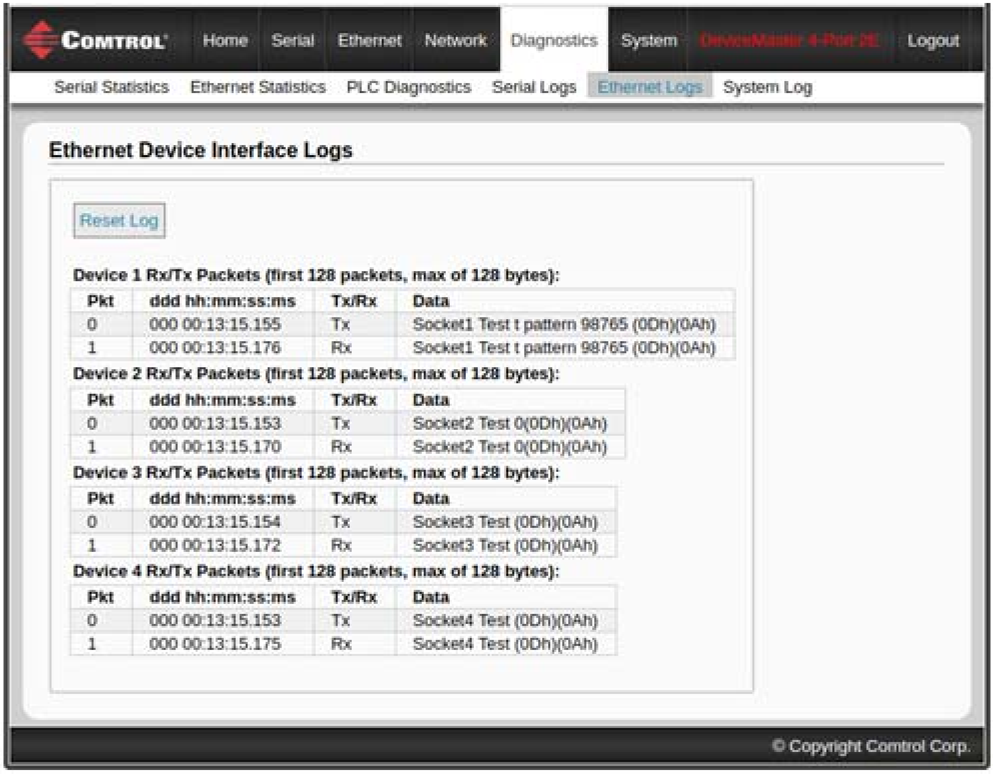
mm – minutes since last system restart

ss – seconds since last system restart

ms – milliseconds since last system restart

<**Data**> - Data packet received.

- ASCII characters displayed as characters
- Non-ASCII displayed in hex (xxh) format



Ethernet Device Interface Logs

Device 1 Rx/Tx Packets (first 128 packets, max of 128 bytes):

Pkt	ddd hh:mm:ss:ms	Tx/Rx	Data
0	000 00:13:15.155	Tx	Socket1 Test t pattern 98765 (0Dh)(0Ah)
1	000 00:13:15.176	Rx	Socket1 Test t pattern 98765 (0Dh)(0Ah)

Device 2 Rx/Tx Packets (first 128 packets, max of 128 bytes):

Pkt	ddd hh:mm:ss:ms	Tx/Rx	Data
0	000 00:13:15.153	Tx	Socket2 Test 0(0Dh)(0Ah)
1	000 00:13:15.170	Rx	Socket2 Test 0(0Dh)(0Ah)

Device 3 Rx/Tx Packets (first 128 packets, max of 128 bytes):

Pkt	ddd hh:mm:ss:ms	Tx/Rx	Data
0	000 00:13:15.154	Tx	Socket3 Test (0Dh)(0Ah)
1	000 00:13:15.172	Rx	Socket3 Test (0Dh)(0Ah)

Device 4 Rx/Tx Packets (first 128 packets, max of 128 bytes):

Pkt	ddd hh:mm:ss:ms	Tx/Rx	Data
0	000 00:13:15.153	Tx	Socket4 Test (0Dh)(0Ah)
1	000 00:13:15.175	Rx	Socket4 Test (0Dh)(0Ah)

5.6. System Log

The *System Log* page provides system level information, which is updated every 10 seconds.

Click the **Refresh** button to view the latest system log information.

Click the **Clear** button to restart a new system log file.

Click the **Save Logfile** button to save the system log file, if requested by Technical Support.

Chapter 6. Programming the PLC

This chapter provides information to help you configure PLC programs for the DeviceMaster. It includes instructions for modifying the PLC program examples included with the DeviceMaster. The PLC program examples are designed to work with the ControlLogix line of PLCs, PLC-5s or SLCs.

6.1. Programming Guidelines

Choose the receive communication method that best suits your implementation. The following receive communication methods are available.

Communications Methods	Description
Unsolicited - Write-to-Tag/File (Recommended Method)	DeviceMaster writes the serial/socket data directly into a tag or file on the PLC. Use this method along with the **Maximum PLC Update Rate** configuration setting to ensure the PLC can scan and consume serial/socket data faster than the DeviceMaster can send it
Unsolicited - Write-to-Tag/File-Synced	DeviceMaster writes the serial/socket data into a tag or file on the PLC and provides a mechanism to synchronize the data flow between the PLC and DeviceMaster. Use this method when you want to ensure that the tag or file is not overwritten before the PLC can consume the data.
Polling	The PLC requests data on a periodic basis. It provides the ability to control the received data flow. However, it does require periodic data requests and the request rate must be fast enough to ensure that the serial/socket port receive queues on the DeviceMaster do not overflow.
Class 1	First, the PLC sends a forward open message to the DeviceMaster to establish a UDP Ethernet connection between the PLC and the gateway. Once the connection is established, the PLC and DeviceMaster send input and output data to each other in a cyclic, scheduled manner. Typical cyclic rates

- *For ControlLogix PLCs*: The receive data tags must be large enough to handle the largest packet that can be received from your serial/socket device.
  - For *Polling receive* method: Including the sequence number and length parameters, the DeviceMaster can return a received data message up to 444 bytes.
  - For *Write-to-Tag* and *Write-to-Tag-Synced receive* methods: If the data is larger than the maximum of 440 bytes (up to 1518 serial bytes and 2048 socket bytes), the DeviceMaster sends the data to a series of tags.
  - For Class1, the receive and transmit connection sizes must be able to transfer all data between the PLC and the DeviceMaster, where the data is transmit to and received from the devices.

  See *3.2.2. Serial Port Data Transfer Object Definition (71 Hex)* on Page 33 and *3.2.5. Socket Port Data Transfer Definition Object (74 Hex)* on Page 52 for more information.
- *For PLC-5/SLC PLCs*: The receive data file area must be large enough to handle the largest packet that can be received from your serial/socket device.
  - For *Polling receive* method: Including the sequence number and length parameters, the DeviceMaster can return a received data message up to approximately 224 bytes.
  - For *Write-to-File* and *Write-to-File-Synced receive* methods: If the data size is large (up to 1518 serial bytes and 2048 socket bytes), the DeviceMaster sends the data to a series of files in sequential order.

  See *3.3.4. Receive Data Message* on Page 79 for details.
- *For some PLCs*, it is best to have only one PLC generated Class 3 EtherNet/IP message active at any one time in the PLC program.

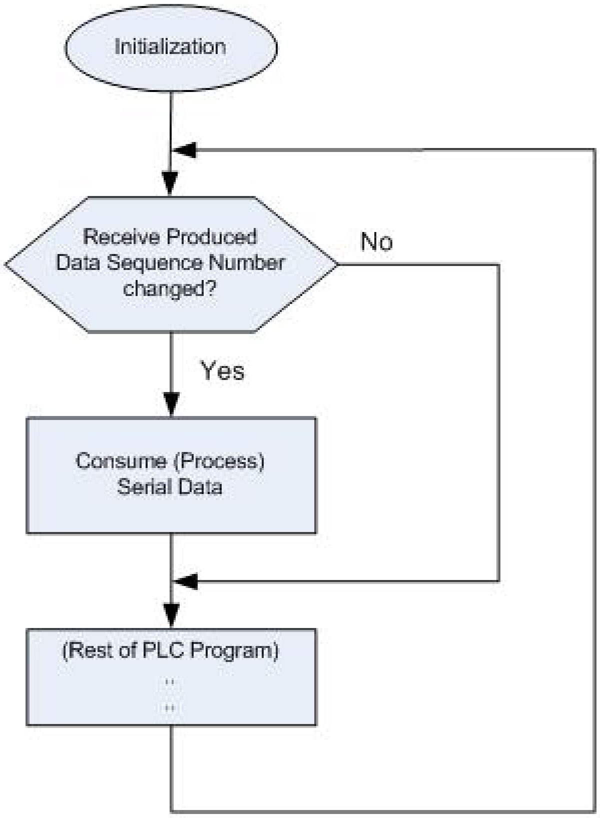
6.2. PLC Algorithms

This section displays the following PLC algorithms:

- *Unsolicited - Write-to-Tag/File PLC Algorithm* on Page 146
- *Unsolicited - Write-to-Tag/File-Synced PLC Algorithm* on Page 147
- *Polling PLC Algorithm* on Page 148

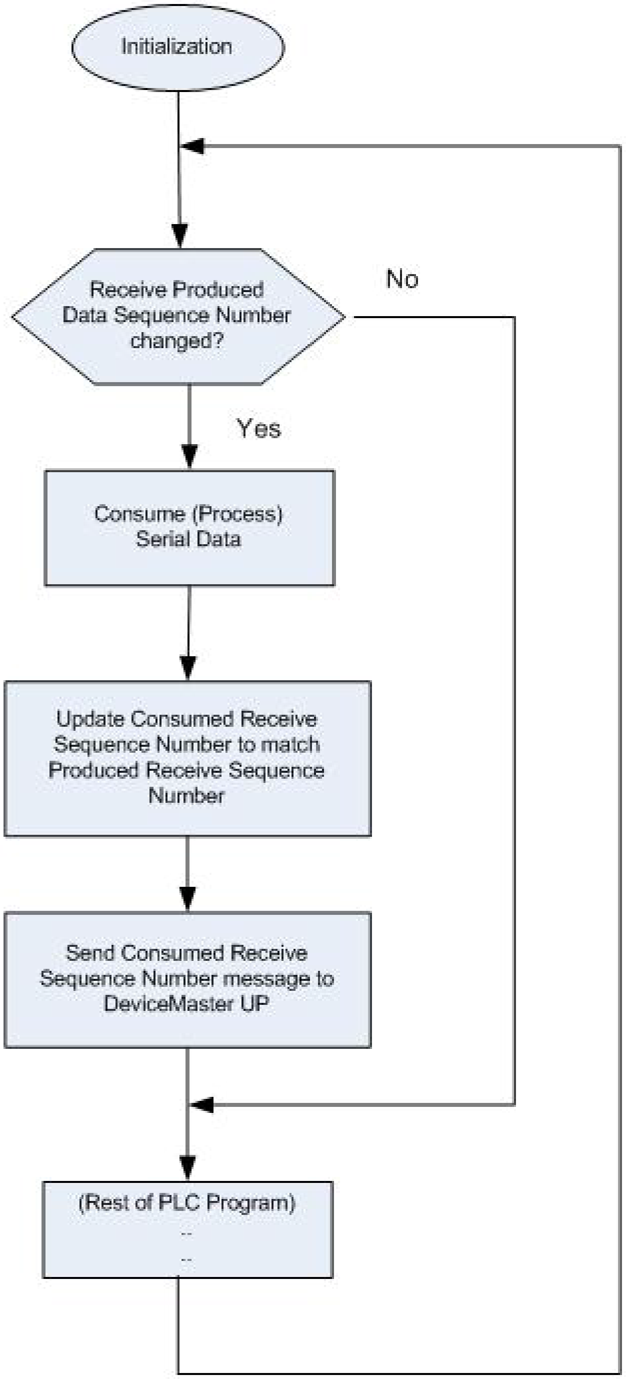
6.2.1. Unsolicited - Write-to-Tag/File PLC Algorithm

Use the following algorithm to receive data in the *Unsolicited - Write-to-Tag/File* mode.


Initialization
Receive Produced Data Sequence Number changed?
No
Yes
Consume (Process) Serial Data
(Rest of PLC Program)
..
..

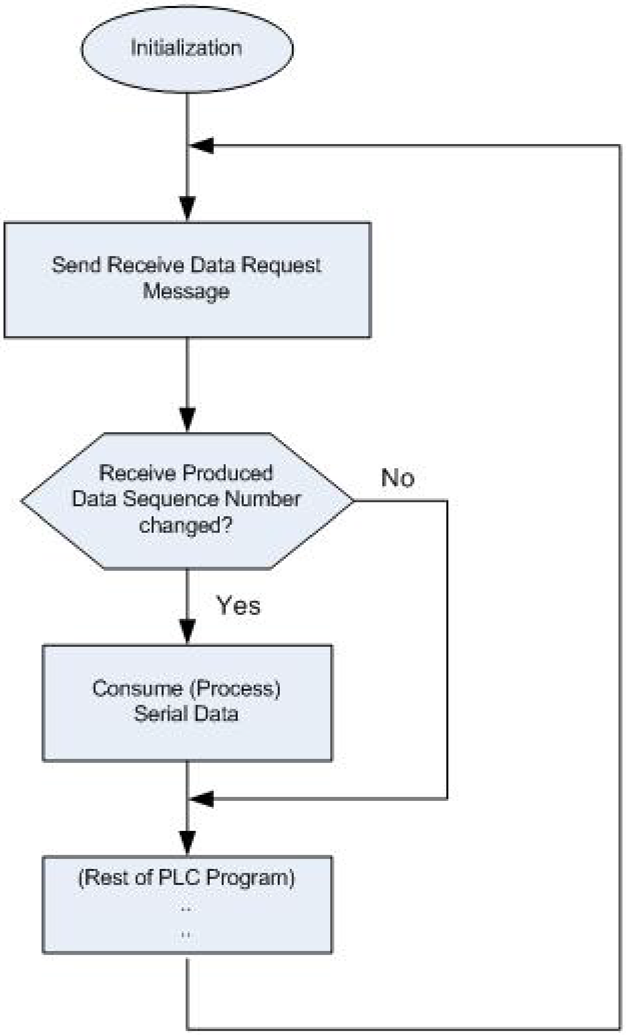
6.2.2. Unsolicited - Write-to-Tag/File-Synced PLC Algorithm

Use the following algorithm to receive data in the *Unsolicited - Write-to-Tag/File-Synced* mode.


Initialization
Receive Produced Data Sequence Number changed?
No
Yes
Consume (Process) Serial Data
Update Consumed Receive Sequence Number to match Produced Receive Sequence Number
Send Consumed Receive Sequence Number message to DeviceMaster UP
(Rest of PLC Program)
--
--

6.2.3. Polling PLC Algorithm

Use the following algorithm to *Receive Data in the Polling* mode.

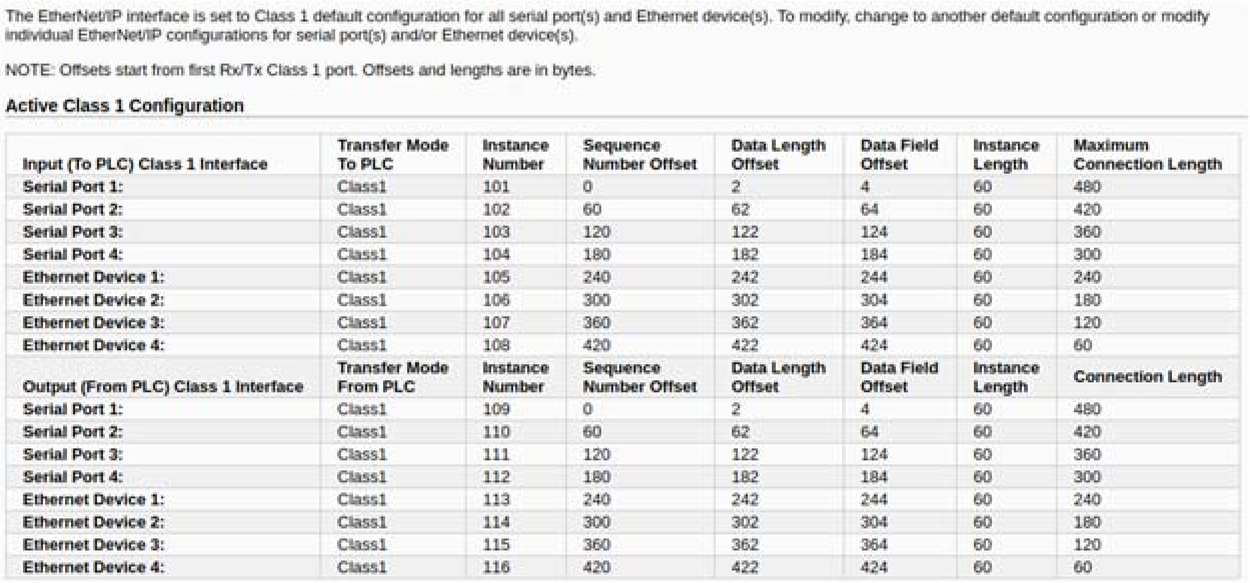

Initialization
Send Receive Data Request Message
Receive Produced Data Sequence Number changed?
No
Yes
Consume (Process) Serial Data
(Rest of PLC Program)
..
..

6.3. Class 1 Interface

The EtherNet/IP firmware provides configurable Class 1 interface capability for all serial and Ethernet socket ports. While each port can be configured to operate in several different receive and transmit modes, these examples are meant to display the case where all serial and socket ports are configured to the **All Ports Default Class 1** configuration.

6.3.1. Configuring an I/O Ethernet Module

Each DeviceMaster must be best set up as a Generic Ethernet Module to interface to the PLC. The Class 1 interface for a four port gateway is displayed on the *Class 1 Overview* web page.



The EtherNet/IP interface is set to Class 1 default configuration for all serial port(s) and Ethernet device(s). To modify, change to another default configuration or modify individual EtherNet/IP configurations for serial port(s) and/or Ethernet device(s).

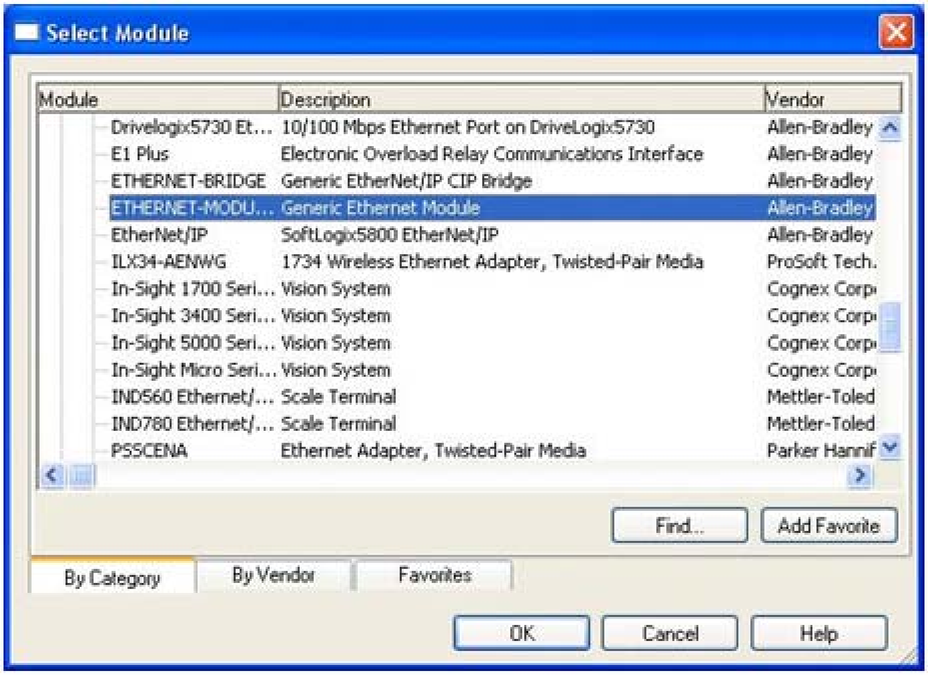
NOTE: Offsets start from first Rx/Tx Class 1 port. Offsets and lengths are in bytes.

Active Class 1 Configuration

Input (To PLC) Class 1 Interface	Transfer Mode To PLC	Instance Number	Sequence Number Offset	Data Length Offset	Data Field Offset	Instance Length	Maximum Connection Length
Serial Port 1:	Class1	101	0	2	4	60	480
Serial Port 2:	Class1	102	60	62	64	60	420
Serial Port 3:	Class1	103	120	122	124	60	360
Serial Port 4:	Class1	104	180	182	184	60	300
Ethernet Device 1:	Class1	105	240	242	244	60	240
Ethernet Device 2:	Class1	106	300	302	304	60	180
Ethernet Device 3:	Class1	107	360	362	364	60	120
Ethernet Device 4:	Class1	108	420	422	424	60	60
Output (From PLC) Class 1 Interface	**Transfer Mode From PLC**	**Instance Number**	**Sequence Number Offset**	**Data Length Offset**	**Data Field Offset**	**Instance Length**	**Connection Length**
Serial Port 1:	Class1	109	0	2	4	60	480
Serial Port 2:	Class1	110	60	62	64	60	420
Serial Port 3:	Class1	111	120	122	124	60	360
Serial Port 4:	Class1	112	180	182	184	60	300
Ethernet Device 1:	Class1	113	240	242	244	60	240
Ethernet Device 2:	Class1	114	300	302	304	60	180
Ethernet Device 3:	Class1	115	360	362	364	60	120
Ethernet Device 4:	Class1	116	420	422	424	60	60

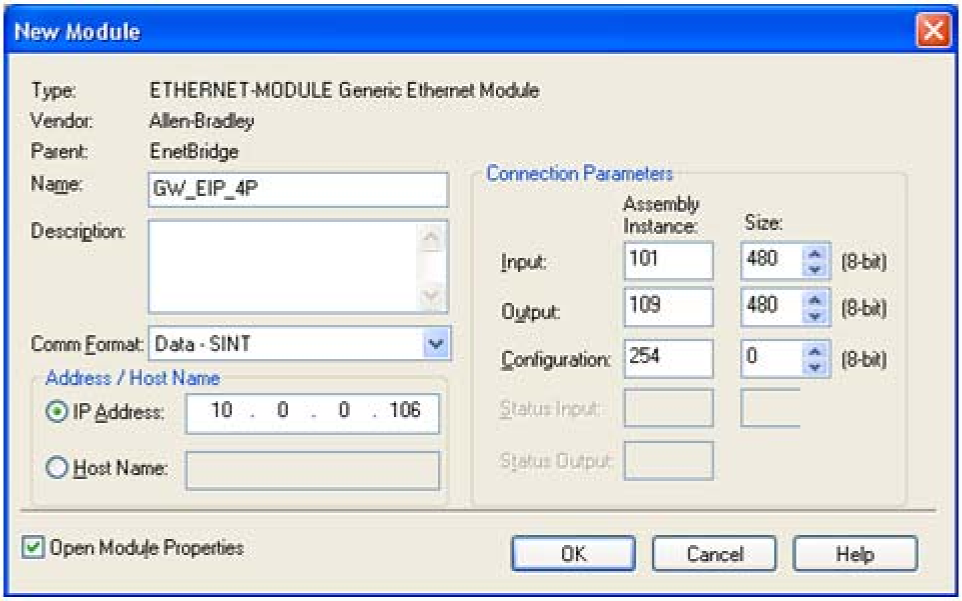
1. Right-click the Ethernet module on RSLogix5000 and select **New Module**.
2. Click **Communications**.

3. Scroll down and select **Generic Ethernet Module**.



Module	Description	Vendor
Drivelogix5730 Et...	10/100 Mbps Ethernet Port on DriveLogix5730	Allen-Bradley
E1 Plus	Electronic Overload Relay Communications Interface	Allen-Bradley
ETHERNET-BRIDGE	Generic EtherNet/IP CIP Bridge	Allen-Bradley
ETHERNET-MODU...	Generic Ethernet Module	Allen-Bradley
EtherNet/IP	SoftLogix5800 EtherNet/IP	Allen-Bradley
ILX34-AENWG	1734 Wireless Ethernet Adapter, Twisted-Pair Media	ProSoft Tech.
In-Sight 1700 Seri...	Vision System	Cognex Corp
In-Sight 3400 Seri...	Vision System	Cognex Corp
In-Sight 5000 Seri...	Vision System	Cognex Corp
In-Sight Micro Seri...	Vision System	Cognex Corp
IND560 Ethernet/...	Scale Terminal	Mettler-Toled
IND780 Ethernet/...	Scale Terminal	Mettler-Toled
PSSCENA	Ethernet Adapter, Twisted-Pair Media	Parker Hannif

4. Click **Ok** and the following pane appears.



Type: ETHERNET-MODULE Generic Ethernet Module

Vendor: Allen-Bradley

Parent: EnetBridge

Name: GW_EIP_4P

Description:

Comm Format: Data - SINT

Address / Host Name

● IP Address: 10 . 0 . 0 . 106

○ Host Name:

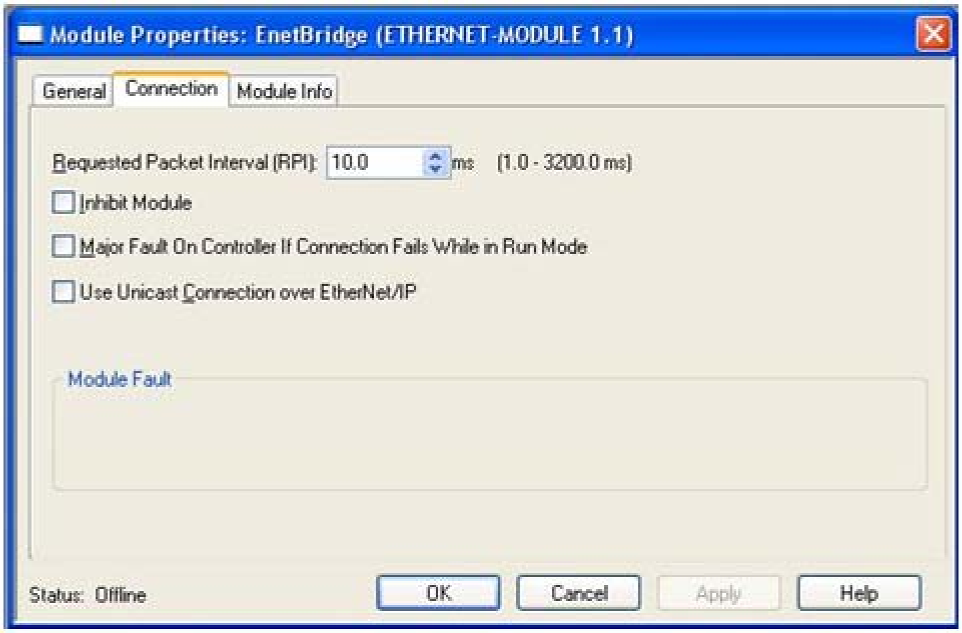
Connection Parameters

	Assembly Instance:	Size:	
Input:	101	480	(8-bit)
Output:	109	480	(8-bit)
Configuration:	254	0	(8-bit)
Status Input:			
Status Output:			

☑ Open Module Properties

a. Enter a **Name**.
b. Select the **Comm Format** as **Data-SINT**.
c. Enter the IP Address of the gateway.
d. Using the data displayed on the Class 1 Overview web page, enter the **Connection Parameters**.

5. Click **OK** and the following pane appears.



Module Properties: EnetBridge (ETHERNET-MODULE 1.1)

Requested Packet Interval (RPI): 10.0 ms (1.0 - 3200.0 ms)

☐ Inhibit Module

☐ Major Fault On Controller If Connection Fails While in Run Mode

☐ Use Unicast Connection over EtherNet/IP

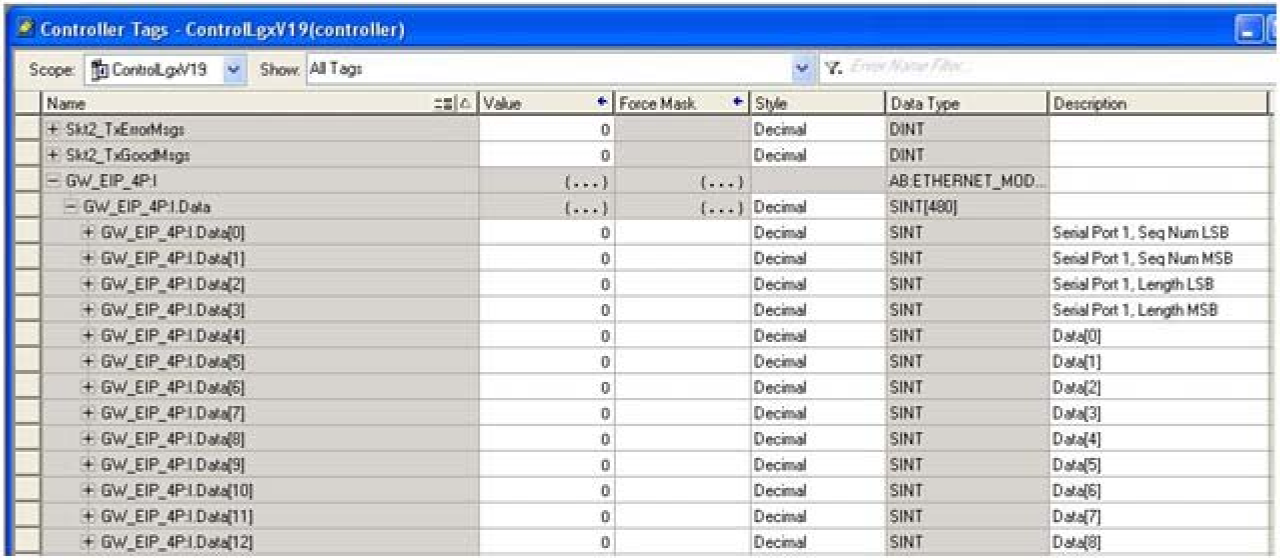
Module Fault

Status: Offline

- The fastest allowable RPI is 10 ms.
- Both Unicast, (point-to-point), and Multicast, (one-to-many), connections are supported.

6. Click **OK**. The module will be added.
7. View the corresponding Input and Output data tags created when the gateway module was added. Note the location of the sequence number, length and data fields as displayed on the *Class 1 Overview* page. Only the first serial is displayed below.

Input Tag:

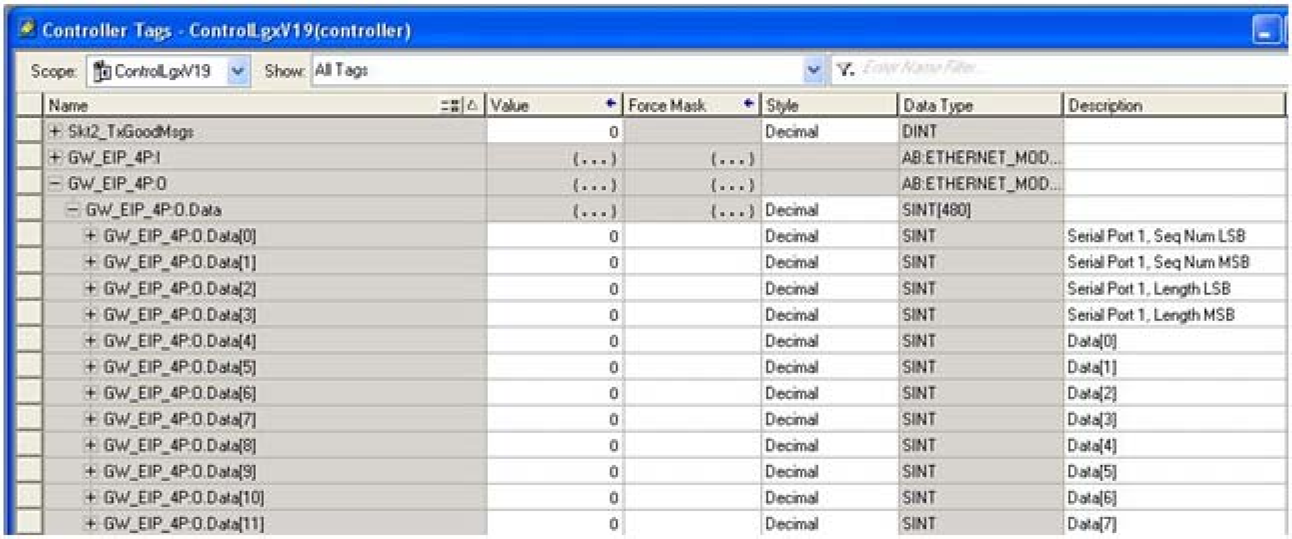


Controller Tags - ControlLgxV19(controller)

Scope: ControlLgxV19 Show: All Tags

Name	Value	Force Mask	Style	Data Type	Description
+ Slot2_TxErrorMsgs	0		Decimal	DINT	
+ Slot2_TxGoodMsgs	0		Decimal	DINT	
− GW_EIP_4P:I	{...}	{...}		AB:ETHERNET_MOD...	
− GW_EIP_4P:I.Data	{...}	{...}	Decimal	SINT[480]	
+ GW_EIP_4P:I.Data[0]	0		Decimal	SINT	Serial Port 1, Seq Num LSB
+ GW_EIP_4P:I.Data[1]	0		Decimal	SINT	Serial Port 1, Seq Num MSB
+ GW_EIP_4P:I.Data[2]	0		Decimal	SINT	Serial Port 1, Length LSB
+ GW_EIP_4P:I.Data[3]	0		Decimal	SINT	Serial Port 1, Length MSB
+ GW_EIP_4P:I.Data[4]	0		Decimal	SINT	Data[0]
+ GW_EIP_4P:I.Data[5]	0		Decimal	SINT	Data[1]
+ GW_EIP_4P:I.Data[6]	0		Decimal	SINT	Data[2]
+ GW_EIP_4P:I.Data[7]	0		Decimal	SINT	Data[3]
+ GW_EIP_4P:I.Data[8]	0		Decimal	SINT	Data[4]
+ GW_EIP_4P:I.Data[9]	0		Decimal	SINT	Data[5]
+ GW_EIP_4P:I.Data[10]	0		Decimal	SINT	Data[6]
+ GW_EIP_4P:I.Data[11]	0		Decimal	SINT	Data[7]
+ GW_EIP_4P:I.Data[12]	0		Decimal	SINT	Data[8]

Output Tag:



Controller Tags - ControlLgxV19(controller)

Scope: ControlLgxV19 Show: All Tags

Name	Value	Force Mask	Style	Data Type	Description
+ Skt2_TxGoodMsgs	0		Decimal	DINT	
+ GW_EIP_4P:I	{...}	{...}		AB:ETHERNET_MOD...	
− GW_EIP_4P:O	{...}	{...}		AB:ETHERNET_MOD...	
− GW_EIP_4P:O.Data	{...}	{...}	Decimal	SINT[480]	
+ GW_EIP_4P:O.Data[0]	0		Decimal	SINT	Serial Port 1, Seq Num LSB
+ GW_EIP_4P:O.Data[1]	0		Decimal	SINT	Serial Port 1, Seq Num MSB
+ GW_EIP_4P:O.Data[2]	0		Decimal	SINT	Serial Port 1, Length LSB
+ GW_EIP_4P:O.Data[3]	0		Decimal	SINT	Serial Port 1, Length MSB
+ GW_EIP_4P:O.Data[4]	0		Decimal	SINT	Data[0]
+ GW_EIP_4P:O.Data[5]	0		Decimal	SINT	Data[1]
+ GW_EIP_4P:O.Data[6]	0		Decimal	SINT	Data[2]
+ GW_EIP_4P:O.Data[7]	0		Decimal	SINT	Data[3]
+ GW_EIP_4P:O.Data[8]	0		Decimal	SINT	Data[4]
+ GW_EIP_4P:O.Data[9]	0		Decimal	SINT	Data[5]
+ GW_EIP_4P:O.Data[10]	0		Decimal	SINT	Data[6]
+ GW_EIP_4P:O.Data[11]	0		Decimal	SINT	Data[7]

6.4. ControlLogix PLC Programming Example Instructions

This topic describes how to use RSLogix 5000 to configure and run the DeviceMaster in a ControlLogix environment.

You can configure the RSLogix 5000 PLC program examples to your site's requirements. These programs are included in the self-installing file (**.MSI**) and are copied to the **Comtrol/EtherNetIP** directory on your computer when you open the **.MSI** file and follow the prompts. The self-installing file includes the following RSLogix 5000 PLC program examples:

- *6.4.3. loopbackExampleTagWrite.L5K* on Page 154
- *6.4.4. loopbackExampleTagWriteSynced.L5K* on Page 155
- *6.4.5. loopbackExamplePolling.L5K* on Page 156

These program examples are intended to aid the PLC programmer. The PLC programmer can use and modify these PLC program examples to suit their needs.

Note: *The PLC program examples are designed to interface with a* DeviceMaster *1-port or on Port 1 of a 2-port or 4-port. Additional programming is required to use all ports on a 2-port or 4-port.*

⚠ Caution

Disclaimer: **Comtrol supplies example PLC programs for demonstration purposes only. They are intended for the sole purpose of an example loop-back demonstration in a controlled lab environment. They are not intended for use in a production environment and may not function correctly on all PLCs. Comtrol does not warrant these example programs or any part thereof. The user assumes all liability for any modification to and use of a modified example program.**

6.4.1. What is RSLogix 5000?

RSLogix 5000 is a programming application that is designed to interface with the ControlLogix line of PLCs. You can use it for sequential, process, drive, and motion control programming. It provides an easy-to-use interface, symbolic programming with structures and arrays and an instruction set that serves many types of applications. It simplifies plant maintenance by providing one programming environment for all of your applications.

Note: *See the RSLogix 5000 Help for more information on this product.*

6.4.2. Requirements

- The DeviceMaster must be installed and configured as described in the *EtherNet/IP Hardware Installation and Configuration Guide*.
- The DeviceMaster must be installed on the same Ethernet network segment as the PLC.
- RSLogix 5000 must be installed on a computer. Note that the instructions in this guide require that you have some familiarity with this programming application.
- A loopback plug is required for each port on the DeviceMaster. See the *EtherNet/IP Hardware Installation and Configuration Guide* for information on loopback plugs.
- The PLC program (**.L5K** files) examples are required. You can copy the PLC program examples from the CD or download the latest program examples from the Internet. See 1.4.1. Locating the Latest Software and Documents on Page 8 for the location of the PLC program examples.

6.4.3. loopbackExampleTagWrite.L5K

This example program demonstrates an RSLogix 5000 loopback PLC program using the *Unsolicited - Write-to-Tag receive* method. This program configures a DeviceMaster 1-port at startup and then loops data by means of the loopback plug on the serial port. The **Com1_RxData** tag transmits and receives the data and increments the sequence numbers.

You can configure and run the **loopbackExampleTagWrite.L5K** program through RSLogix 5000. For additional information on the RSLogix 5000, see *6.4.7. RSLogix 5000 Screen Examples* on Page 158.

1. Attach a loopback plug to the serial port. (If necessary, see the *EtherNet/IP Hardware Installation and Configuration Guid*)
2. Open **RSLogix 5000**.
3. Import the **loopbackExampleTagWrite.L5K** file.

   Note: *If you have problems loading the PLC program example, see 6.4.8. Modifying an RSLogix 5000 PLC Program Example (Older Versions) on Page 164 for a solution.*
4. Modify the PLC program (**loopbackExampleTagWrite.L5K**) for your system. *3.2. EtherNet/IP Interface Profile (ControlLogix)* on Page 20 provides a description of the objects in the PLC program.

   This PLC program is configured to run on a ControlLogix 5550 controller, so you may need to make the following changes:

   a. From the **Controller Organizer** panel, click **Controller ComtrolCLX** and select **Properties**.

   b. Click **Change Controller**, select your PLC controller and revision.

   c. Right-click the **I/O Configuration**, select **New Module**, and add your EtherNet/IP interface to the project.

   d. Double-click **MainRoutine** under **Tasks** > **MainTask** > **MainProgram** in the **Controller Organizer** panel and modify the message communication paths. Click each message and change the communication path to:

   `<Ethernet Interface Name>,2,`<DeviceMaster **`IP Address>`**

   Where <**Ethernet Interface Name**> specifies the name of your Ethernet interface and <DeviceMaster **IP Address**> specifies the IP address for this device.
5. Use one of the following procedures:

   For embedded web page configuration (recommended): Perform the steps outlined in *6.4.6. Configuring the DeviceMaster for the RSLogix 5000 Example Programs Using the Web Page* on Page 157.

   For the optional PLC Configuration:

   a. Double-click **Controller Tags** and click the plus sign (**+**) next to **Com1_SetConfigData** to expand the option and change the serial port configuration tag.

   b. Change the **plcSlotNumber** parameter to match the PLC slot number on your PLC chassis.

   c. Change the **plcIPAddress** parameter to match the IP address of your PLC EtherNet/IP card. (The value must be entered in 32-bit hex format.

   d. Enable **DoSetConfig** on rung one of the example PLC program.
6. Download the PLC program to your PLC and run the program.

6.4.4. loopbackExampleTagWriteSynced.L5K

This example program demonstrates an RSLogix 5000 loopback PLC program using the *Write-to-Tag-Synced receive* method. This program configures a DeviceMaster 1-port at startup and then loops data through the loopback plug on the serial port. The **Com1_RxData** tag transmits and receives the data, increments the sequence numbers, and sends the consumed receive sequence number to the DeviceMaster after each received data packet.

You can configure and run the **loopbackExampleTagWriteSynced.L5K** program through RSLogix 5000. For additional information on the RSLogix 5000, see *6.4.7. RSLogix 5000 Screen Examples* on Page 158.

1. Attach a loopback plug to the serial port. (If necessary, see the *EtherNet/IP Hardware Installation and Configuration Guide*.)
2. Open **RSLogix 5000**.
3. Import the **loopbackExampleTagWriteSynced.L5K** file.

   Note: *If you have problems loading the PLC program example, see 6.4.8. Modifying an RSLogix 5000 PLC Program Example (Older Versions) on Page 164 for a solution.*
4. Modify the PLC program (**loopbackExampleTagWriteSynced.L5K**) for your system.

   3.2. EtherNet/IP Interface Profile (ControlLogix) on Page 20 provides a description of the objects in the PLC program. This PLC program is configured to run on a ControlLogix 5550 controller, so you may need to make the following changes:

   a. From the **Controller Organizer** panel, click **Controller ComtrolCLX** and select **Properties**.

   b. Click **Change Controller**, select your PLC controller and revision.

   c. Right-click the **I/O Configuration**, select **New Module**, and add your EtherNet/IP interface to the project.

   d. Double-click **MainRoutine** under **Tasks** > **MainTask** > **MainProgram** in the **Controller Organizer** panel and modify the message communication paths. Click each message and change the communication path to:

   `<Ethernet Interface Name>,2,<`DeviceMaster **`IP Address>`**

   Where <**Ethernet Interface Name**> specifies the name of your Ethernet interface and <DeviceMaster **IP Address**> specifies the IP address for this device.
5. Use one of the following procedures:

   For embedded Web page configuration (recommended): Perform the steps outlined in *6.4.6. Configuring the DeviceMaster for the RSLogix 5000 Example Programs Using the Web Page* on Page 157.

   For the optional PLC Configuration:

   a. Double-click **Controller Tags** and click the plus sign (**+**) next to **Com1_SetConfigData** to expand the option and change the serial port configuration tag.

   b. Change the **plcSlotNumber** parameter to match the PLC slot number on your PLC chassis.

   c. Change the **plcIPAddress** parameter to match the IP address of your PLC EtherNet/IP card. (The value must be entered in 32-bit hex format.

   d. Enable **DoSetConfig** on rung one of the example PLC program.
6. Download the PLC program to your PLC and run the program.

6.4.5. loopbackExamplePolling.L5K

This example program demonstrates an RSLogix 5000 loopback PLC program using the *Polling receive* method. This program configures a DeviceMaster 1-port at startup and then loops data through the loopback plug on the serial port. The *Request Data Message* transmits and receives the data and increments the sequence numbers.

You can configure and run the **loopbackExamplePolling.L5K** program through RSLogix 5000. For additional information on the RSLogix 5000, see *6.4.7. RSLogix 5000 Screen Examples* on Page 158.

1. Attach a loopback plug to the serial port. (If necessary, see the *EtherNet/IP Hardware Installation and Configuration Guide*.)
2. Open **RSLogix 5000**.
3. Import the **loopbackExamplePolling.L5K** file.

   Note: *If you have problems loading the PLC program example, see 6.4.8. Modifying an RSLogix 5000 PLC Program Example (Older Versions) on Page 164 for a solution.*
4. Modify the PLC program (**loopbackExamplePolling.L5K**) for your system.

   3.2. EtherNet/IP Interface Profile (ControlLogix) on Page 20 provides a description of the objects in the PLC program.

   Note that this PLC program is configured to run on a ControlLogix 5550 controller Virtual Chassis, so you may need to make the following changes:

   a. From the **Controller Organizer** panel, click **Controller ComtrolCLX** and select **Properties**.

   b. Click **Change Controller**, select your PLC controller and revision.

   c. Right-click the **I/O Configuration**, select **New Module**, and add your EtherNet/IP interface to the project.

   d. Double-click **MainRoutine** under **Tasks** > **MainTask** > **MainProgram** in the **Controller Organizer** panel and modify the message communication paths. Click each message and change the communication path to:

   `<Ethernet Interface Name>,2,<`DeviceMaster **`IP Address>`**

   Where <**Ethernet Interface Name**> specifies the name of your Ethernet interface and <DeviceMaster **IP Address**> specifies the IP address for this device.
5. Use one of the following procedures:

   For the embedded Web page configuration (recommended): Perform the steps outlined in *6.4.6. Configuring the DeviceMaster for the RSLogix 5000 Example Programs Using the Web Page* on Page 157.

   For the optional PLC Configuration: Enable **DoSetConfig** on rung one of the example PLC program.
6. Download the PLC program to your PLC and run the program.

6.4.6. Configuring the DeviceMaster for the RSLogix 5000 Example Programs Using the Web Page

The following procedure configures the DeviceMaster for the RSLogix 500 example programs. You must perform this task before you configure and run the RSLogix 500 example program. For more information on the *Port Configuration* web pages, see *Chapter 4. Embedded Configuration Pages* on Page 85.

1. Attach a loopback plug to the serial port.
2. Access the *Port Configuration* web page, using one of these methods.
   - Open PortVision DX, right-click the DeviceMaster for which you want to program network information and click **Webpage.**
   - Open a browser and type the IP address for the DeviceMaster in the **Address** box.
3. Click **Serial | Serial Settings |Port *n***. Where ***n*** is the port number.
4. Set the serial port settings under **Serial Configuration** to the following values.

Field	Setting
Mode	RS-232
Baud	57600
Parity	none
Data Bits	8
Stop Bits	1
Flow Control	none
DTR	off
Rx Timeout Between Packets	200

5. Set the serial port settings under **Serial Packet Identification** to these values.

Field	Setting
STX RX Detect	Set to one byte and Byte 1 to 2.
ETX Rx Detect	Set to one byte and Byte 1 to 3.
STX Tx Append	Set to one byte and Byte 1 to 2.
ETX Tx Append	Set to one byte and Byte 1 to 3.
Strip Rx STX/ETX	Select
Discard Rx Packets With Errors	Select

6. Click the **Save** button.
7. Click **EtherNet/IP Settings** and set the serial port settings to the following values:

Field	Selection
TX Sequence Number Checking (*right column*)	Select.
Rx (To PLC) Ethernet Transfer Method	• Set to Polling for **lpbkExamplePolling**. • Set to Write-to-Tag/File for **lpbkExampleTagWrite**. • Set to Write-to-Tag/File-Synced for **lpbkExampleTagWriteSynced**.
PLC IP Address	• Leave blank for Polling. • Set to IP Address of PLC for **Write-to-File** and **Write-to-File-Synced**.
PLC Controller Slot Number	Set to PLC Controller slot number which typically starts at zero. (Set to zero for CompactLogix PLC.)
Rx (To PLC) Produced Data Tag/File Name	• Leave blank for Polling. • Set to Com1_RxData, the PLC receive filename for **Write-to-Tag** and **Write-to-Tag-Synced**.

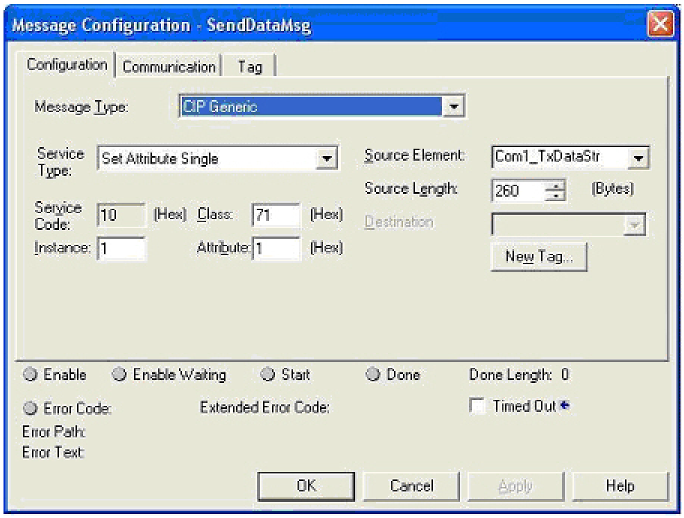
8. Click the **Save** button.

6.4.7. RSLogix 5000 Screen Examples

The following subsections explain how to configure the DeviceMaster through RSLogix 5000.

6.4.7.1. Transmit Data to the DeviceMaster

Use the **Configuration** tab in the **Message Configuration** dialog to set options for transmitting data through a specified port on the DeviceMaster.



Message Type: CIP Generic

Service Type: Set Attribute Single

Source Element: Com1_TxDataStr

Source Length: 260 (Bytes)

Destination:

Service Code: 10 (Hex) Class: 71 (Hex)

Instance: 1 Attribute: 1 (Hex)

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

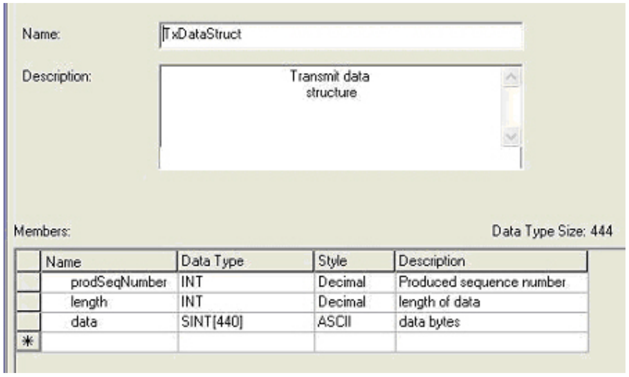
Error Path:

Error Text:

Provide the following information.

Field	Selection
Message Type	CIP Generic
Service Type	Set Attribute Single
Service Code	10 Hex (Set Attribute Single)
Class	71 Hex (Serial Port Data Transfer object)
Instance	1 (Port 1)
Attribute	1 (Transmit message data attribute)
Source Element	**Com1_TxDataStr** The transmit data structure includes: • Optional produced data sequence number (one INT) • Data Length in bytes (one INT) • Data array (one to 440-bytes)
Source Length	Specifies the length of the transmit data structure. (In this example, the value is 260 to transmit 256 data bytes. Since the maximum data size is 440-bytes, the maximum transmit data structure size is 444-bytes.)

The following image displays the user defined data type for transmit data.



Name: TxDataStruct

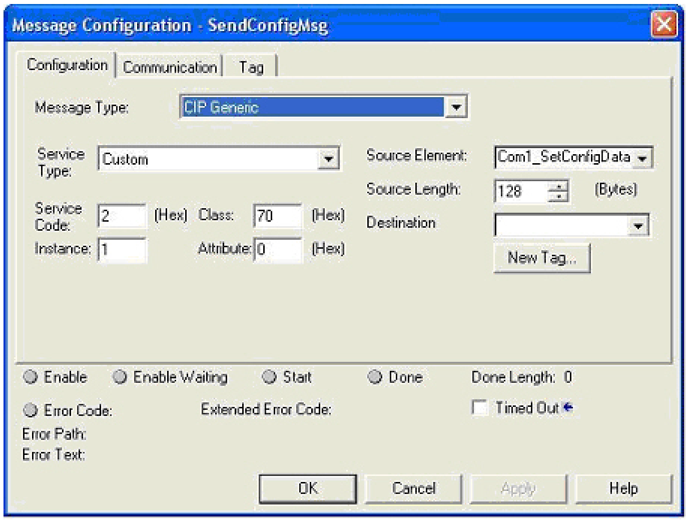
Description: Transmit data structure

Members: Data Type Size: 444

Name	Data Type	Style	Description
prodSeqNumber	INT	Decimal	Produced sequence number
length	INT	Decimal	length of data
data	SINT[440]	ASCII	data bytes

6.4.7.2. **Send Configuration to** DeviceMaster

Use the **Configuration** tab in the **Message Configuration** dialog for sending a serial port configuration to the DeviceMaster.



Message Configuration - SendConfigMsg

Message Type: CIP Generic

Service Type: Custom

Service Code: 2 (Hex) Class: 70 (Hex)

Instance: 1 Attribute: 0 (Hex)

Source Element: Com1_SetConfigData

Source Length: 128 (Bytes)

Destination:

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

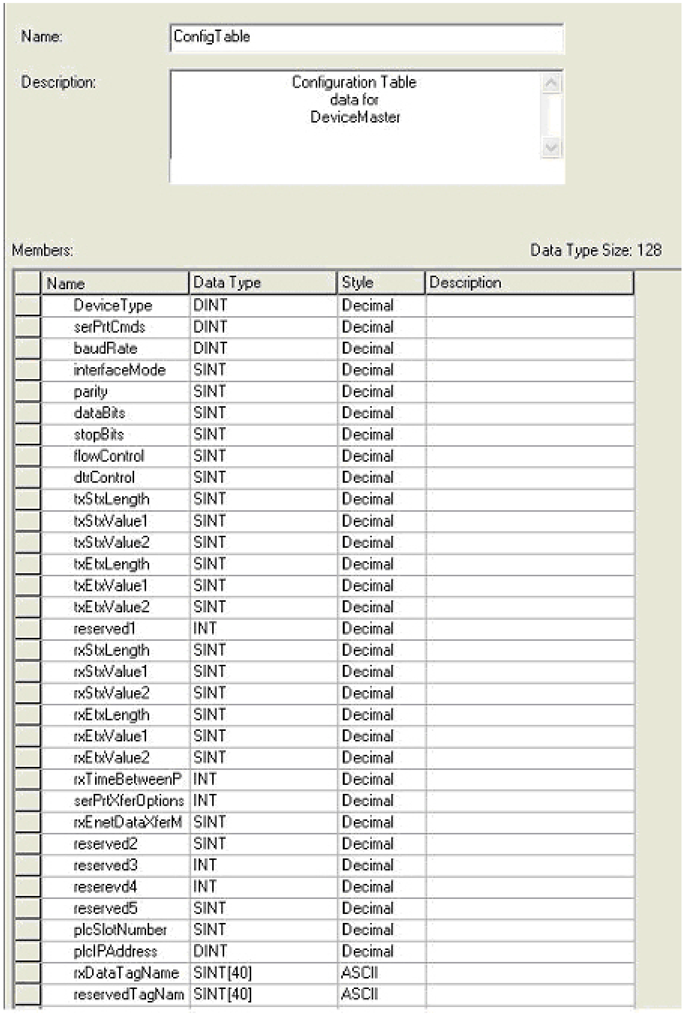
Error Path:

Error Text:

Provide the following information.

Field	Selection
Message Type	CIP Generic
Service Type	Custom
Service Code	2 Hex (Set Attribute All)
Class	70 Hex (Serial Port Configuration object)
Instance	1 (Port 1)
Attribute	0 (Unused)
Source Element	**Com1_SetConfigData** (Configuration data structure.)

The following image displays the user defined data type for a serial port configuration.



Name: ConfigTable

Description:
Configuration Table
data for
DeviceMaster

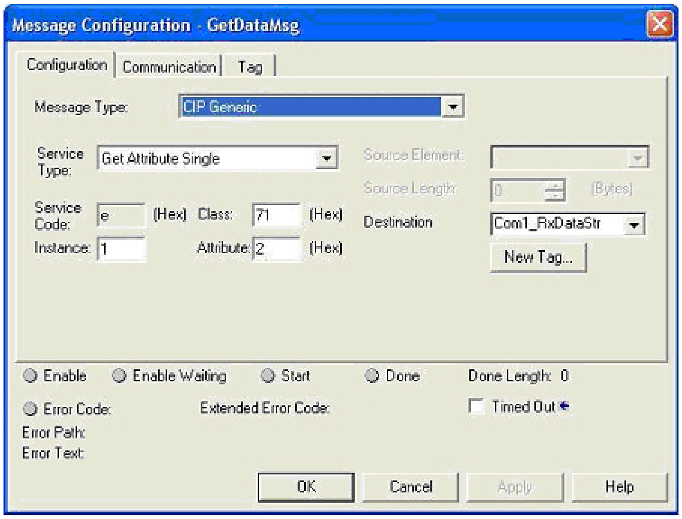
Members:

Data Type Size: 128

Name	Data Type	Style	Description
DeviceType	DINT	Decimal	
serPrtCmds	DINT	Decimal	
baudRate	DINT	Decimal	
interfaceMode	SINT	Decimal	
parity	SINT	Decimal	
dataBits	SINT	Decimal	
stopBits	SINT	Decimal	
flowControl	SINT	Decimal	
dtrControl	SINT	Decimal	
txStxLength	SINT	Decimal	
txStxValue1	SINT	Decimal	
txStxValue2	SINT	Decimal	
txEtxLength	SINT	Decimal	
txEtxValue1	SINT	Decimal	
txEtxValue2	SINT	Decimal	
reserved1	INT	Decimal	
rxStxLength	SINT	Decimal	
rxStxValue1	SINT	Decimal	
rxStxValue2	SINT	Decimal	
rxEtxLength	SINT	Decimal	
rxEtxValue1	SINT	Decimal	
rxEtxValue2	SINT	Decimal	
rxTimeBetweenP	INT	Decimal	
serPrtXferOptions	INT	Decimal	
rxEnetDataXferM	SINT	Decimal	
reserved2	SINT	Decimal	
reserved3	INT	Decimal	
reserevd4	INT	Decimal	
reserved5	SINT	Decimal	
plcSlotNumber	SINT	Decimal	
plcIPAddress	DINT	Decimal	
rxDataTagName	SINT[40]	ASCII	
reservedTagNam	SINT[40]	ASCII	

6.4.7.3. Request Data from DeviceMaster

Use the **Configuration** tab in the **Message Configuration** dialog to request data from a specified serial port on the DeviceMaster. Only the *Polling receive* method uses this feature.



Message Type: CIP Generic

Service Type: Get Attribute Single

Service Code: e (Hex) Class: 71 (Hex)

Instance: 1 Attribute: 2 (Hex)

Source Element:

Source Length: 0 (Bytes)

Destination: Com1_RxDataStr

New Tag...

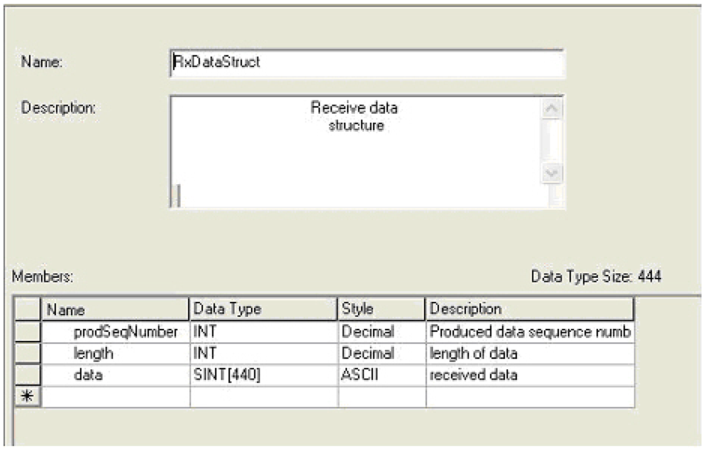
Done Length: 0

□ Timed Out

Provide the following information.

Field	Selection
Message Type	CIP Generic
Service Type	Get Attribute Single
Service Code	0E Hex (Get Attribute Single)
Class	71 Hex (Serial Port Data Transfer object)
Instance	1 (Port 1)
Attribute	2 (Receive message data attribute)
Destination	**Com1_RxDataStr** (Receive data structure) The Receive data structure includes: • Produced data sequence number (one INT). A change indicates new data. • Data length in bytes (one INT). • Data array (zero to 440-bytes)
Source Length	128 (Length of the configuration data structure.)

The following image displays the user defined data type for received data.



Name: RxDataStruct

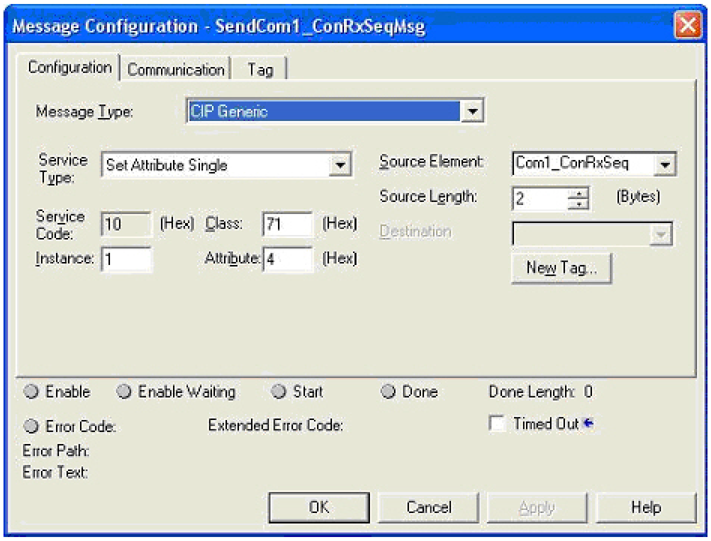
Description: Receive data structure

Members: Data Type Size: 444

Name	Data Type	Style	Description
prodSeqNumber	INT	Decimal	Produced data sequence numb
length	INT	Decimal	length of data
data	SINT[440]	ASCII	received data
*			

6.4.7.4. Send Consumed Sequence Number to DeviceMaster

Use the **Configuration** tab in the **Message Configuration** dialog to update the consumed sequence number for a specified port on the DeviceMaster. Only the *Write-to-Tag-Synced receive* method uses this feature.



Message Configuration - SendCom1_ConRxSeqMsg

Message Type: CIP Generic

Service Type: Set Attribute Single

Service Code: 10 (Hex) Class: 71 (Hex)

Instance: 1 Attribute: 4 (Hex)

Source Element: Com1_ConRxSeq

Source Length: 2 (Bytes)

Destination:

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

Error Path:

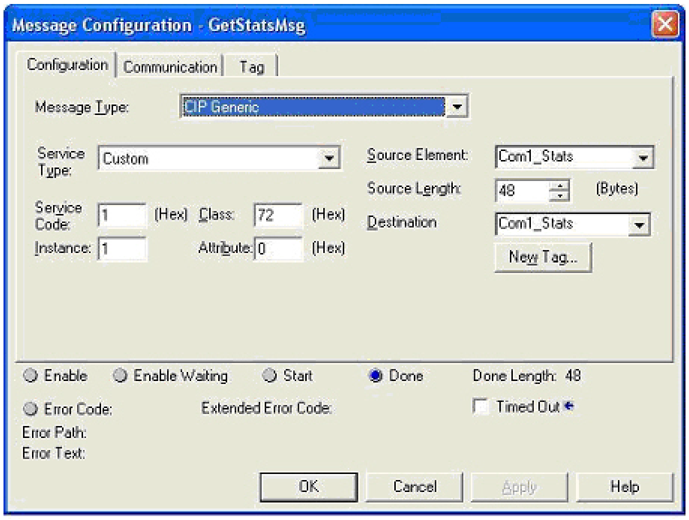
Error Text:

Provide the following information:

Field	Selection
Message Type	CIP Generic
Service Type	Set Attribute Single
Service Code	10 Hex (Set Attribute Single)
Class	71 Hex (Serial Port Data Transfer object)
Instance	1 (Port 1)
Attribute	4 (Receive consumed sequence number attribute)
Source Element	**Com1_ConRxSeq** (one INT)
Source Length	2 (one INT)

6.4.7.5. Request Statistics from DeviceMaster

Use the **Configuration** tab in the **Message Configuration** dialog for requesting statistics for a specified port from the DeviceMaster.



Message Type: CIP Generic

Service Type: Custom

Service Code: 1 (Hex) Class: 72 (Hex)

Instance: 1 Attribute: 0 (Hex)

Source Element: Com1_Stats

Source Length: 48 (Bytes)

Destination: Com1_Stats

○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 48

○ Error Code: Extended Error Code: □ Timed Out

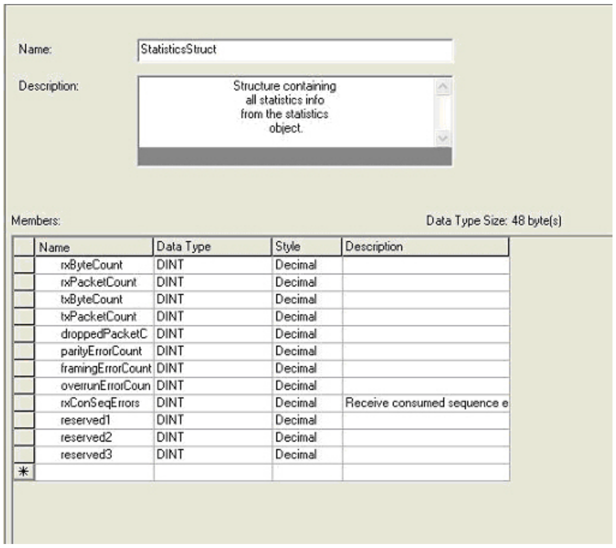
Error Path:

Error Text:

Provide the following information.

Field	Selection
Message Type	CIP Generic
Service Type	Custom
Service Code	1 Hex (Get Attribute All)
Class	72 Hex (Serial Port Statistics object)
Instance	1 (Port 1)
Attribute	0 (Unused)
Source Element	Com1_Stats (not actually used)
Source Length	48 (size of Statistics data structure)
Destination	**Com1_Stats** (Statistics data structure)

The following image displays the user defined data type for statistics.



Name: StatisticsStruct

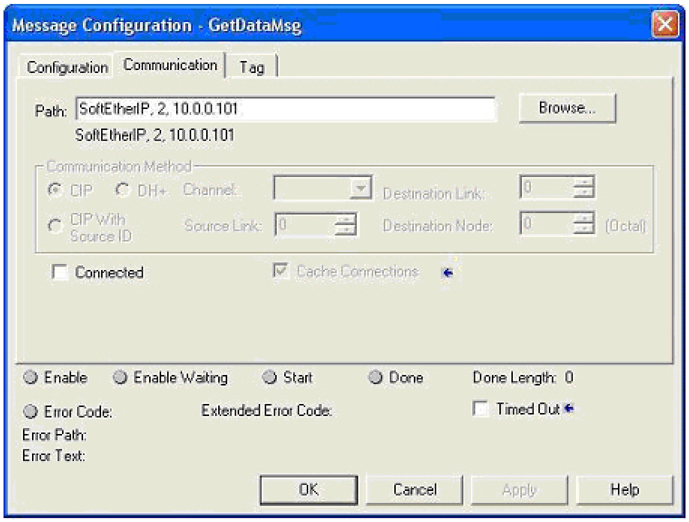
Description: Structure containing all statistics info from the statistics object.

Members: Data Type Size: 48 byte(s)

	Name	Data Type	Style	Description
	rxByteCount	DINT	Decimal	
	rxPacketCount	DINT	Decimal	
	txByteCount	DINT	Decimal	
	txPacketCount	DINT	Decimal	
	droppedPacketC	DINT	Decimal	
	parityErrorCount	DINT	Decimal	
	framingErrorCount	DINT	Decimal	
	overrunErrorCoun	DINT	Decimal	
	rxConSeqErrors	DINT	Decimal	Receive consumed sequence e
	reserved1	DINT	Decimal	
	reserved2	DINT	Decimal	
	reserved3	DINT	Decimal	
*				

6.4.7.6. Communication Window for all Messages sent to the DeviceMaster

Use the **Communication** tab in the **Message Configuration** dialog for sending all EtherNet/IP messages to the DeviceMaster.



Message Configuration - GetDataMsg

Path: SoftEtherIP, 2, 10.0.0.101

SoftEtherIP, 2, 10.0.0.101

Communication Method: ● CIP ○ DH+ ○ CIP With Source ID

Channel: Destination Link: 0

Source Link: 0 Destination Node: 0 (Octal)

□ Connected ☑ Cache Connections

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

Provide the following information.

Field	Selection
Path	Provide the following information in this box: • **SoftEtherIP** - Specifies the name of the EtherNet/IP interface • **2** - Required for routing on ControlLogix PLCs. • **10.0.0.101** - Specifies the IP address on the DeviceMaster used to create the example programs.

6.4.8. Modifying an RSLogix 5000 PLC Program Example (Older Versions)

The EtherNet/IP PLC example programs included in the firmware software assembly (**.msi**) file were developed on version 13.03 of RSLogix 5000 and may not load properly into older versions of RSLogix 5000. You can use the following procedure to modify a PLC example program for older versions of RSLogix 5000 when the PLC example program does not load properly.

1. Start RSLogix 5000 and load a known functional PLC program. (Preferably one with an EtherNet/IP interface.)
2. Create an **.L5K** file by saving the file to the **.L5K** format.
3. Open the known functional **.L5K** file with a text editor.
4. Open the example **.L5K** file with a text editor.
5. Make the following modifications to the **.L5K** file:
   - Modify the version number (**IE_VER**) of the example **.L5K** file to match the version number of the known functional **.L5K** file (line 8 of the example **.L5K** file).
   - Modify the **Major** revision number of the example **.L5K** file to match the **Major** revision number in the known functional **.L5K** file (line 11 of the example **.L5K** file).
6. Load the example **.L5K** file into RSLogix 5000.
   - If it loads without errors, follow the appropriate process to modify the example program to run on your system. The following topics describes how to modify the example **.L5K** files:
     - *6.4.3. loopbackExampleTagWrite.L5K* on Page 154
     - *6.4.4. loopbackExampleTagWriteSynced.L5K* on Page 155

  - *6.4.5. loopbackExamplePolling.L5K* on Page 156
- If you still have problems loading the example **.L5K** file, go to the next step.

7. Make the following modifications to the **.L5K** file:
   - Modify the **ProcessorType** setting in the example **.L5K** file to match the **ProcessorType** in the known functional **.L5K** file (line 10 of the example **.L5K** file).
   - Replace the **MODULE local** section of the example **.L5K** file with the **Module local** section in the known functional **.L5K** file (lines 89 to 102).
   - Choose one of the following options:
     - Replace the **MODULE EnetBridge** section of the example **.L5K** file with the EtherNet/IP interface section from the known functional **.L5K** file.
     - Delete the **MODULE EnetBridge** section from the example file (lines 104 to 117).
8. Load the example **.L5K** file into RSLogix 5000.
   - If it loads without errors, follow the appropriate process to modify the example program to run on your system. The following topics describes how to modify the example**.L5K** files:
     - *6.4.3. loopbackExampleTagWrite.L5K* on Page 154
     - *6.4.4. loopbackExampleTagWriteSynced.L5K* on Page 155
     - *6.4.5. loopbackExamplePolling.L5K* on Page 156
   - If you still have problems loading the example**.L5K** file, remove or modify all references to **EnetBridge** in the example **.L5K** file and re-load the example **.L5K** file into RSLogix 5000.

6.5. SLC or MicroLogix PLC Programming Example Instructions

This topic describes how to use RSLogix 500 to configure and run the DeviceMaster in an SLC or MicroLogix PLC environment.

You can configure the RSLogix 500 PLC program examples to your site's requirements. These programs are included in the self-installing file (**.MSI**) and are copied to the **Comtrol/EtherNetIP** directory on your computer when you open the **.MSI** file and follow the prompts. The self-installing file includes the following RSLogix 500 PLC program examples:

- *6.5.3.1. lpbkExampleSlcMsgPollRS500 - SLC PLC*
- *6.5.3.2. lpbkExamplePlc5MsgPollRS500 - SLC PLC*

These program examples are intended to aid the PLC programmer. These program examples were developed with version 6.30.00 (CPR 6) of RSLogix 500 and a C series SLC 5/05 with FRN 9 firmware.

Note: *The PLC program examples are designed to interface with a* DeviceMaster *1-port or on Port 1 of a 2-port or 4-port. Additional programming is required to use all ports on a 2-port or 4-port.*

Caution

***Disclaimer:* Comtrol supplies example PLC programs for demonstration purposes only. They are intended for the sole purpose of an example loop-back demonstration in a controlled lab environment. They are not intended for use in a production environment and may not function correctly on all PLCs. Comtrol does not warrant these example programs or any part thereof. The user assumes all liability for any modification to and use of a modified example program.**

6.5.1. What is RSLogix 500?

RSLogix 500 is a Windows ladder logic programming package for the SLC 500 and MicroLogix PLCs.

Note: *See the RSLogix 500 Help for more information on this product.*

6.5.2. Requirements

- The DeviceMaster must be installed and configured as described in the *EtherNet/IP Hardware Installation and Configuration Guide*.
- The DeviceMaster must be installed on the same Ethernet network segment as the PLC.
- RSLogix 500 must be installed on your computer. Note that the instructions in this guide require that you have some familiarity with this programming application.
- A loopback plug is required for the first port on the DeviceMaster when running an example PLC program. If necessary, see the *EtherNet/IP Hardware Installation and Configuration Guide* for information on loopback plugs.
- The PLC program examples (**.SLC**, **.SY5** and **.SY6** files) are optional. You can copy the PLC program examples from the CD or download the latest program examples from the Internet. See 1.4.1. Locating the Latest Software and Documents on Page 8, for the location of the PLC program examples.

6.5.3. Example Program Considerations

- While the RSLogix example programs are simple in nature, they include retry mechanisms for timed-out messages. You may or may not want to include the time-out mechanism in your application.
- While the receive and transmit sequence numbers are cleared on the DeviceMaster at the start of the programs, the only requirement is that the sequence numbers be in sync between the PLC and DeviceMaster.
- Statistics retrieval is not included in the example programs, but you can easily add it by inserting a request statistics message.

6.5.3.1. lpbkExampleSlcMsgPollRS500 - SLC PLC

This example program demonstrates an RSLogix 500 loopback PLC program using the SLC Typed messages in the *Polling receive* method. This program initializes receive and transmit produced data sequence numbers at startup on the DeviceMaster and then loops data through a loopback plug on the serial port. The SLC Typed Write data messages transmit the data, the SLC Typed Read Data messages receive the data and the sequence numbers are incremented.

This example program includes the following files:

- **lpbkExampleSlcMsgPollRS500.SLC** - Ladder logic in ASCII format.
- **lpbkExampleSlcMsgPollRS500.SY5** - Symbol definitions for RSLogix 500 Version 5.xx.xx.
- **lpbkExampleSlcMsgPollRS500.SY6** - Symbol definitions for RSLogix 500 Version 6.xx.xx.

6.5.3.2. lpbkExamplePlc5MsgPollRS500 - SLC PLC

This example program demonstrates an RSLogix 500 loopback PLC program using the PLC-5 Typed messages in the *Polling receive* method. This program initializes receive and transmit produced data sequence numbers at startup on the DeviceMaster and then loops data through a loopback plug on the serial port. The PLC-5 Typed Write data messages transmit the data, the PLC-5 Typed Read Data messages receive the data and the sequence numbers are incremented.

This example program includes the following files:

- **lpbkExamplePlc5MsgPollRS500.SLC** - Ladder logic in ASCII format.
- **lpbkExamplePlc5MsgPollRS500.SY5** - Symbol definitions for RSLogix 500 Version 5.xx.xx.
- **lpbkExamplePlc5MsgPollRS500.SY6** - Symbol definitions for RSLogix 500 Version 6.xx.xx.

6.5.3.3. lpbkExampleSlcMsgFileRS500 - SLC PLC

This example program demonstrates a loop-back RSLogix 500 PLC program using SLC Typed messages in the *Write-to-File receive* method. This program initializes the produced receive and transmit data sequence numbers at startup and then loops data via a loop-back plug on the serial port. The data is transmitted via SLC Typed Write data messages and received automatically via a write to file message from the DeviceMaster. The sequence numbers are incremented with each message.

The following files apply:

- **lpbkExampleSlcMsgFileRS500.SLC** – ladder logic in ASCII form
- **lpbkExampleSlcMsgFileRS500.SY5** – symbol definitions for RSLogix 500 Version 5.xx.xx.
- **lpbkExampleSlcMsgFileRS500.SY6** – symbol definitions for RSLogix 500 Version 6.xx.xx.

6.5.3.4. lpbkExampleSlcMsgFileSyncRS500 - SLC PLC

This example program demonstrates a loop-back RSLogix 500 PLC program using SLC Typed messages in the *Write-to-File-Synced receive* method. This program initializes the produced receive and transmit sequence numbers as well as the consumed receive sequence number at startup and then loops data via a loop-back plug on the serial port. The data is transmitted via SLC Typed Write data messages and received automatically via a write to file message from the DeviceMaster. The consumed receive sequence number is updated to match the produced receive sequence number and sent to the DeviceMaster to complete the synchronization process. All sequence numbers are incremented with each message.

The following files apply:

- **lpbkExampleSlcMsgFileSyncRS500.SLC** – ladder logic in ASCII form
- **lpbkExampleSlcMsgFileSyncRS500.SY5** – symbol definitions for RSLogix 500 Version 5.xx.xx.
- **lpbkExampleSlcMsgFileSyncRS500.SY6** – symbol definitions for RSLogix 500 Version 6.xx.xx.

6.5.3.5. LPBKEXAMPLESLCMSGFILERS500_MICROLGX - MicroLogix PLC

This example program demonstrates a loop-back RSLogix 500 PLC program using SLC Typed messages in the *Write-to-File receive* method. This program initializes the produced receive and transmit data sequence numbers at startup and then loops data via a loop-back plug on the serial port. The data is transmitted via SLC Typed Write data messages and received automatically via a write to file message from the DeviceMaster. The sequence numbers are incremented with each message.

The following file applies: **LPBKEXAMPLESLCMSGFILERS500_MICROLGX.RSS** , a MicroLogix PLC example program.

6.5.3.6. LPBKEXAMPLESLCMSGPOLLRS500_MICROLGX - MicroLogix PLC

This example program demonstrates an RSLogix 500 loopback PLC program using the SLC Typed messages in the *Polling receive* method. This program initializes receive and transmit produced data sequence numbers at startup on the DeviceMaster and then loops data through a loopback plug on the serial port. The SLC Typed Write data messages transmit the data, the SLC Typed Read Data messages receive the data and the sequence numbers are incremented.

The following file applies: **LPBKEXAMPLESLCMSGPOLLRS500_MICROLGX.RSS**, a MicroLogix PLC example program.

6.5.4. Configure the DeviceMaster for the RSLogix 500 Example Program - SLC PLC

The following procedure configures the DeviceMaster for the RSLogix 500 example programs. You must perform this task before you configure and run the RSLogix 500 example program. For more information on the *Port Configuration* web pages, see *Chapter 4. Embedded Configuration Pages* on Page 85.

1. Attach a loopback plug to the serial port.
2. Access the *Port Configuration* web page, using one of these methods.
   - Open PortVision DX, right-click the DeviceMaster for which you want to program network information and click **Webpage.**
   - Open a browser and type the IP address for the DeviceMaster in the **Address** box.
3. Click **Serial | Port *n***. Where ***n*** is the port number.
4. Set the serial port settings under **Serial Configuration** to the following values.

Field	Setting
Mode	RS-232
Baud	57600
Parity	none
Data Bits	8
Stop Bits	1
Flow Control	none
DTR	off
Rx Timeout Between Packets	200

5. Click the **Save** button.
6. Set the serial port settings under **Serial Packet Identification** to these values.

Field	Setting
STX RX Detect	Set to one byte and Byte 1 to 2.
ETX Rx Detect	Set to one byte and Byte 1 to 3.
STX Tx Append	Set to one byte and Byte 1 to 2.
ETX Tx Append	Set to one byte and Byte 1 to 3.
Strip Rx STX/ETX	Select
Discard Rx Packets With Errors	Select
(PLC-5/SLC) Rx MS Byte First	Optionally, select
(PLC-5/SLC) Tx MS Byte First	Optionally, select

7. Click **EtherNet/IP Settings** and set the serial port settings to the following values:

Field	Selection
TX Sequence Number Checking	Select.
Rx (To PLC) Ethernet Transfer Method	• Set to **Polling for lpbkExampleSlcMsgPollRS500** and **lpbkExamplePlc5MsgPollRS500**. • Set to Write-to-Tag/File for **lpbkExampleSlcMsgFileRS500**. • Set to Write-to-Tag/File-Synced for **lpbkExampleSlcMsgFileSyncRS500**.
PLC IP Address	• Leave blank for Polling. • Set to IP Address of PLC for Write-to-File and Write-to-File-Synced.
PLC Controller Slot Number	Unused and can remain blank.
Rx (To PLC) Produced Data Tag/File Name	• Leave blank for Polling. • For SLC PLCs, set to **$N10:0** and for MicroLogix PLCs, set to **#N10:0**; the PLC receive filename for *Write-to-File* and *Write-to-File-Synced*.

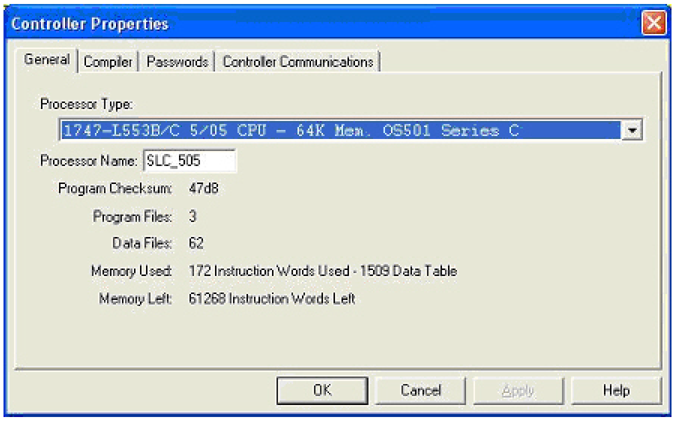
8. Click the **Save** button.

6.5.5. Configure and Run the RSLogix 500 Example Program - SLC PLC

You can configure and run the RSLogix 500 example programs through RSLogix 500. For additional information on the RSLogix 500, see 6.5.6. RSLogix 500 Screen Examples - SLC PLC on Page 173.

Note: *Configure the* DeviceMaster *before you configure and download the RSLogix 500 example program. For instructions on configuring* DeviceMaster, *see 6.5.4. Configure the DeviceMaster for the RSLogix 500 Example Program - SLC PLC, earlier in this chapter.*

1. Select the appropriate message type example programs (SLC or PLC-5 typed messages) and copy the files (**.SLC**, **.SY5** and **.SY6**) to the desired directory.
2. Start RSLogix 500 and open the **.SLC** file through RSLogix 500.
3. To modify the PLC program for your system, double-click **Controller Properties**.
4. In the **General** tab, select your SLC processor type under **Processor Type** and provide a **Processor Name**.



Processor Type: 1747-L553B/C 5/05 CPU - 64K Mem. OS501 Series C

Processor Name: SLC_505

Program Checksum: 47d8

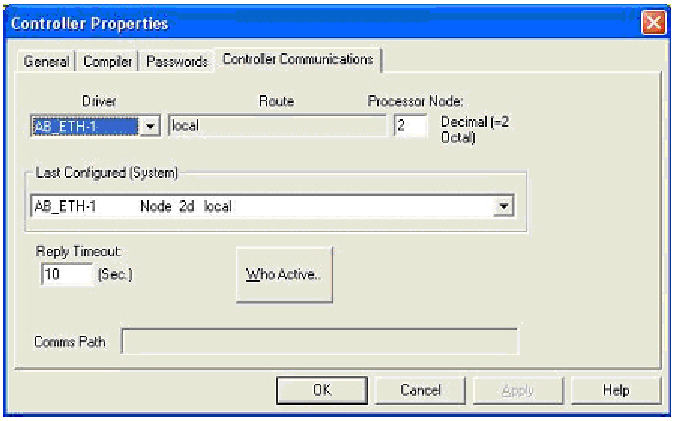
Program Files: 3

Data Files: 62

Memory Used: 172 Instruction Words Used - 1509 Data Table

Memory Left: 61268 Instruction Words Left

5. Select the **Controller Communications** tab and select the following options:
   a. Set **Driver** to the appropriate type to allow RSLogix 500 to communicate with the SLC processor.



Driver: AB_ETH-1

Route: local

Processor Node: 2 Decimal (=2 Octal)

Last Configured (System): AB_ETH-1 Node 2d local

Reply Timeout: 10 (Sec.)

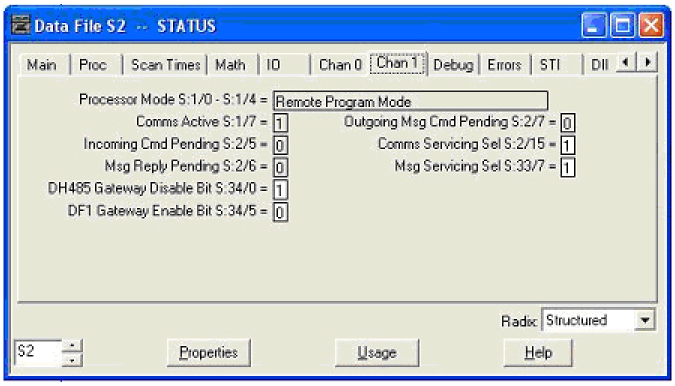
Comms Path:

   b. Type the processor node number in the **Processor Node** box. (You may reference the **Last Configured (System)** node or select **Who Active**.)
6. Click **OK** to apply your changes and close the **Controller Properties** dialog.
7. Double-click **Processor Status**, select the **Chan 1** tab on the **Data File S2 -- STATUS** dialog and verify the following:
   a. **DH485 Gateway Disable Bit S:34/0** option is set to **1**.
   b. **DF1 Gateway Enable Bit S:34/5** option is set to **0** (zero).
   c. **Comms Servicing Sel S:2/15** option is set to **1**.
   d. **Msg Servicing Sel S:33/7** option is set to **1**.
8. Close the **Data File S2 -- Status** dialog.
9. Double-click **I/O Configuration** and select your chassis type in the **Racks** panel.
10. Close the **I/O Configuration** dialog.

11. Double-click **Channel Configuration**, and select the following in the **General** tab:

    a. In the **Channel 1** panel, type 60 in the **Diagnostic File** box.

    b. In the **Channel 0** panel, type 61 in the **Diagnostic File** box.



Processor Mode S:1/0 - S:1/4 = Remote Program Mode

Comms Active S:1/7 = 1

Incoming Cmd Pending S:2/5 = 0

Msg Reply Pending S:2/6 = 0

DH485 Gateway Disable Bit S:34/0 = 1

DF1 Gateway Enable Bit S:34/5 = 0

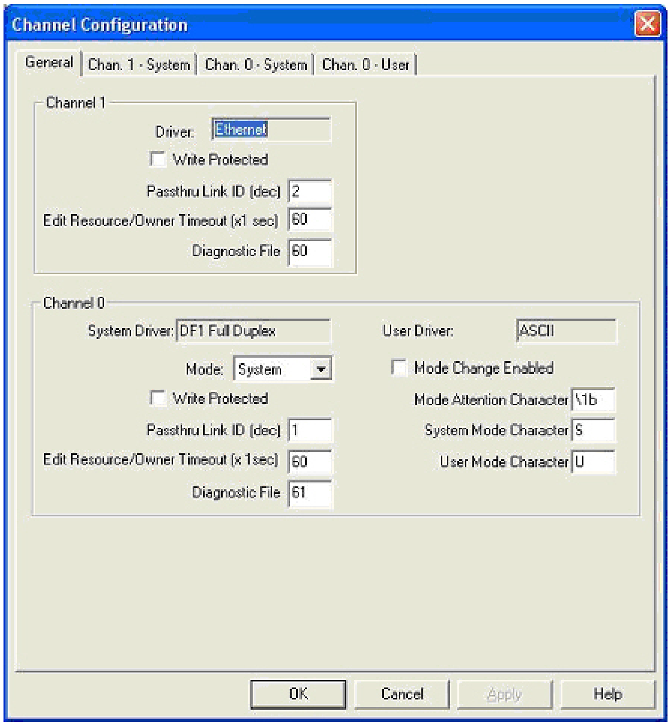
Outgoing Msg Cmd Pending S:2/7 = 0

Comms Servicing Sel S:2/15 = 1

Msg Servicing Sel S:33/7 = 1

Radix: Structured

S2



Channel 1

Driver: Ethernet

□ Write Protected

Passthru Link ID (dec) 2

Edit Resource/Owner Timeout (x1 sec) 60

Diagnostic File 60

Channel 0

System Driver: DF1 Full Duplex

Mode: System

□ Write Protected

Passthru Link ID (dec) 1

Edit Resource/Owner Timeout (x 1sec) 60

Diagnostic File 61

User Driver: ASCII

□ Mode Change Enabled

Mode Attention Character \1b

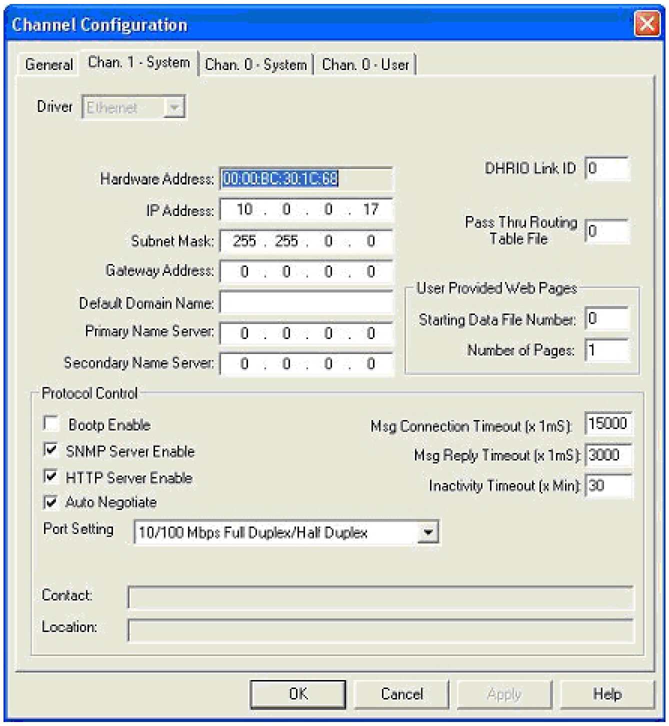
System Mode Character S

User Mode Character U

12. Select the **Chan. 1 System** tab on the **Channel Configuration** dialog and select the following options.
    a. Type the IP address for your PLC in the **IP Address** box if you are not using **Bootp**.
    b. Type the subnet mask for your PLC in the **Subnet Mask** box.
    c. Type the gateway address for your PLC in the **Gateway Address** box.
    d. Select **Bootp Enable** if you are using Bootp to initialize your network settings.
    e. Select **SNMP Server Enable**.
    f. Select **HTTP Server Enable.**
    g. Select **Auto Negotiate** if your network is capable of Ethernet auto-negotiation.

       If you select **Auto Negotiate**, set the **Port Setting** to **10/100 Mbps Full Duplex/Half Duplex**.

       If you do not select **Auto Negotiate**, select the speed and duplex for your network connection.


Channel Configuration

Driver: Ethernet

Hardware Address: 00:00:BC:30:1C:68

IP Address: 10 . 0 . 0 . 17

Subnet Mask: 255 . 255 . 0 . 0

Gateway Address: 0 . 0 . 0 . 0

Default Domain Name:

Primary Name Server: 0 . 0 . 0 . 0

Secondary Name Server: 0 . 0 . 0 . 0

DHRIO Link ID: 0

Pass Thru Routing Table File: 0

User Provided Web Pages

Starting Data File Number: 0

Number of Pages: 1

Protocol Control

☐ Bootp Enable

☑ SNMP Server Enable

☑ HTTP Server Enable

☑ Auto Negotiate

Port Setting: 10/100 Mbps Full Duplex/Half Duplex

Msg Connection Timeout (x 1mS): 15000

Msg Reply Timeout (x 1mS): 3000

Inactivity Timeout (x Min): 30

Contact:

Location:

13. Click **OK** to apply your changes and close the **Channel Configuration** dialog.
14. In the ladder logic, double-click **Setup Screen** in an **MSG** instruction.
15. Select the **MultiHop** tab and make the following changes.
    a. On the first line, type the IP address for the DeviceMaster in the **To Address** box.
    b. If you are using an SLC 5/03 or 5/04 with an EtherNet/IP sidecar, you may need to add additional hops here.
16. Close the **MSG** dialog.
17. Repeat steps 14 through 16 for each **MSG** instruction in the ladder logic.
18. Download the PLC program to your PLC and run the program.

6.5.6. RSLogix 500 Screen Examples - SLC PLC

The following subsections explain how to configure the DeviceMaster through RSLogix 500. Use the screens to set up the PLC and program the various messages.

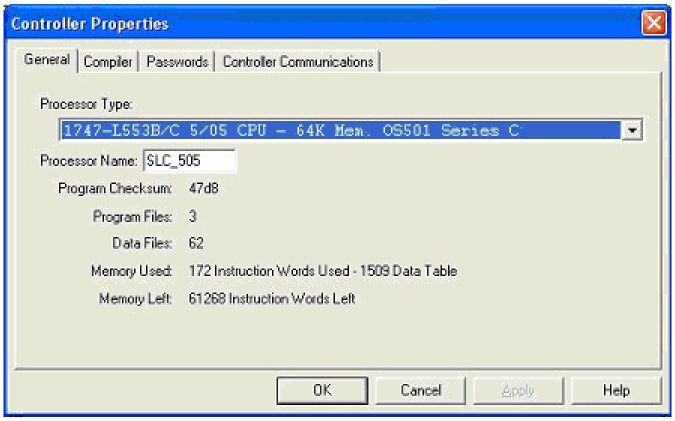
6.5.6.1. Setting up Processor and Ethernet (Channel 1)

You must set up the Processor and Ethernet communication port properly for EtherNet/IP to function. Read the information and follow the procedures provided in the following Rockwell documents:

- SLC 5/03, 5/04 and 5/05 Modular Processors Installation Instructions (publication 1747-IN009D-MU-P)
- SLC 500 Instruction Set (publication 1747-RM001D-EN-P, pages 13-22 through 13-47).

The following screens show the recommended settings that allow EtherNet/IP to function properly on an SLC or MicroLogix PLC.

1. Start **RSLogix 500**.
2. Double-click **Controller Properties** and select the correct processor type from the **General** tab on the **Controller Properties** dialog.



Controller Properties

Processor Type: 1747-L553B/C 5/05 CPU - 64K Mem. OS501 Series C

Processor Name: SLC_505

Program Checksum: 47d8

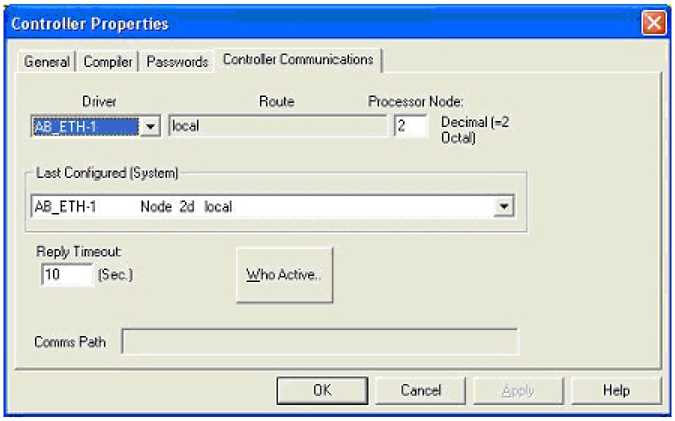
Program Files: 3

Data Files: 62

Memory Used: 172 Instruction Words Used - 1509 Data Table

Memory Left: 61268 Instruction Words Left

3. Select the **Controller Communications** tab and select the proper driver for RSLogix 500.



Controller Properties

Driver: AB_ETH-1

Route: local

Processor Node: 2 Decimal (=2 Octal)

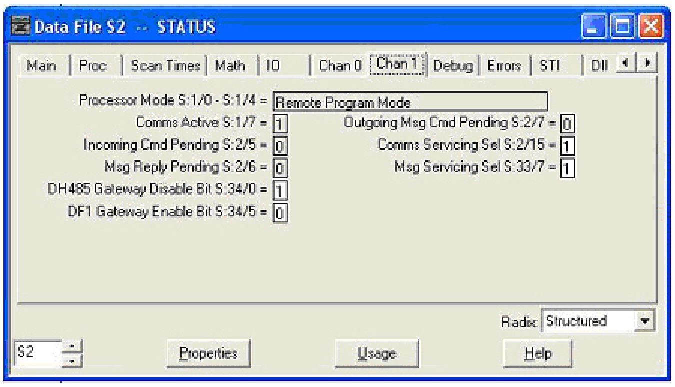
Last Configured (System): AB_ETH-1 Node 2d local

Reply Timeout: 10 (Sec.)

Comms Path:

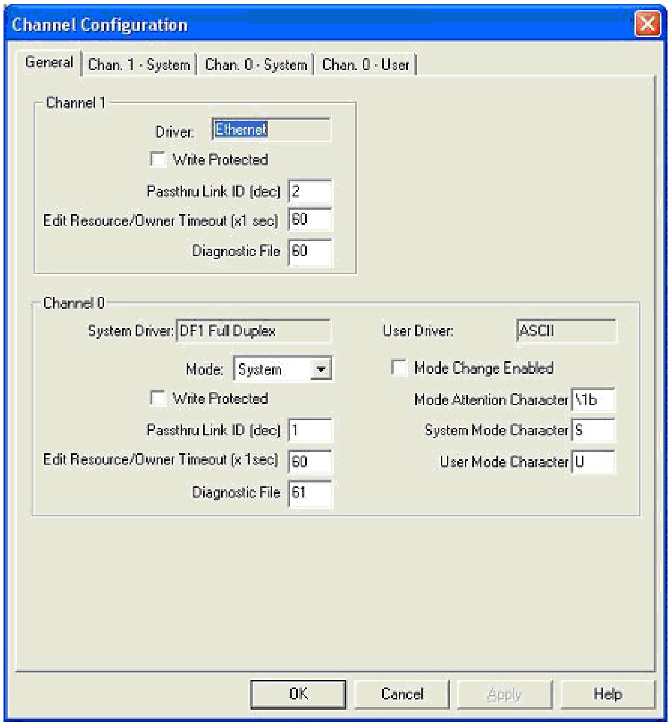
4. Click **OK** to apply your changes and close the **Controller Properties** dialog.

5. Double-click **Processor Status**, and select the **Chan 1** tab on the **Data File S2 -- STATUS** dialog.
6. Make the following recommended changes.
   a. Select the **DH485 Gateway Disable Bit S:34/0** option.
   b. Clear the **DF1 Gateway Enable BIT S:34/5** option.
   c. Select the **Comms Servicing Sel S:2/15** option.
   d. Select the **Msg Servicing Sel S:33/7** option. (You must enable this option if you want to run EtherNet/IP.)



Processor Mode S:1/0 - S:1/4 = Remote Program Mode
Comms Active S:1/7 = 1
Incoming Cmd Pending S:2/5 = 0
Msg Reply Pending S:2/6 = 0
DH485 Gateway Disable Bit S:34/0 = 1
DF1 Gateway Enable Bit S:34/5 = 0
Outgoing Msg Cmd Pending S:2/7 = 0
Comms Servicing Sel S:2/15 = 1
Msg Servicing Sel S:33/7 = 1
Radix: Structured

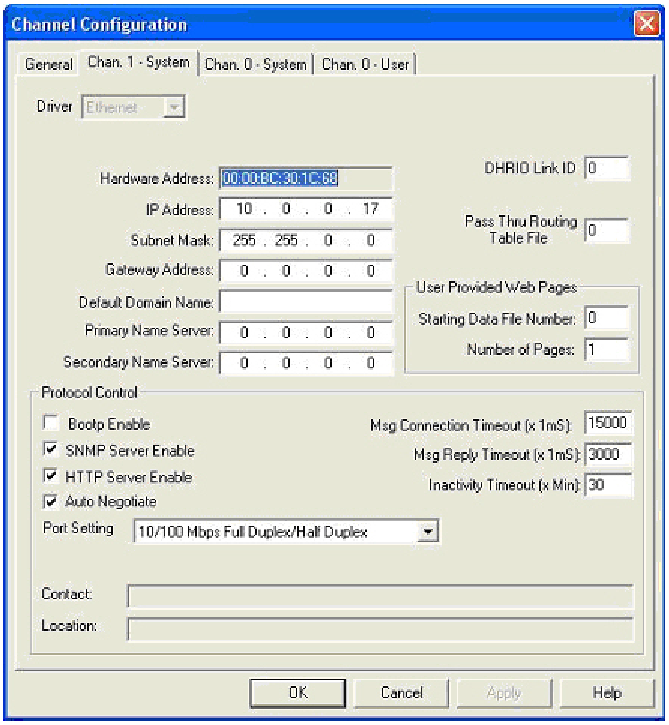
7. Optionally, double-click **Channel Configuration** and type a value (between 0 and 256) in the **Diagnostic File** box for an integer diagnostic file. You can use the diagnostic file to help solve any network-related problems.



Channel 1
Driver: Ethernet
□ Write Protected
Passthru Link ID (dec) 2
Edit Resource/Owner Timeout (x1 sec) 60
Diagnostic File 60

Channel 0
System Driver: DF1 Full Duplex
Mode: System
□ Write Protected
Passthru Link ID (dec) 1
Edit Resource/Owner Timeout (x 1sec) 60
Diagnostic File 61
User Driver: ASCII
□ Mode Change Enabled
Mode Attention Character \1b
System Mode Character S
User Mode Character U

8. Select the **Chan. 1 System** tab on the **Channel Configuration** dialog.
9. Make the following recommended changes.
   a. Select the **SNMP server Enable** option. (EtherNet/IP may not function without this setting.)
   b. Select the **HTTP Server Enable** option. (EtherNet/IP may not function without this setting.)
   c. To automatically select the proper Ethernet speed and duplex settings:
      - Select the **Auto Negotiate** option.
      - Select the **10/100 Mbps Full Duplex/Half Duplex** option.
10. Click **OK** to apply your changes and close the **Channel Configuration** dialog.



Driver: Ethernet

Hardware Address: 00:00:BC:30:1C:68

IP Address: 10 . 0 . 0 . 17

Subnet Mask: 255 . 255 . 0 . 0

Gateway Address: 0 . 0 . 0 . 0

Default Domain Name:

Primary Name Server: 0 . 0 . 0 . 0

Secondary Name Server: 0 . 0 . 0 . 0

DHRIO Link ID: 0

Pass Thru Routing Table File: 0

User Provided Web Pages

Starting Data File Number: 0

Number of Pages: 1

Protocol Control

□ Bootp Enable

☑ SNMP Server Enable

☑ HTTP Server Enable

☑ Auto Negotiate

Port Setting: 10/100 Mbps Full Duplex/Half Duplex

Msg Connection Timeout (x 1mS): 15000

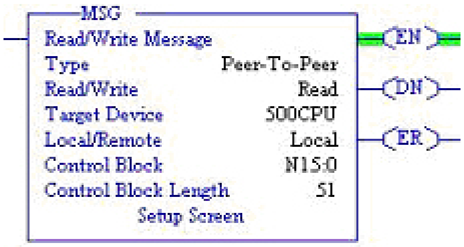
Msg Reply Timeout (x 1mS): 3000

Inactivity Timeout (x Min): 30

Contact:

Location:

6.5.6.2. SLC Typed Read - Receive Data Message - SLC PLC



MSG
Read/Write Message
Type Peer-To-Peer
Read/Write Read
Target Device 500CPU
Local/Remote Local
Control Block N15:0
Control Block Length 51
Setup Screen

The following screen depicts an *SLC Typed Read - Receive Data* message in ladder logic.

Make the following changes to the ladder logic.

1. Select the **Read** option.
2. Select the **500CPU** option.
3. Select **Local**.
4. Assign a dedicated integer file of 51 integers to the **Control** block.
5. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
6. Make the following changes:

   a. Specify the file address to receive data information in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.

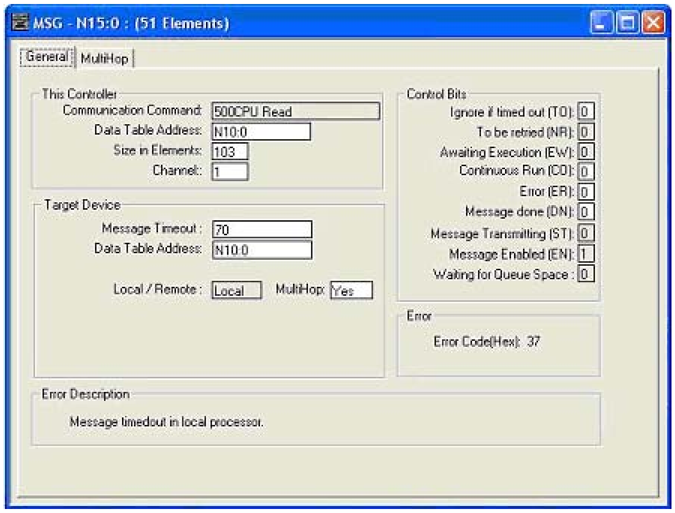
   b. Specify a size in the **Size of Elements** box that is large enough to receive the entire data message including the sequence number and length fields.

   c. Set the **Channel** parameter to **1** to use the Ethernet port.

   The **Message Timeout** parameter is not actually configurable. The RSLogix 500 sets the value in this box based on the Ethernet timeout settings.

   d. Specify the port-specific read file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.

   e. Set the **MultiHop** option to **Yes**.



MSG - N15:0 : (51 Elements)

This Controller
Communication Command: 500CPU Read
Data Table Address: N10:0
Size in Elements: 103
Channel: 1

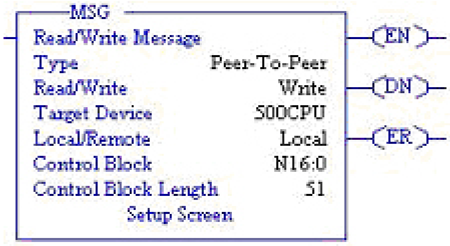
Target Device
Message Timeout : 70
Data Table Address: N10:0
Local / Remote : Local
MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 1
Waiting for Queue Space : 0

Error
Error Code(Hex): 37

Error Description
Message timedout in local processor.

6.5.6.3. SLC Typed Write - Transmit Data Message - SLC PLC



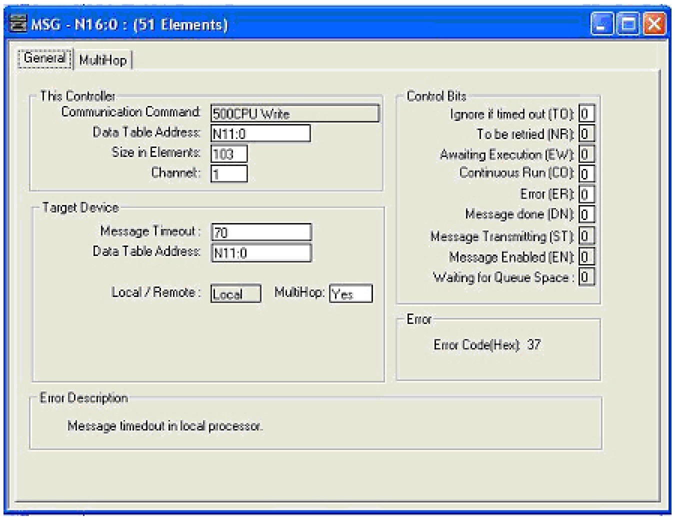
MSG
Read/Write Message
Type Peer-To-Peer
Read/Write Write
Target Device 500CPU
Local/Remote Local
Control Block N16:0
Control Block Length 51
Setup Screen

The following screen depicts an *SLC Typed Write - Transmit Data* message in ladder logic.

Make the following changes to the ladder logic.

1. Select the **Write** option.
2. Select the **500CPU** option.
3. Select **Local**.
4. Assign a dedicated integer file of 51 integers to the **Control** block.
5. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
6. Make the following changes:
   a. Specify the file address to transmit data information in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
   b. Specify a size in the **Size of Elements** box that is large enough to send the entire data message including the sequence number and length fields.

      Set the **Channel** parameter to **1** to use the Ethernet port.
   c. The **Message Timeout** parameter is not actually configurable. RSLogix 500 sets the value in this box based on the Ethernet timeout settings.
   d. Specify the port-specific transmit file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
   e. Set the **MultiHop** option to **Yes**.



MSG - N16:0 : (51 Elements)

This Controller
Communication Command: 500CPU Write
Data Table Address: N11:0
Size in Elements: 103
Channel: 1

Target Device
Message Timeout : 70
Data Table Address: N11:0
Local / Remote : Local
MultiHop: Yes

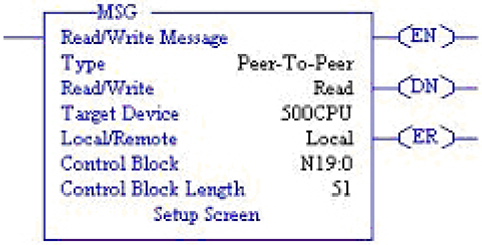
Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0
Waiting for Queue Space : 0

Error
Error Code(Hex): 37

Error Description
Message timedout in local processor.

6.5.6.4. SLC Typed Read - Retrieve Statistics Message - SLC PLC

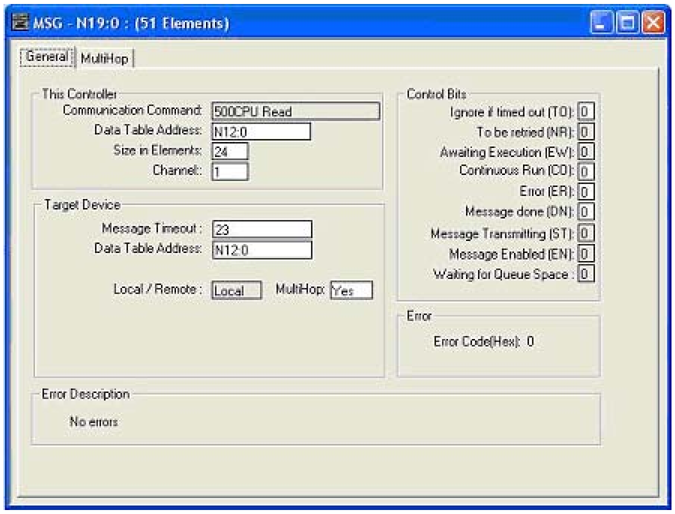
The following screen depicts an *SLC Typed Read - Retrieve Statistics* message in ladder logic.



MSG
Read/Write Message
Type Peer-To-Peer
Read/Write Read
Target Device 500CPU
Local/Remote Local
Control Block N19:0
Control Block Length 51
Setup Screen

Make the following changes to the ladder logic.

1. Select the **Read** option.
2. Select the **500CPU** option.
3. Select **Local**.
4. Assign a dedicated integer file of 51 integers to the **Control** block.
5. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.



MSG - N19:0 : (51 Elements)

This Controller
Communication Command: 500CPU Read
Data Table Address: N12:0
Size in Elements: 24
Channel: 1

Target Device
Message Timeout : 23
Data Table Address: N12:0
Local / Remote : Local
MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0
Waiting for Queue Space : 0

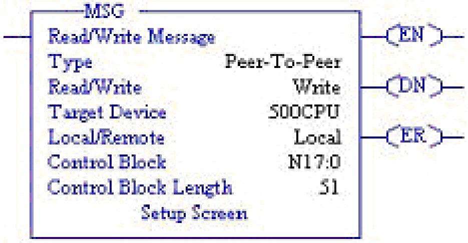
Error
Error Code(Hex): 0

Error Description
No errors

6. Make the following changes:
   a. Specify the file address to receive statistics data information in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
   b. Type **24** (twelve 32-bit integers) in the **Size of Elements** box.
   c. Set the **Channel** parameter to **1** to use the Ethernet port.

      The **Message Timeout** parameter is not actually configurable. RSLogix 500 sets the value in this box based on the Ethernet timeout settings.
   d. Specify the port-specific statistics file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
   e. Set the **MultiHop** option to **Yes**.

6.5.6.5. SLC Typed Write - Set Receive Produced Sequence Number Message - SLC PLC

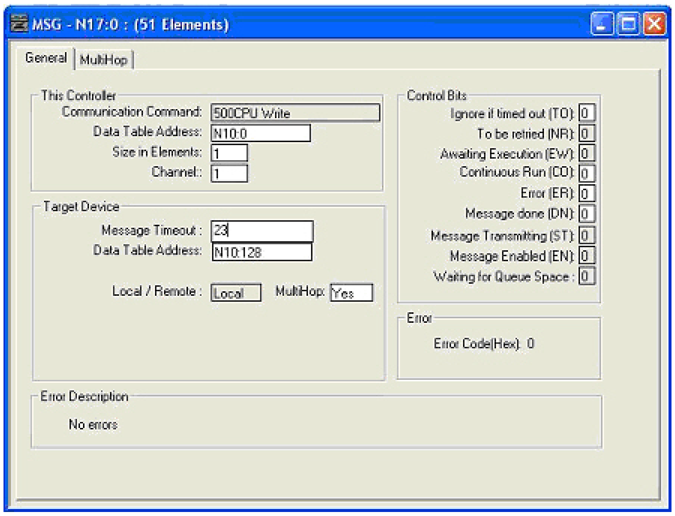


MSG
Read/Write Message
Type Peer-To-Peer
Read/Write Write
Target Device 500CPU
Local/Remote Local
Control Block N17:0
Control Block Length 51
Setup Screen

The following screen depicts an *SLC Typed Write - Set Receive Produced Sequence Number* message in ladder logic.

Make the following changes to the ladder logic.

1. Select the **Write** option.
2. Select the **500CPU** option.
3. Select **Local**.
4. Assign a dedicated integer file of 51 integers to the **Control** block.
5. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
6. Make the following changes.



MSG - N17:0 : (51 Elements)

This Controller
Communication Command: 500CPU Write
Data Table Address: N10:0
Size in Elements: 1
Channel: 1

Target Device
Message Timeout : 23
Data Table Address: N10:128
Local / Remote : Local
MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0
Waiting for Queue Space : 0

Error
Error Code(Hex): 0

Error Description
No errors

   a. Specify the file address where the receive sequence number resides in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.

   b. Type **1** in the **Size of Elements** box.

   c. Set the **Channel** parameter to **1** to use the Ethernet port.

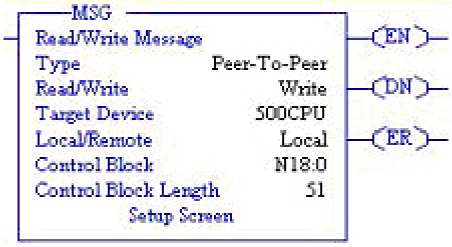
      The **Message Timeout** parameter is not actually configurable. RSLogix 500 sets the value in this box based on the Ethernet timeout settings.

   d. Specify the port-specific receive sequence file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.

   e. Set the **MultiHop** option to **Yes**.

6.5.6.6. SLC Typed Write - Set Transmit Produced Sequence Number Message - SLC PLC

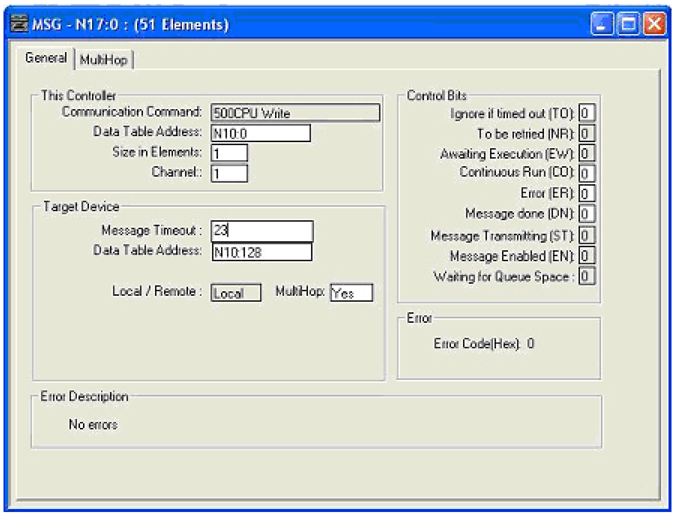
The following screen depicts an *SLC Typed Write - Set Transmit Produced Sequence Number* message in ladder logic.


MSG
Read/Write Message
Type Peer-To-Peer
Read/Write Write
Target Device 500CPU
Local/Remote Local
Control Block N18:0
Control Block Length 51
Setup Screen

Make the following changes to the ladder logic.

1. Select the **Write** option.
2. Select the **500CPU** option.
3. Select **Local**.
4. Assign a dedicated integer file of 51 integers to the **Control** block.
5. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
6. Make the following changes:
    a. Specify the file address where the transmit sequence number resides in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
    b. Type **1** in the **Size of Elements** box.
    c. Set the **Channel** parameter to **1** to use the Ethernet port.

       The **Message Timeout** parameter is not actually configurable. RSLogix 500 sets the value in this box based on the Ethernet timeout settings.
    d. Specify the port-specific transmit sequence number file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
    e. Set the **MultiHop** option to **Yes**.


MSG - N17:0 : (51 Elements)

This Controller
Communication Command: 500CPU Write
Data Table Address: N10:0
Size in Elements: 1
Channel:: 1

Target Device
Message Timeout : 23
Data Table Address: N10:128
Local / Remote : Local
MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0
Waiting for Queue Space : 0

Error
Error Code(Hex): 0

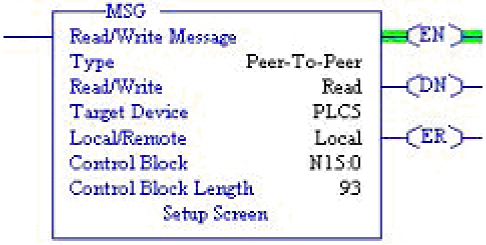
Error Description
No errors

6.5.6.7. PLC-5 Typed Read - Receive Data Message - SLC PLC

The following screen depicts a *PLC-5 Typed Read - Receive Data* message in ladder logic.

Make the following changes to the ladder logic.

1. Select the **Read** option.
2. Select the **PLC5** option.
3. Select **Local**.
4. Assign a dedicated integer file of 93 integers to the **Control** block.



MSG
Read/Write Message
Type: Peer-To-Peer
Read/Write: Read
Target Device: PLC5
Local/Remote: Local
Control Block: N15:0
Control Block Length: 93
Setup Screen

In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.

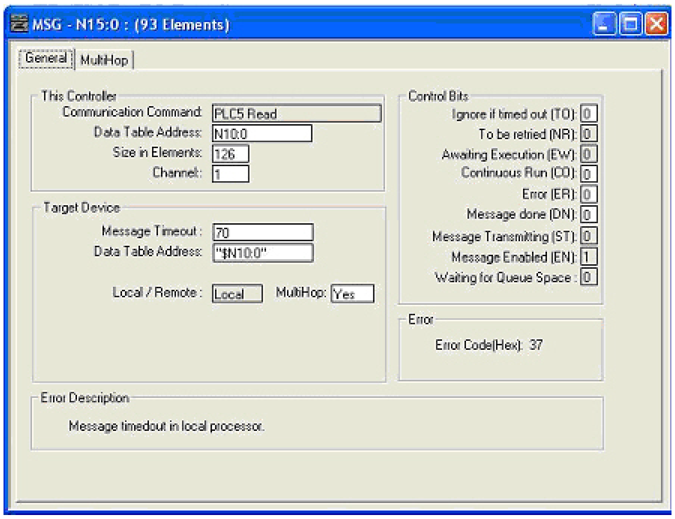
5. Make the following changes.

   a. Specify the file address to receive data information in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.

   b. Specify a size in the **Size of Elements** box that is large enough to receive the entire data message including the sequence number and length fields.

   c. Set the **Channel** parameter to **1** to use the Ethernet port.

   The **Message Timeout** parameter is not actually configurable. RSLogix 500 sets the value in this box based on the Ethernet timeout settings.



MSG - N15:0 : (93 Elements)

This Controller
Communication Command: PLC5 Read
Data Table Address: N10:0
Size in Elements: 126
Channel: 1

Target Device
Message Timeout : 70
Data Table Address: "$N10:0"
Local / Remote : Local
MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 1
Waiting for Queue Space : 0

Error
Error Code(Hex): 37

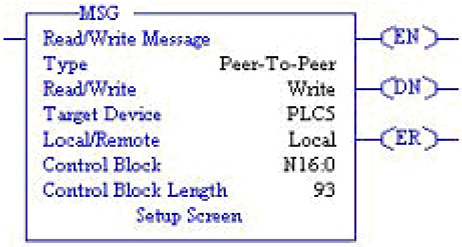
Error Description
Message timedout in local processor.

   d. Specify the port-specific receive file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. The receive file address must be specified in logical ASCII format. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.

   e. Set the **MultiHop** option to **Yes**.

6.5.6.8. PLC-5 Typed Write - Transmit Data Message - SLC PLC

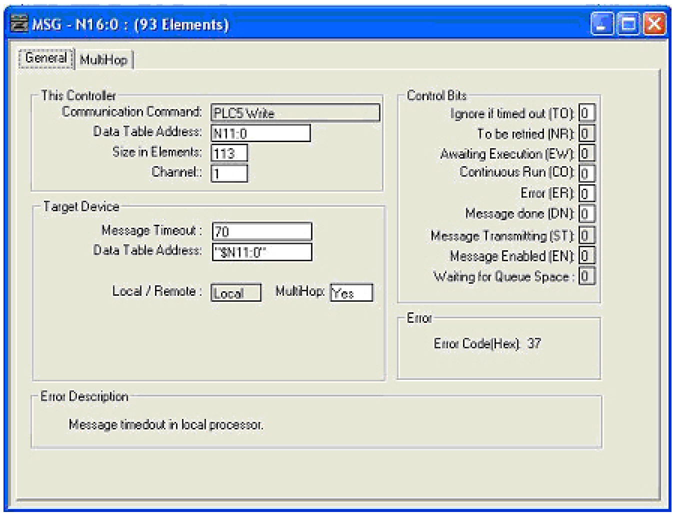
The following screen depicts a PLC-5 Typed Write - Transmit Data message in ladder logic.



MSG
Read/Write Message
Type Peer-To-Peer
Read/Write Write
Target Device PLC5
Local/Remote Local
Control Block N16:0
Control Block Length 93
Setup Screen

Make the following changes to the ladder logic.

1. Select the **Write** option.
2. Select the **PLC5** option.
3. Select **Local**.
4. Assign a dedicated integer file of 93 integers to the **Control** block.
5. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
6. Make the following changes:



MSG - N16:0 : (93 Elements)

This Controller
Communication Command: PLC5 Write
Data Table Address: N11:0
Size in Elements: 113
Channel:: 1

Target Device
Message Timeout : 70
Data Table Address: "$N11:0"
Local / Remote : Local MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0
Waiting for Queue Space : 0

Error
Error Code(Hex): 37

Error Description
Message timedout in local processor.

   a. Specify the file address to transmit data information in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.

   b. Specify a size in the **Size of Elements** box that is large enough to send the entire data message including the sequence number and length fields.

   c. Set the **Channel** parameter to **1** to use the Ethernet port.

      The **Message Timeout** parameter is not actually configurable. RSLogix 500 sets the value in this box based on the Ethernet timeout settings.

   d. Specify the port-specific transmit file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. The transmit file address must be specified in logical ASCII format. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.

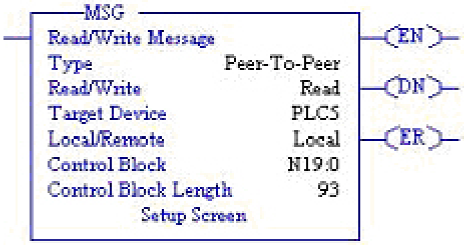
   e. Set the **MultiHop** option to **Yes**.

6.5.6.9. PLC-5 Typed Read - Retrieve Statistics Message - SLC PLC

The following screen depicts an *PLC-5 Typed Read - Retrieve Statistics* message in ladder logic.

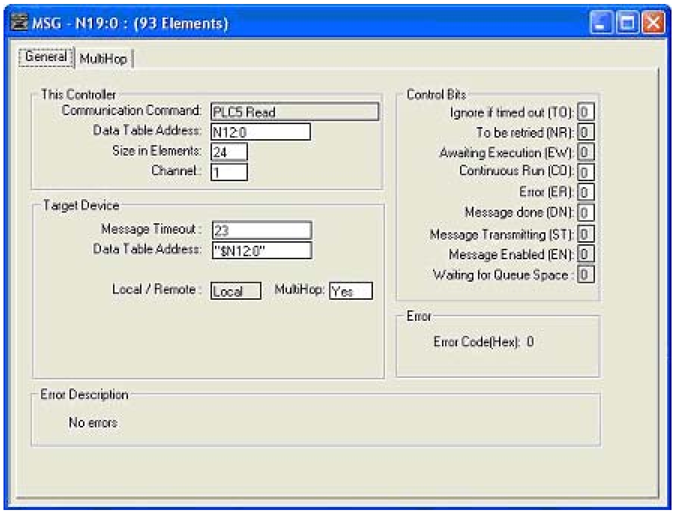
Make the following changes to the ladder logic.

1. Select the **Read** option.
2. Select the **PLC5** option.
3. Select **Local**.
4. Assign a dedicated integer file of 93 integers to the **Control** block.


MSG
Read/Write Message
Type Peer-To-Peer
Read/Write Read
Target Device PLC5
Local/Remote Local
Control Block N19:0
Control Block Length 93
Setup Screen

5. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
6. Make the following changes:
   a. Specify the file address to receive statistics data information in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
   b. Type **24** (twelve 32-bit integers) in the **Size of Elements** box.
   c. Set the **Channel** parameter to **1** to use the Ethernet port.

      The **Message Timeout** parameter is not actually configurable. RSLogix 500 sets the value in this box based on the Ethernet timeout settings.
   d. Specify the port-specific statistics file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. The statistics file address must be specified in logical ASCII format. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
   e. Set the **MultiHop** option to **Yes**.


MSG - N19:0 : (93 Elements)

This Controller
Communication Command: PLC5 Read
Data Table Address: N12:0
Size in Elements: 24
Channel: 1

Target Device
Message Timeout : 23
Data Table Address: "$N12:0"
Local / Remote : Local
MultiHop: Yes

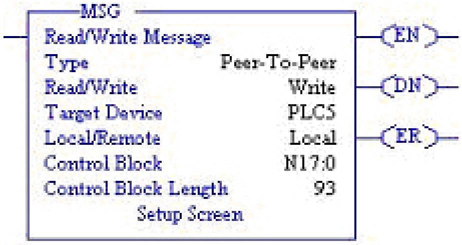
Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0
Waiting for Queue Space : 0

Error
Error Code(Hex): 0

Error Description
No errors

6.5.6.10. PLC-5 Typed Write - Set Receive Produced Sequence Number Message - SLC PLC

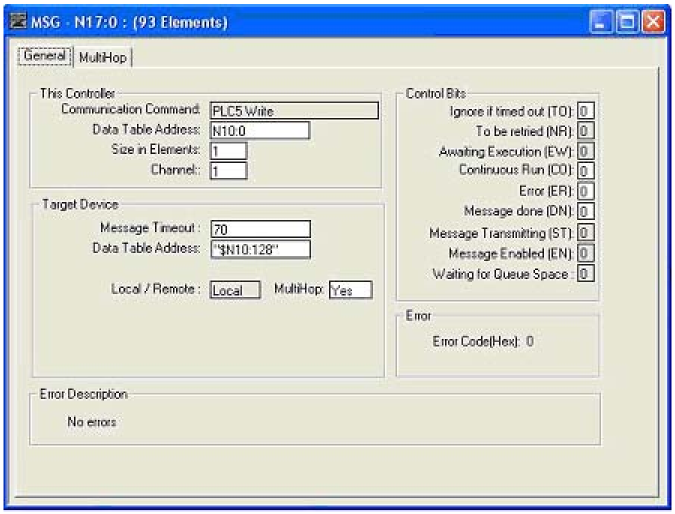
The following screen depicts an *PLC-5 Typed Write - Set Receive Produced Sequence Number* message in ladder logic.



MSG
Read/Write Message
Type Peer-To-Peer
Read/Write Write
Target Device PLC5
Local/Remote Local
Control Block N17:0
Control Block Length 93
Setup Screen

Make the following changes to the ladder logic.

1. Select the **Write** option.
2. Select the **PLC5** option.
3. Select **Local**.
4. Assign a dedicated integer file of 93 integers to the **Control** block.
5. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.



MSG - N17:0 : (93 Elements)

This Controller
Communication Command: PLC5 Write
Data Table Address: N10:0
Size in Elements: 1
Channel: 1

Target Device
Message Timeout : 70
Data Table Address: "$N10:128"
Local / Remote : Local MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0
Waiting for Queue Space : 0

Error
Error Code(Hex): 0

Error Description
No errors

6. Make the following changes.

   a. Specify the file address where the receive sequence number resides in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.

   b. Type **1** in the **Size of Elements** box.

   c. Set the **Channel** parameter to **1** to use the Ethernet port.

      The **Message Timeout** parameter is not actually configurable. The RSLogix 500 sets the value in this box based on the Ethernet timeout settings.

   d. Specify the port-specific receive sequence file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. The receive sequence file address must be specified in logical ASCII format. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.

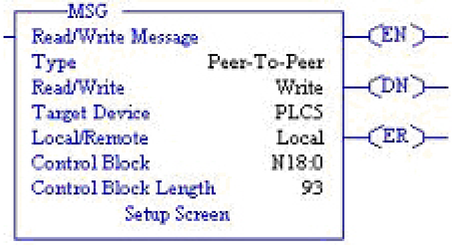
   e. Set the **MultiHop** option to **Yes**.

6.5.6.11. PLC-5 Typed Write - Set Transmit Produced Sequence Number Message - SLC PLC

The following screen depicts an *PLC-5 Typed Write - Set Transmit Produced Sequence Number* message in ladder logic.

Make the following changes to the ladder logic.

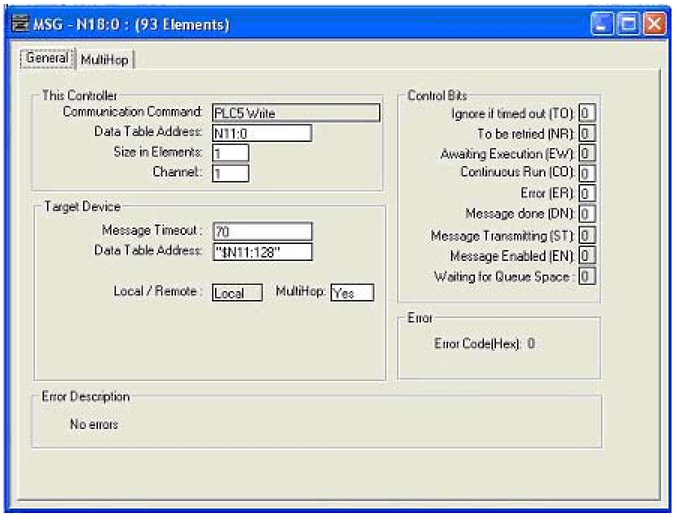
1. Select the **Write** option.
2. Select the **PLC5** option.
3. Select **Local**.
4. Assign a dedicated integer file of 93 integers to the **Control** block.



MSG
Read/Write Message
Type Peer-To-Peer
Read/Write Write
Target Device PLC5
Local/Remote Local
Control Block N18:0
Control Block Length 93
Setup Screen

5. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
6. Make the following changes.
   a. Specify the file address where the transmit sequence number resides in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
   b. Type **1** in the **Size of Elements** box.
   c. Set the **Channel** parameter to **1** to use the Ethernet port.

      The **Message Timeout** parameter is not actually configurable. The RSLogix 500 sets the value in this box based on the Ethernet timeout settings.
   d. Specify the port-specific transmit sequence number file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. The transmit sequence number file address must be specified in logical ASCII format. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
   e. Set the **MultiHop** option to **Yes**.



MSG - N18:0 : (93 Elements)

This Controller
Communication Command: PLC5 Write
Data Table Address: N11:0
Size in Elements: 1
Channel: 1

Target Device
Message Timeout : 70
Data Table Address: "$N11:128"
Local / Remote : Local MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0
Waiting for Queue Space : 0

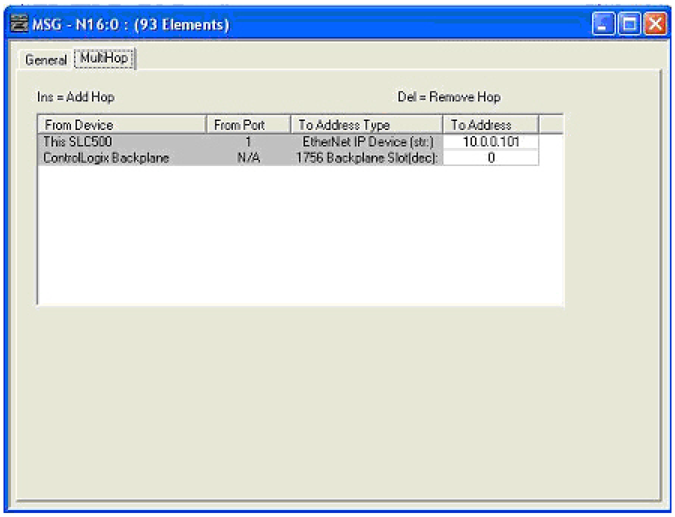
Error
Error Code(Hex): 0

Error Description
No errors

6.5.6.12. MultiHop Screen

1. Select the **MultiHop** tab on the **MSG** dialog.
2. Make the following changes.
   a. Type the IP address for the DeviceMaster (the designated EtherNet/IP Device) in the **To Address** box.
   b. Type **0** (zero) in the **To Address** box for the ControlLogix Backplane.

Note: *If you are using an SLC 5/03 or SLC 5/04 with an EtherNet/IP sidecar, you may need to add Hops to route the message on your PLC.*



MSG - N16:0 : (93 Elements)

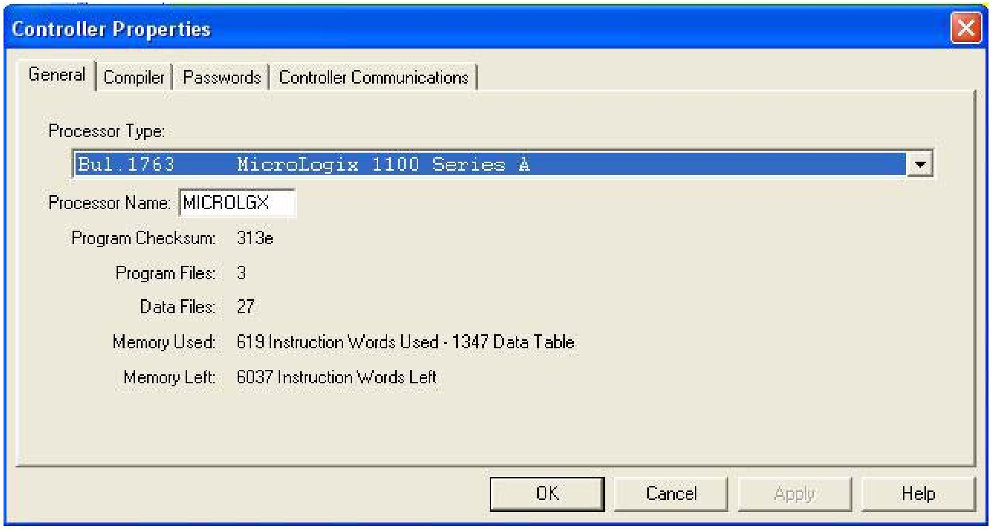
Ins = Add Hop Del = Remove Hop

From Device	From Port	To Address Type	To Address
This SLC500	1	EtherNet IP Device (str:)	10.0.0.101
ControlLogix Backplane	N/A	1756 Backplane Slot(dec):	0

6.5.7. Configuring and Running the MicroLogix RSLogix 500 Example Program

1. Select the appropriate message type example programs (_MICROLGX) and copy the files (**.RSS**) to the desired directory.
2. Start RSLogix 500 and open the **.RSS** file through RSLogix 500.
3. To modify the PLC program for your system, double-click **Controller Properties**.

In the **General** tab, select your MicroLogix processor type under **Processor Type** and provide a **Processor Name**.



Processor Type: Bul.1763 MicroLogix 1100 Series A

Processor Name: MICROLGX

Program Checksum: 313e

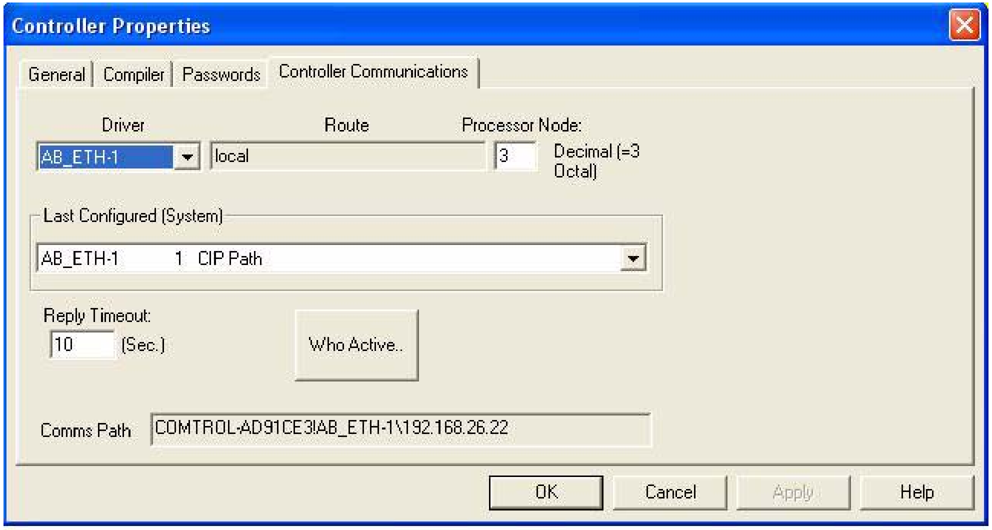
Program Files: 3

Data Files: 27

Memory Used: 619 Instruction Words Used - 1347 Data Table

Memory Left: 6037 Instruction Words Left

4. Select the **Controller Communications** tab and select the following options:
   a. Set **Driver** to the appropriate type to allow RSLogix 500 to communicate with the MicroLogix processor.
   b. Type the processor node number in the **Processor Node** box. (You may reference the **Last Configured (System)** node or select **Who Active**.).
5. Click **OK** to apply your changes and close the **Controller Properties** dialog.



Driver: AB_ETH-1

Route: local

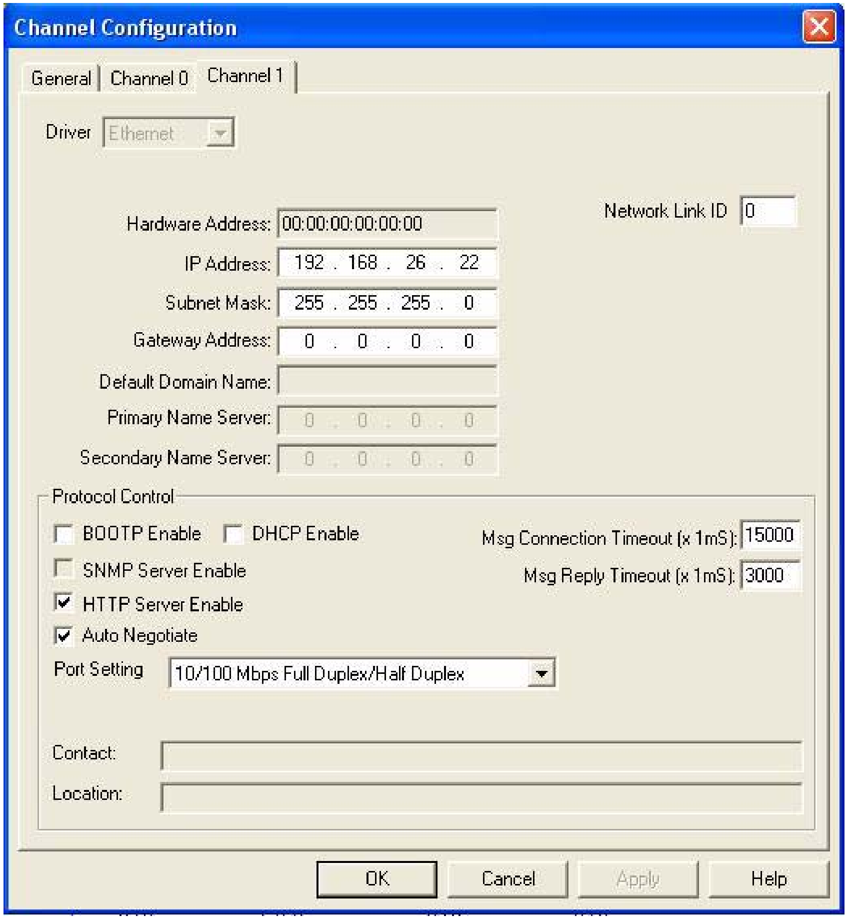
Processor Node: 3 Decimal (=3 Octal)

Last Configured (System): AB_ETH-1 1 CIP Path

Reply Timeout: 10 (Sec.)

Comms Path: COMTROL-AD91CE3!AB_ETH-1\192.168.26.22

6. Double-click **Processor Status**, select the **Chan. 1 System** tab on the **Channel Configuration** dialog and select the following options.
   a. Type the IP address for your PLC in the **IP Address** box if you are not using **Bootp**.
   b. Type the subnet mask for your PLC in the **Subnet Mask** box.
   c. Type the gateway address for your PLC in the **Gateway Address** box.
   d. Select **Bootp Enable** if you are using Bootp to initialize your network settings.
   e. Select **HTTP Server Enable.**
   f. Select **Auto Negotiate** if your network is capable of Ethernet auto-negotiation. If you select **Auto Negotiate**, set the **Port Setting** to **10/100 Mbps Full Duplex/Half Duplex**. If you do not select **Auto Negotiate**, select the speed and duplex for your network connection.
   g. Click **OK** to apply your changes and close the **Channel Configuration** dialog.
7. In the ladder logic, double-click **Setup Screen** in an **MSG** instruction.
8. Select the **MultiHop** tab and make the following changes.
9. On the first line, type the IP address for the DeviceMaster in the **To Address** box.
10. Close the **MSG** dialog.
11. Repeat steps 14 through 17 for each **MSG** instruction in the ladder logic.
12. Download the PLC program to your PLC and run the program.



Driver: Ethernet

Hardware Address: 00:00:00:00:00:00

Network Link ID: 0

IP Address: 192 . 168 . 26 . 22

Subnet Mask: 255 . 255 . 255 . 0

Gateway Address: 0 . 0 . 0 . 0

Default Domain Name:

Primary Name Server: 0 . 0 . 0 . 0

Secondary Name Server: 0 . 0 . 0 . 0

Protocol Control

□ BOOTP Enable □ DHCP Enable

□ SNMP Server Enable

☑ HTTP Server Enable

☑ Auto Negotiate

Port Setting: 10/100 Mbps Full Duplex/Half Duplex

Msg Connection Timeout (x 1mS): 15000

Msg Reply Timeout (x 1mS): 3000

Contact:

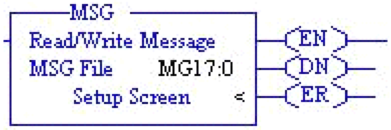
Location:

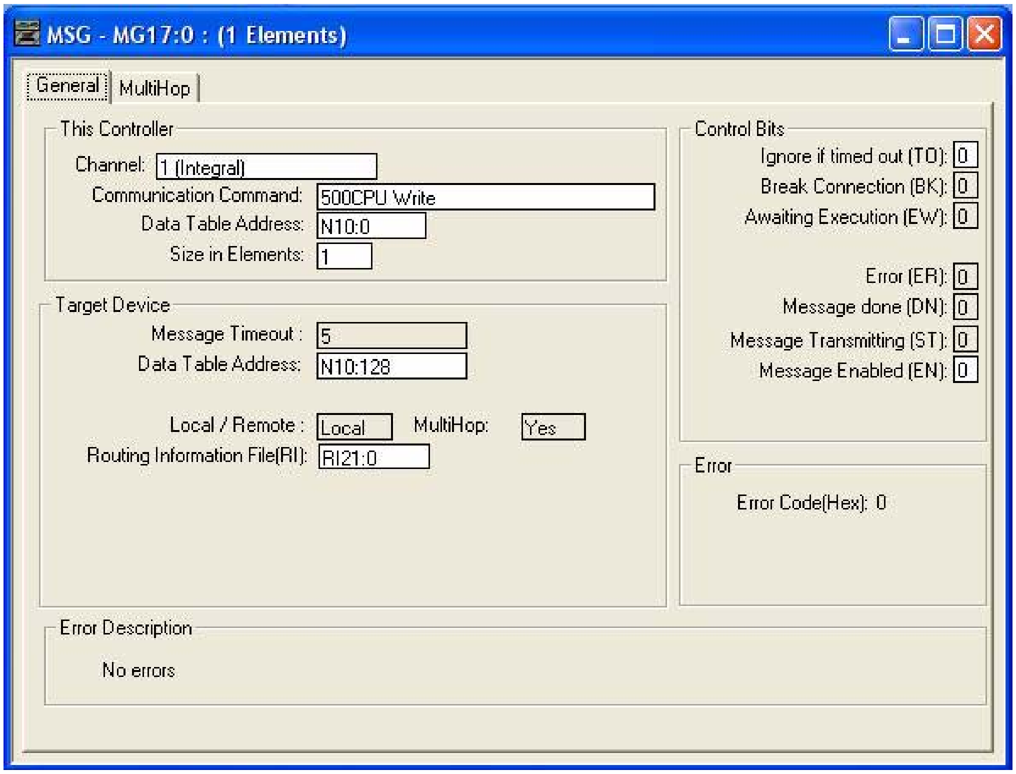
6.5.7.1. Receive Sequence Number Init Message

The following screen depicts a *SLC Typed Read - Receive Sequence Number Init* message in ladder logic. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.

Make the following changes:

1. Set the **Channel** parameter to **1** to use the Ethernet port.
2. Specify the file address where the transmit sequence number resides in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
3. Type **1** in the **Size of Elements** box.
4. Specify the port-specific transmit sequence number file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
5. Set the **Routing Information File (RI)** to that of your PLC program.
6. Select the **MultiHop** pane. On the first line, type the IP address for the DeviceMaster in the **To Address** box.





This Controller

Channel: 1 (Integral)
Communication Command: 500CPU Write
Data Table Address: N10:0
Size in Elements: 1

Target Device

Message Timeout : 5
Data Table Address: N10:128
Local / Remote : Local MultiHop: Yes
Routing Information File(RI): RI21:0

Control Bits

Ignore if timed out (TO): 0
Break Connection (BK): 0
Awaiting Execution (EW): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0

Error

Error Code(Hex): 0

Error Description

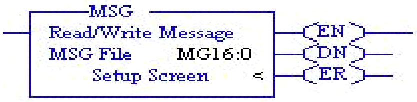
No errors

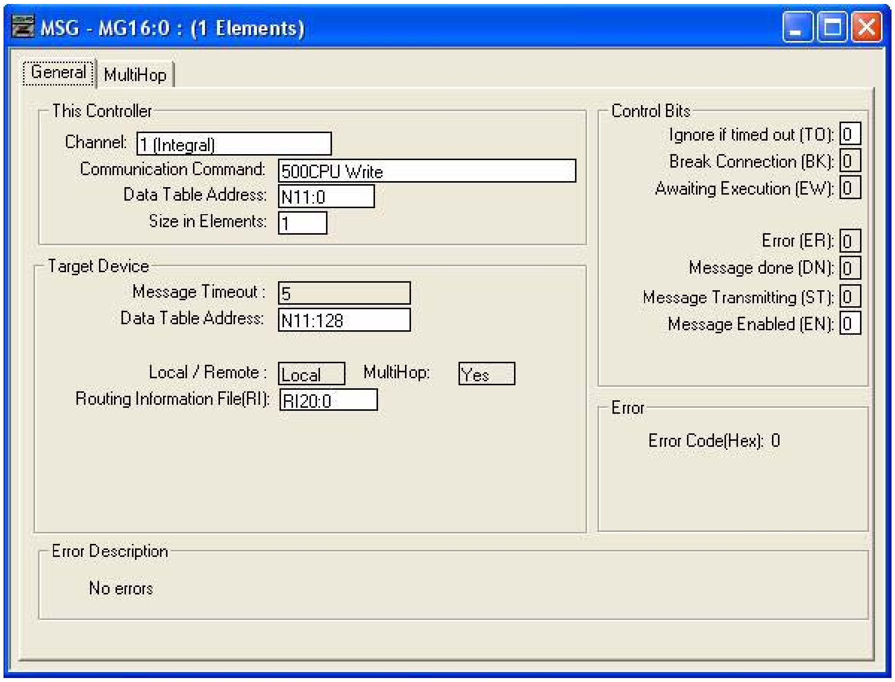
6.5.7.2. Transmit Sequence Number Init Message

The following screen depicts a *SLC Typed Read - Transmit Sequence Number Init* message in ladder logic. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.

Make the following changes:

1. Set the **Channel** parameter to **1** to use the Ethernet port.
2. Specify the file address where the transmit sequence number resides in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
3. Type **1** in the **Size of Elements** box.
4. Specify the port-specific transmit sequence number file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
5. Set the **Routing Information File (RI)** to that of your PLC program.
6. Select the **MultiHop** pane. On the first line, type the IP address for the DeviceMaster in the **To Address** box.




MSG - MG16:0 : (1 Elements)

This Controller

Channel:	1 (Integral)
Communication Command:	500CPU Write
Data Table Address:	N11:0
Size in Elements:	1

Target Device

Message Timeout :	5
Data Table Address:	N11:128
Local / Remote :	Local
MultiHop:	Yes
Routing Information File(RI):	RI20:0

Control Bits

Ignore if timed out (TO):	0
Break Connection (BK):	0
Awaiting Execution (EW):	0
Error (ER):	0
Message done (DN):	0
Message Transmitting (ST):	0
Message Enabled (EN):	0

Error

Error Code(Hex): 0

Error Description

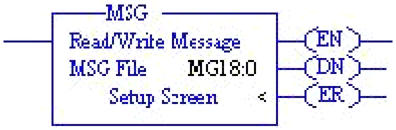
No errors

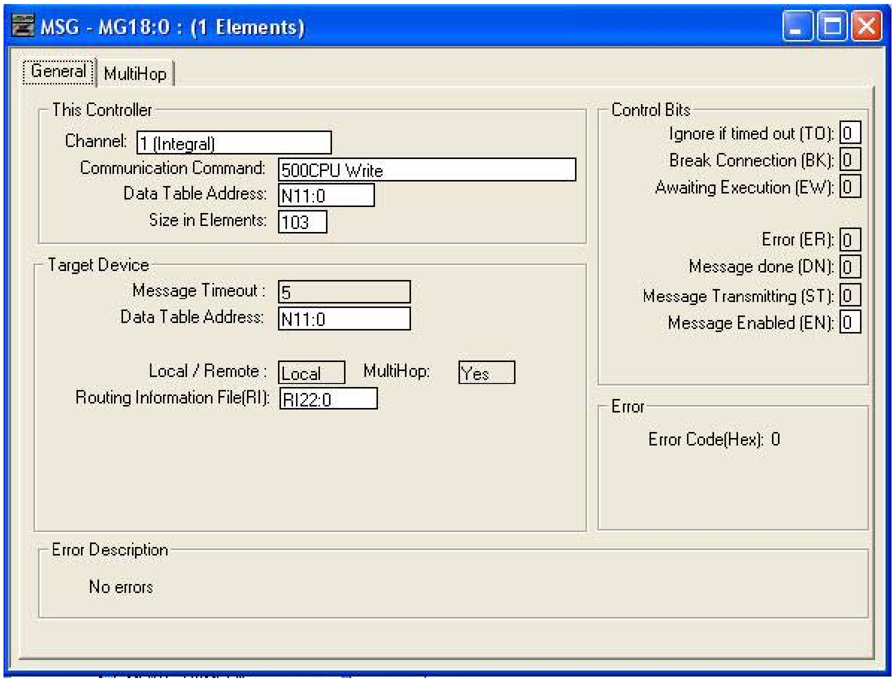
6.5.7.3. Transmit Data Message

The following screen depicts an *SLC Typed Write - Transmit Data* message in ladder logic. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.

Make the following changes:

1. Set the **Channel** parameter to **1** to use the Ethernet port.
2. Specify the file address where the transmit sequence number resides in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
3. Type **103** in the **Size of Elements** box to transmit receive the maximum data size for this type of message. This size must be large enough to include the sequence number (one integer), length (one integer), and enough integers to transmit all of your data.
4. Specify the port-specific transmit sequence number file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
5. Set the **Routing Information File (RI)** to that of your PLC program.
6. Select the **MultiHop** pane. On the first line, type the IP address for the DeviceMaster in the **To Address** box.




MSG - MG18:0 : (1 Elements)

This Controller

Channel: 1 (Integral)

Communication Command: 500CPU Write

Data Table Address: N11:0

Size in Elements: 103

Target Device

Message Timeout : 5

Data Table Address: N11:0

Local / Remote : Local MultiHop: Yes

Routing Information File(RI): RI22:0

Control Bits

Ignore if timed out (TO): 0

Break Connection (BK): 0

Awaiting Execution (EW): 0

Error (ER): 0

Message done (DN): 0

Message Transmitting (ST): 0

Message Enabled (EN): 0

Error

Error Code(Hex): 0

Error Description

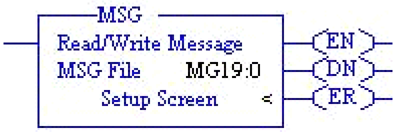
No errors

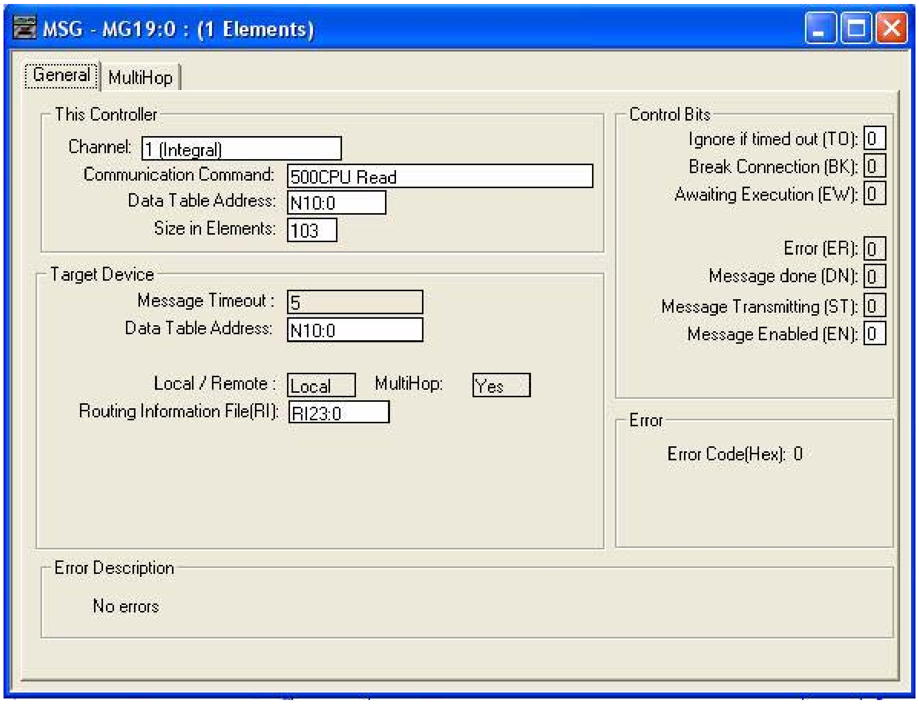
6.5.7.4. Receive Data Message

The following screen depicts an *SLC Typed Read - Receive Data* message in ladder logic. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.

Make the following changes:

1. Set the **Channel** parameter to **1** to use the Ethernet port.
2. Specify the file address where the transmit sequence number resides in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
3. Type **103** in the **Size of Elements** box to receive the maximum sized data size for this type of message. This size must be large enough to include the sequence number (one integer), length (one integer), and enough integers to receive all of your data.
4. Specify the port-specific transmit sequence number file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
5. Set the **Routing Information File (RI)** to that of your PLC program.
6. Select the **MultiHop** pane. On the first line, type the IP address for the DeviceMaster in the **To Address** box.





This Controller

Channel: 1 (Integral)

Communication Command: 500CPU Read

Data Table Address: N10:0

Size in Elements: 103

Target Device

Message Timeout : 5

Data Table Address: N10:0

Local / Remote : Local MultiHop: Yes

Routing Information File(RI): RI23:0

Control Bits

Ignore if timed out (TO): 0

Break Connection (BK): 0

Awaiting Execution (EW): 0

Error (ER): 0

Message done (DN): 0

Message Transmitting (ST): 0

Message Enabled (EN): 0

Error

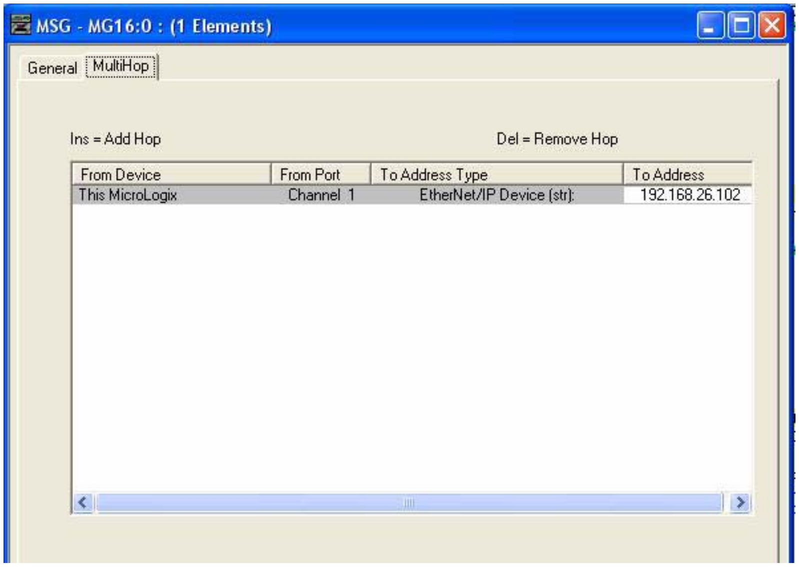
Error Code(Hex): 0

Error Description

No errors

6.5.7.5. MultiHop Screen

Select the **MultiHop** tab on the **MSG** dialog. 2. Make the following changes. a. Type the IP address for the DeviceMaster (the designated EtherNet/IP Device) in the **To Address** box.



From Device	From Port	To Address Type	To Address
This MicroLogix	Channel 1	EtherNet/IP Device (str):	192.168.26.102

6.6. PLC-5 PLC Programming Example Instructions

This topic describes how to use RSLogix 5 to configure and run the DeviceMaster in a PLC-5 PLC environment.

You can configure the RSLogix 5 PLC program examples to your site's requirements. This program is included in the self-installing file (**.MSI**) and is copied to the **Comtrol/EtherNetIP** directory on your computer when you open the **.MSI** file and follow the prompts. The self-installing file includes the following RSLogix 5 PLC program examples:

- *6.6.4. lpbkExampleSlcMsgPollRS5*
- *6.6.5. lpbkExamplePlc5MsgPollRS5*

These program examples are intended to aid the PLC programmer. These program examples were developed with:

- RSLogix 5 (version 6.00.00)
- Enhanced PLC-5/20 (series E with revision J firmware)
- Ethernet sidecar (version Enet/B)

Note: *The PLC program examples are designed to interface with a* DeviceMaster *1-port or on Port 1 of a 2-port or 4-port. Additional programming is required to use all ports on a 2-port or 4-port.*

Caution

Disclaimer: **Comtrol supplies example PLC programs for demonstration purposes only. They are intended for the sole purpose of an example loop-back demonstration in a controlled lab environment. They are not intended for use in a production environment and may not function correctly on all PLCs. Comtrol does not warrant these example programs or any part thereof. The user assumes all liability for any modification to and use of a modified example program.**

6.6.1. What is RSLogix 5?

RSLogix 5 is a Windows ladder logic programming package for the PLC-5 PLCs.

Note: *See the RSLogix 5 Help for more information on this product.*

6.6.2. Requirements

- The EtherNet/IP firmware must be installed on the DeviceMaster and configured as described in the *EtherNet/IP Hardware Installation and Configuration Guide*.
- The DeviceMaster must be installed on the same Ethernet network segment as the PLC.
- RSLogix 5 must be installed on your computer. Note that the instructions in this guide require that you have some familiarity with this programming application.
- A loopback plug is required for the first port on the DeviceMaster when running an example PLC program. See the *EtherNet/IP Hardware Installation and Configuration Guide* for information on loopback plugs.
- The PLC program examples (**.PC5**, **.SY5** and **.SY6** files) are optional. You can copy the PLC program examples from the CD or download the latest program examples from the Internet. See 1.4.1. Locating the Latest Software and Documents on Page 8, for the location of the PLC program examples.

6.6.3. Example Program Considerations

- While the RSLogix example programs are simple in nature, they include error counters and transmit retry mechanisms for timed-out messages. You may or may not want to include the error counters and transmit retry mechanisms in your own application.
- The receive and transmit sequence numbers are cleared on the DeviceMaster when you start the programs. However, the sequence numbers must be in sync between the PLC and DeviceMaster for the programs to operate correctly.
- Statistics retrieval is not included in the example programs, but you can easily add it by inserting a request statistics message.
- The socket ports can be accessed the same way as the serial ports and return the data in the same manner. To access a socket port, just change the associated DeviceMaster file addresses.

6.6.4. lpbkExampleSlcMsgPollRS5

This example program demonstrates an RSLogix 5 loopback PLC program using the SLC Typed messages in the *Polling receive* method. This program initializes receive and transmit produced data sequence numbers at startup on the DeviceMaster and then loops data through a loopback plug on the serial port. The SLC Typed Write data messages transmit the data and the SLC Typed Read Data messages receive the data and the sequence numbers are incremented.

This example program includes the following files:

- **lpbkExampleSlcMsgPollRS5.PC5** - Ladder logic in ASCII format.
- **lpbkExampleSlcMsgPollRS5.SY5** - Symbol definitions for RSLogix 5 Version 5.xx.xx.
- **lpbkExampleSlcMsgPollRS5.SY6** - Symbol definitions for RSLogix 5 Version 6.xx.xx.

6.6.5. lpbkExamplePlc5MsgPollRS5

This example program demonstrates an RSLogix 5 loopback PLC program using the PLC-5 Typed messages in the *Polling receive* method. This program initializes receive and transmit produced data sequence numbers at startup on the DeviceMaster and then loops data through a loopback plug on the serial port. The PLC-5 Typed Write data messages transmit the data and the PLC-5 Typed Read Data messages receive the data and the sequence numbers are incremented.

This example program includes the following files:

- **lpbkExamplePlc5MsgPollRS5.PC5** - Ladder logic in ASCII format.
- **lpbkExamplePlc5MsgPollRS5.SY5** - Symbol definitions for RSLogix 5 Version 5.xx.xx.
- **lpbkExamplePlc5MsgPollRS5.SY6** - Symbol definitions for RSLogix 5 Version 6.xx.xx.

6.6.6. lpbkExamplePlc5MsgFileRS500

This example program demonstrates a loop-back RSLogix 5 PLC program using PLC-5 Typed messages in the *Write-to-File receive* method. This program initializes the produced receive and transmit data sequence numbers at startup and then loops data via a loop-back plug on the serial port. The data is transmitted via PLC-5 Typed Write data messages and received automatically via a write to file command from the DeviceMaster. The sequence numbers are incremented with each message.

The following files apply:

- **lpbkExamplePlc5MsgFileRS5.PC5** – ladder logic in ASCII form.
- **lpbkExamplePlc5MsgFileRS5.SY5** – symbol definitions for RSLogix 5 Version 5.xx.xx.
- **lpbkExamplePlc5MsgFileRS5.SY6** – symbol definitions for RSLogix 5 Version 6.xx.xx.

6.6.7. lpbkExamplePlc5MsgFileSyncRS5

This example program demonstrates a loop-back RSLogix 5 PLC program using PLC-5 Typed messages in the *Write-to-File-Synced receive* method. This program initializes the produced receive and transmit sequence numbers as well as the consumed receive sequence number at startup and then loops data via a loop-back plug on the serial port. The data is transmitted via PLC-5 Typed Write data messages and received automatically via a write to file command from the DeviceMaster. The consumed receive sequence number is updated to match the produced receive sequence number and sent to the DeviceMaster to complete the synchronization process. All sequence numbers are incremented with each message.

The following files apply:

- **lpbkExamplePlc5MsgFileSyncRS5.PC5** – ladder logic in ASCII form.
- **lpbkExamplePlc5MsgFileSyncRS5.SY5** – symbol definitions for RSLogix 5 Version 5.xx.xx.
- **lpbkExamplePlc5MsgFileSyncRS5.SY6** – symbol definitions for RSLogix 5 Version 6.xx.xx.

6.6.8. Configure the DeviceMaster for the RSLogix 5 Program

The following procedure configures the DeviceMaster for PLC-5 and SLC PLCs. You must perform this task before you configure and run the example RSLogix 5 program. For more information on the embedded web pages, see *Chapter 4. Embedded Configuration Pages* on Page 85.

1. Attach a loopback plug to the serial port.
2. Access the *Serial Settings* web page. Open PortVision DX, right-click the DeviceMaster and click **Webpage** or open a browser and type the IP address for the DeviceMaster in the **Address** box.
3. Click **Serial |Port *n***. Where ***n*** is the port number.
4. Set the serial port settings under **Serial Configuration** to the following values

Field	Setting
Mode	RS-232
Baud	57600
Parity	none
Data Bits	8
Stop Bits	1
Flow Control	none
DTR	off
Rx Timeout Between Packets	200

5. Set the serial port settings under **Serial Packet Identification** to the following values.

Field	Settings
STX RX Detect	Set to **one byte** and **Byte 1** to **2**.
ETX Rx Detect	Set to **one byte** and **Byte 1** to **3**.
STX Tx Append	Set to **one byte** and **Byte 1** to **2**.
ETX Tx Append	Set to **one byte** and **Byte 1** to **3**.
Strip Rx STX/ETX	Select.
Discard Rx Packets With Errors	Select.
(PLC-5/SLC) Rx MS Byte First	Optionally, select.
(PLC-5/SLC) Tx MS Byte First	Optionally, select.

6. Click the **Save** button.

7. Click **EthernNet/IP Settings** and set the serial port settings to the following values.

Field	Settings
TX Sequence Number Checking.	Select
Rx (To PLC) Ethernet Transfer Method	• Set to Polling for lpbkExampleSlcMsgPollRS5 and lpbkExamplePlc5MsgPollRS5. • Set to Write-to-Tag/File for lpbkExamplePlc5MsgFileRS5. • Set to Write-to-Tag/File-Synced for lpbkExamplePlc5MsgFileSyncRS5.
PLC IP Address	• Leave blank for Polling. • Set to IP Address of PLC for Write-to-Tag/File and Write-to-Tag/File-Synced.
PLC Controller Slot Number	Unused and can remain blank.
Rx (To PLC) Produced Data Tag/File Name	• Leave blank for Polling. • Set to $N10:0, the PLC receive filename for Write-to-File and Write-to-File-Synced.

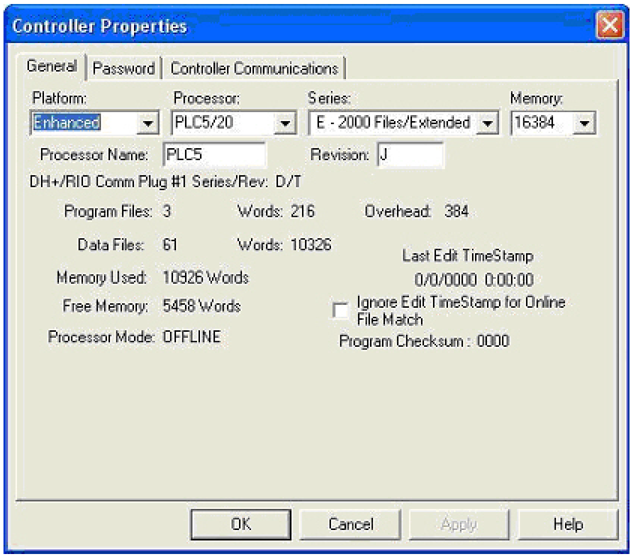
8. Click **Save**.

6.6.9. Configure and Run the Example RSLogix 5 Program

You can configure and run the RSLogix 5 example programs through RSLogix 5. For additional information on the RSLogix 5, see *6.6.10. RSLogix 5 Screen Examples* on Page 199.

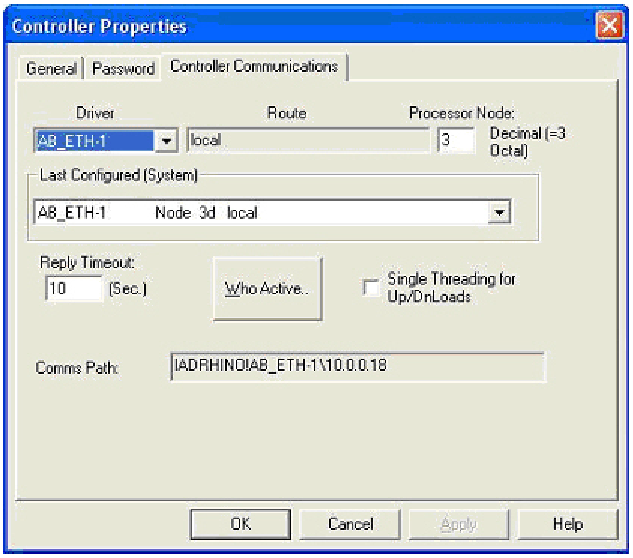
Note: *The DeviceMaster must be configured for PLC-5/SLC before you can configure and download the example RSLogix 5 program. For instructions on configuring DeviceMaster, see 6.6.8. Configure the DeviceMaster for the RSLogix 5 Program, earlier in this section.*

1. Select the appropriate message type example programs (SLC or PLC-5) and copy the files (**.PC5**, **.SY5** and **.SY6**) to the desired directory.
2. Start RSLogix 5 and open the **.PC5** file.
3. To modify the PLC program for your system, double-click **Controller Properties**.
4. In the **General** tab, select your PLC-5 processor type under **Processor Type** and provide a **Processor Name**.
5. Select your PLC-5 series in the **Series** box and select your firmware revision in the **Revision** box.



Platform: Enhanced
Processor: PLC5/20
Series: E - 2000 Files/Extended
Memory: 16384
Processor Name: PLC5
Revision: J
DH+/RIO Comm Plug #1 Series/Rev: D/T
Program Files: 3 Words: 216 Overhead: 384
Data Files: 61 Words: 10326
Memory Used: 10926 Words
Free Memory: 5458 Words
Processor Mode: OFFLINE
Last Edit TimeStamp 0/0/0000 0:00:00
□ Ignore Edit TimeStamp for Online File Match
Program Checksum : 0000

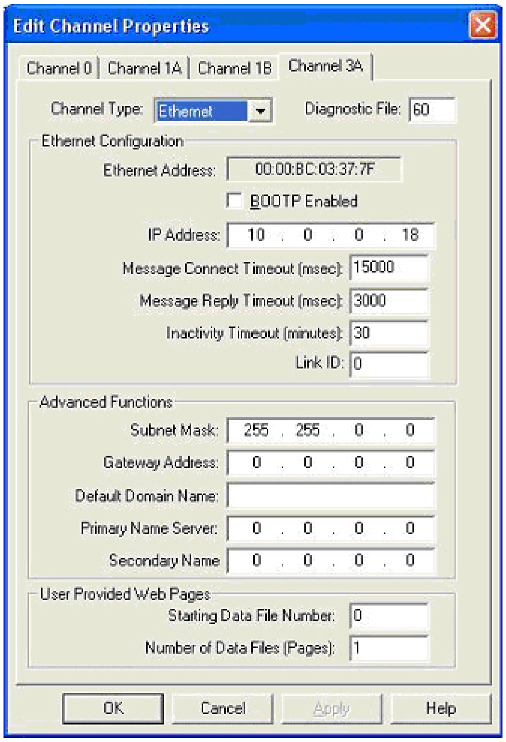
6. Select the **Controller Communications** tab and select the following options.
   a. Set **Driver** to the appropriate type to allow RSLogix 5 to communicate with the PLC-5 processor.
   b. Type the processor node number in the **Processor Node** box. (You may reference the **Last Configured (System)** node or select **Who Active**.)
7. Click **OK** to apply your changes and close the **Controller Properties** dialog.
8. Double-click **I/O Configuration** and verify your chassis and PLC-5 type. If the chassis type is not correct:
   a. Right-click the chassis type (for example, **1771-A1B (4 Slots)**) and select **Properties**.
   b. Select your chassis.
   c. Optionally, select the appropriate DIP switch settings for your system from the **DIP Switches** panel.
   d. Click **OK** to save settings.
9. Click **OK** to save settings.



Driver: AB_ETH-1
Route: local
Processor Node: 3 Decimal (=3 Octal)
Last Configured (System): AB_ETH-1 Node 3d local
Reply Timeout: 10 (Sec.)
□ Single Threading for Up/DnLoads
Comms Path: IADRHINO!AB_ETH-1\10.0.0.18

10. Double-click **Channel Configuration** and click the **Channel 3A** tab.

    Note: *The port number for the Ethernet channel may be different on your PLC-5.*



Edit Channel Properties

Channel 0 | Channel 1A | Channel 1B | Channel 3A

Channel Type: Ethernet

Diagnostic File: 60

Ethernet Configuration

Ethernet Address: 00:00:BC:03:37:7F

□ BOOTP Enabled

IP Address: 10 . 0 . 0 . 18

Message Connect Timeout (msec): 15000

Message Reply Timeout (msec): 3000

Inactivity Timeout (minutes): 30

Link ID: 0

Advanced Functions

Subnet Mask: 255 . 255 . 0 . 0

Gateway Address: 0 . 0 . 0 . 0

Default Domain Name:

Primary Name Server: 0 . 0 . 0 . 0

Secondary Name 0 . 0 . 0 . 0

User Provided Web Pages

Starting Data File Number: 0

Number of Data Files (Pages): 1

11. In the **Ethernet Configuration** panel, make the following changes:
    a. Set **Channel Type** to **Ethernet**.
    b. Type **60** in the **Diagnostic File** box. You can use the diagnostic file to help solve any network-related problems.
    c. Type the IP address for your PLC-5 in the **IP address** box.
    d. Type the subnet mask for your PLC-5 in the **Subnet Mask** box.
    e. Type the gateway address for your PLCI-5 in the **Gateway Address** box.
    f. If applicable to your network, type the addresses for the **Primary Name Server** and **Secondary Name Server**.
12. Click **OK** to apply your changes and close the **Channel Configuration** dialog.
13. In the ladder logic, double-click **Setup Screen** in an **MSG** instruction.
14. Select the **MultiHop** tab and type the IP address for the DeviceMaster in the **To Address** box.
15. Close the **MSG** dialog.
16. Repeat steps 13 through 15 for each MSG instruction in the ladder logic.
17. Download the PLC program to your PLC and run the program.

6.6.10.RSLogix 5 Screen Examples

The following subsections explain how to configure the DeviceMaster through RSLogix 5.

Use these screens to set up PLC and program the various messages.

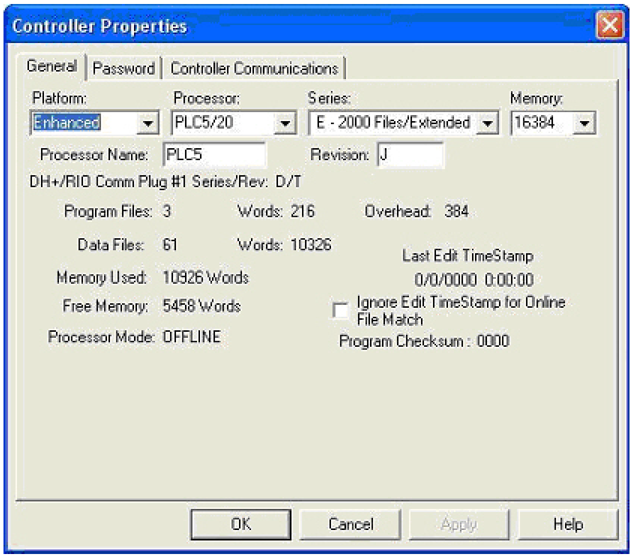
6.6.10.1. Requirements

- PLC-5 PLCs require EtherNet/IP firmware 2.01 or later running on the DeviceMaster.
- The PLC-5 PLC firmware must support MultiHop, ControlLogix devices and EtherNet/IP. The tables in *3.3.1. Requirements* on Page 77 list PLCs that support EtherNet/IP and the required firmware version for each PLC.
- The PLC program examples (**.SLC**, **.SY5** and **.SY6** files) are required. You can copy the PLC program examples from the CD or download the latest program examples from the Internet. See 1.4.1. Locating the Latest Software and Documents on Page 8 for the location of the PLC program examples.
- You must set up the Processor and Ethernet communication port properly for EtherNet/IP to function. Read and follow the instructions in the appropriate Rockwell product documents.
  - Enhanced and Ethernet PLC-5 Programmable Control, Publication 1785-6.5.12
  - ControlNet PLC-5 Programmable Controllers User Manual, Publication 1785-UM022B-EN-P
  - PLC-5 Ethernet Interface Module, Publication 1785-ENET

6.6.10.2. Setting up Processor and Ethernet Channel

The following screens show the recommended settings that allow EtherNet/IP to function properly on a PLC-5 PLC.

1. Start **RSLogix 5**.
2. Double-click **Controller Properties** and select the correct processor type and revision from the **General** tab on the **Controller Properties** dialog.



Platform: Enhanced

Processor: PLC5/20

Series: E - 2000 Files/Extended

Memory: 16384

Processor Name: PLC5

Revision: J

DH+/RIO Comm Plug #1 Series/Rev: D/T

Program Files: 3 Words: 216 Overhead: 384

Data Files: 61 Words: 10326

Last Edit TimeStamp

0/0/0000 0:00:00

Memory Used: 10926 Words

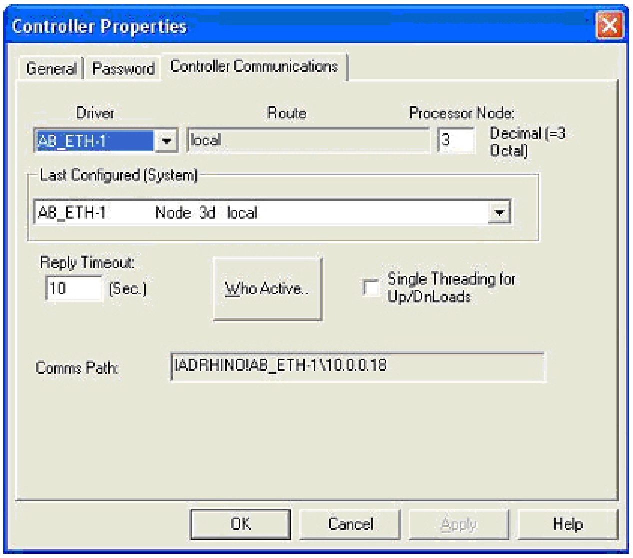
Free Memory: 5458 Words

□ Ignore Edit TimeStamp for Online File Match

Processor Mode: OFFLINE

Program Checksum : 0000

3. Select the **Controller Communications** tab and select the proper driver for RSLogix 5.
4. Click **OK** to apply your changes and close the **Controller Properties** dialog.



Driver: AB_ETH-1
Route: local
Processor Node: 3 Decimal (=3 Octal)

Last Configured (System): AB_ETH-1 Node 3d local

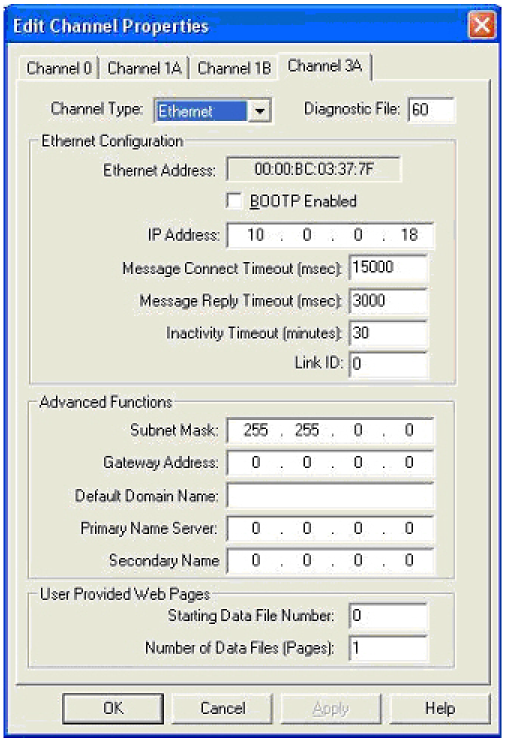
Reply Timeout: 10 (Sec.)

□ Single Threading for Up/DnLoads

Comms Path: IADRHINO!AB_ETH-1\10.0.0.18

5. Double-click **Channel Configuration**, click the **Channel 3A** tab and make the following changes.
   a. Type a value (between 0 and 256) in the **Diagnostic File** box for an integer diagnostic file. (This example uses Diagnostic file 60.) You can use the diagnostic file to help solve any network-related problems.
   b. Type the IP address in the **IP address** box.
   c. Type the subnet mask in the **Subnet Mask** box.
   d. Type the gateway address in the **Gateway Address** box.

   Note: *The Ethernet channel may be different on your PLC-5.*
6. Click **OK** to apply your changes and close the **Edit Channel Properties** dialog.



Channel Type: Ethernet
Diagnostic File: 60

Ethernet Configuration

Ethernet Address: 00:00:BC:03:37:7F

□ BOOTP Enabled

IP Address: 10 . 0 . 0 . 18

Message Connect Timeout (msec): 15000

Message Reply Timeout (msec): 3000

Inactivity Timeout (minutes): 30

Link ID: 0

Advanced Functions

Subnet Mask: 255 . 255 . 0 . 0

Gateway Address: 0 . 0 . 0 . 0

Default Domain Name:

Primary Name Server: 0 . 0 . 0 . 0

Secondary Name 0 . 0 . 0 . 0

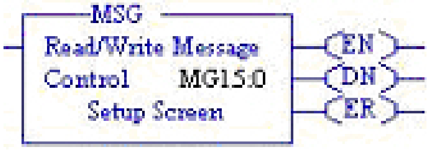
User Provided Web Pages

Starting Data File Number: 0

Number of Data Files (Pages): 1

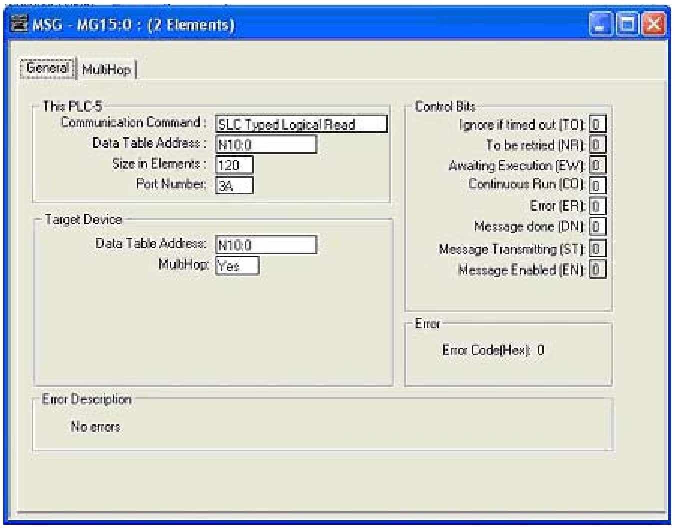
6.6.10.3. SLC Typed Read - Receive Data Message

The following screen depicts an *SLC Typed Read - Receive Data* message in ladder logic.



1. Assign a dedicated message file to the **Control** block in the ladder logic.
2. In the ladder logic, double-click the **Setup Screen** in the MSG instruction.
3. Make the following changes:
   a. Specify the file address to receive data information in the **Data Table Address** box on **This PLC-5** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
   b. Specify a size in the **Size of Elements** box that is large enough to receive the entire data message including the sequence number and length fields.
   c. Set the **Port Number** parameter to **3A** to use the Ethernet port.

      The port number for the Ethernet channel may be different on your PLC-5.
   d. Specify the port-specific read file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
   e. Set the **MultiHop** option to **Yes**.



MSG - MG15:0 : (2 Elements)

This PLC-5
Communication Command : SLC Typed Logical Read
Data Table Address : N10:0
Size in Elements : 120
Port Number: 3A

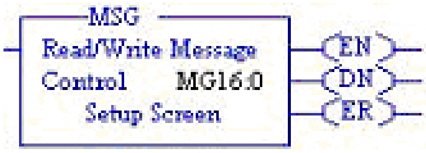
Target Device
Data Table Address: N10:0
MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0

Error
Error Code(Hex): 0

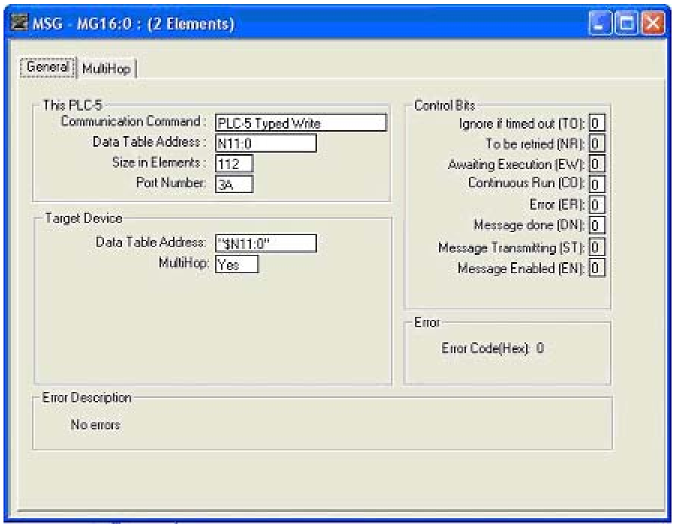
Error Description
No errors

6.6.10.4. SLC Typed Write - Transmit Data Message



The following screen depicts an *SLC Typed Write - Transmit Data* message in ladder logic.

1. Assign a dedicated message file to the **Control** block in the ladder logic (as shown above).
2. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.



This PLC-5
Communication Command : PLC-5 Typed Write
Data Table Address : N11:0
Size in Elements : 112
Port Number: 3A

Target Device
Data Table Address: "$N11:0"
MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0

Error
Error Code(Hex): 0

Error Description
No errors

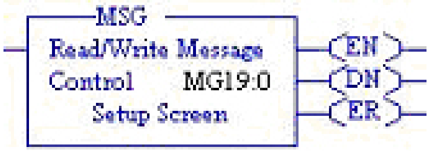
3. Make the following changes:
   a. Specify the file address to transmit data information in the **Data Table Address** box on **This PLC-5** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
   b. Specify a size in the **Size of Elements** box that is large enough to send the entire data message including the sequence number and length fields.
   c. Set the **Port Number** parameter to **3A** to use the Ethernet port.

      The port number for the Ethernet channel may be different on your PLC-5.
   d. Specify the port-specific transmit file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
   e. Set the **MultiHop** option to **Yes**.

6.6.10.5. SLC Typed Read - Retrieve Statistics Message

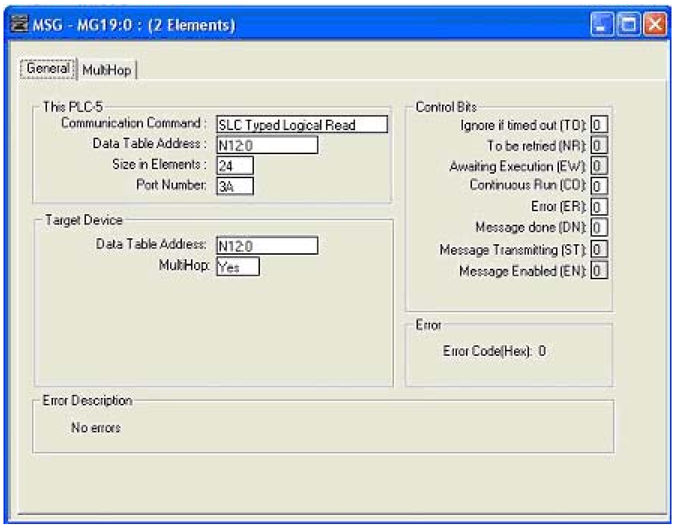
The following screen depicts an *SLC Typed Read - Retrieve Statistics* message in ladder logic.

1. Assign a dedicated message file to the **Control** block in the ladder logic.



2. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
3. Make the following changes:
    a. Specify the file address to receive statistics data information in the **Data Table Address** box on **This PLC-5** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
    b. Type **24** (twelve 32-bit integers) in the **Size of Elements** box.
    c. Set the **Port Number** parameter to **3A** to use the Ethernet port.

       The port number for the Ethernet channel may be different on your PLC-5.
    d. Specify the port-specific statistics file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
    e. Set the **MultiHop** option to **Yes**.



MSG - MG19:0 : (2 Elements)

This PLC-5
Communication Command : SLC Typed Logical Read
Data Table Address : N12:0
Size in Elements : 24
Port Number: 3A

Target Device
Data Table Address: N12:0
MultiHop: Yes

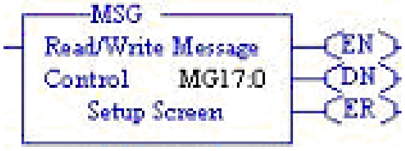
Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0

Error
Error Code(Hex): 0

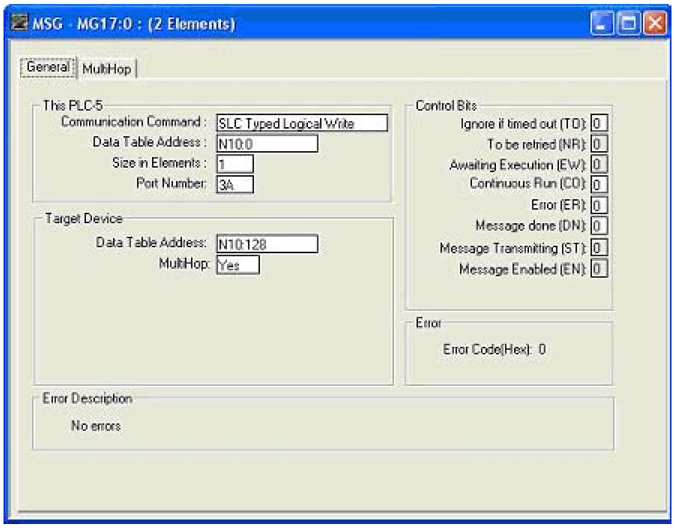
Error Description
No errors

6.6.10.6. SLC Typed Write - Set Receive Produced Sequence Number Message

The following screen depicts an *SLC Typed Write - Set Receive Produced Sequence Number* message in ladder logic.



1. Assign a dedicated message file to the **Control** block in the ladder logic (as shown above).
2. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.



MSG - MG17:0 : (2 Elements)

This PLC-5
Communication Command : SLC Typed Logical Write
Data Table Address : N10:0
Size in Elements : 1
Port Number: 3A

Target Device
Data Table Address: N10:128
MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0

Error
Error Code(Hex): 0

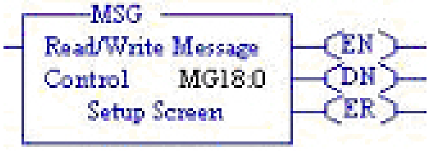
Error Description
No errors

3. Make the following changes:
   a. Specify the file address where the receive sequence number resides in the **Data Table Address** box on **This PLC-5** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
   b. Type **1** in the **Size of Elements** box.
   c. Set the **Port Number** parameter to **3A** to use the Ethernet port.

      Note: *The port number for the Ethernet channel may be different on your PLC-5.*
   d. Specify the port-specific receive sequence file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
   e. Set the **MultiHop** option to **Yes**.

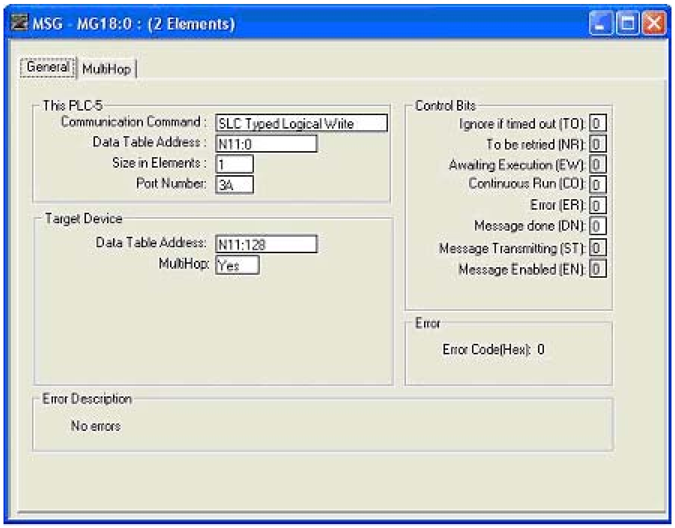
6.6.10.7. SLC Typed Write - Set Transmit Produced Sequence Number Message

The following screen depicts an *SLC Typed Write - Set Transmit Produced Sequence Number* message in ladder logic.



1. Assign a dedicated message file to the **Control** block in the ladder logic (as shown above).

2. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.



MSG - MG18:0 : (2 Elements)

This PLC-5
Communication Command : SLC Typed Logical Write
Data Table Address : N11:0
Size in Elements : 1
Port Number: 3A

Target Device
Data Table Address: N11:128
MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0

Error
Error Code(Hex): 0

Error Description
No errors

3. Make the following changes:

   a. Specify the file address where the file address where the transmit sequence number resides in the **Data Table Address** box on **This PLC-5** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.

   b. Type **1** in the **Size of Elements** box.

   c. Set the **Port Number** parameter to **3A** to use the Ethernet port.

      Note: *The port number for the Ethernet channel may be different on your PLC-5.*

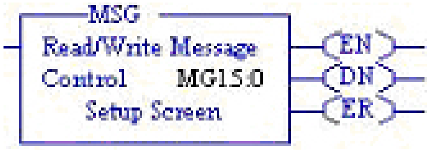
   d. Specify the port-specific transmit sequence number file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.

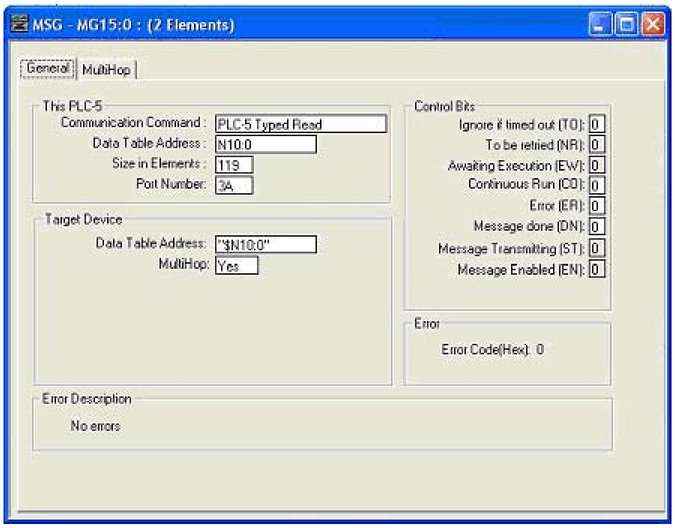
   e. Set the **MultiHop** option to **Yes**.

6.6.10.8. PLC-5 Typed Read - Receive Data Message

The following screen depicts a *PLC-5 Typed Read - Receive Data* message in ladder logic.

1. Assign a dedicated message file to the **Control** block in the ladder logic (as shown above).





MSG - MG15:0 : (2 Elements)

This PLC-5
Communication Command : PLC-5 Typed Read
Data Table Address : N10:0
Size in Elements : 119
Port Number: 3A

Target Device
Data Table Address: "$N10:0"
MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0

Error
Error Code(Hex): 0

Error Description
No errors

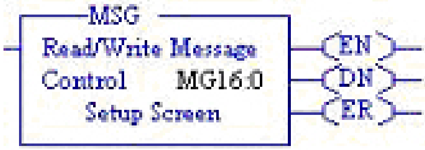
2. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
3. Make the following changes:
   a. Specify the file address to receive data information in the **Data Table Address** box on **This PLC-5** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
   b. Specify a size in the **Size of Elements** box that is large enough to receive the entire data message including the sequence number and length fields.
   c. Set the **Port Number** parameter to **3A** to use the Ethernet port.

      Note: *The port number for the Ethernet channel may be different on your PLC-5.*
   d. Specify the port-specific receive file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. The receive file address must be specified in logical ASCII format. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.
   e. Set the **MultiHop** option to **Yes**.

6.6.10.9. PLC-5 Typed Write - Transmit Data Message

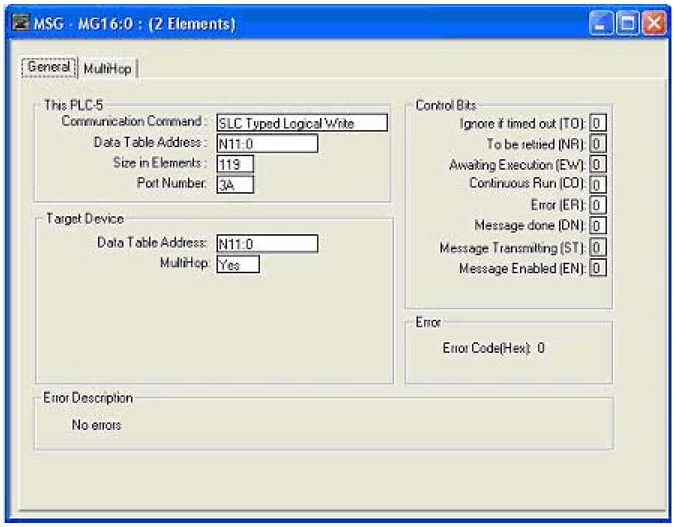
The following screen depicts a *PLC-5 Typed Write - Transmit Data* message in ladder logic.

1. Assign a dedicated message file to the **Control** block in the ladder logic (as shown above).



2. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
3. Make the following changes:
   a. Specify the file address to transmit data information in the **Data Table Address** box on **This PLC-5** panel. For more information on file addresses, see [3.3.3. DeviceMaster File Addressing]() on Page 78.
   b. Specify a size in the **Size of Elements** box that is large enough to send the entire data message including the sequence number and length fields.
   c. Set the **Port Number** parameter to **3A** to use the Ethernet port.

      Note: *The port number for the Ethernet channel may be different on your PLC-5.*



This PLC-5
Communication Command: SLC Typed Logical Write
Data Table Address: N11:0
Size in Elements: 119
Port Number: 3A

Target Device
Data Table Address: N11:0
MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0

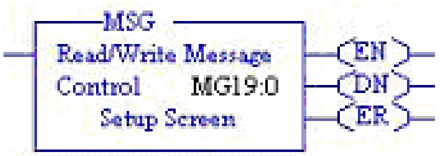
Error
Error Code(Hex): 0

Error Description
No errors

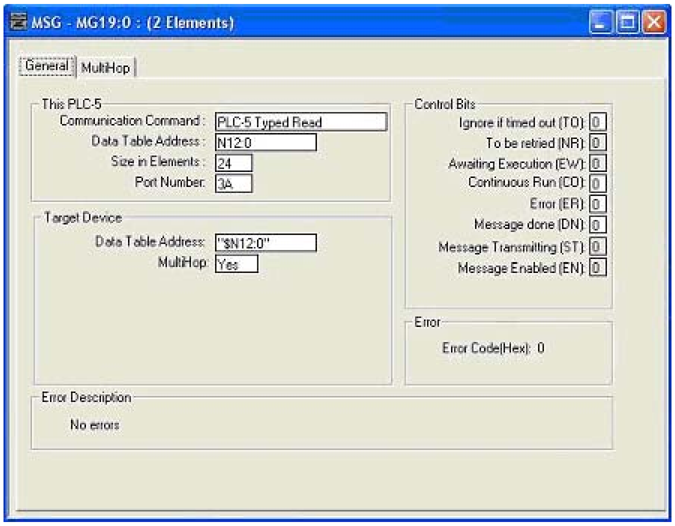
   d. Specify the port-specific transmit file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. The transmit file address must be specified in logical ASCII format. For more information on file addresses, see [3.3.3. DeviceMaster File Addressing]() on Page 78.
   e. Set the **MultiHop** option to **Yes**.

6.6.10.10. PLC-5 Typed Read - Retrieve Statistics Message

The following screen depicts an PLC-5 Typed Read - Retrieve Statistics message in ladder logic.



1. Assign a dedicated message file to the **Control** block in the ladder logic (as shown above).
2. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
3. Make the following changes:



This PLC-5
Communication Command : PLC-5 Typed Read
Data Table Address : N12:0
Size in Elements : 24
Port Number: 3A

Target Device
Data Table Address: "$N12:0"
MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0

Error
Error Code(Hex): 0

Error Description
No errors

   a. Specify the file address to receive statistics data information in the **Data Table Address** box on **This PLC-5** panel. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.

   b. Type **24** (twelve 32-bit integers) in the **Size of Elements** box.

   c. Set the **Port Number** parameter to **3A** to use the Ethernet port.

   Note: *The port number for the Ethernet channel may be different on your PLC-5.*

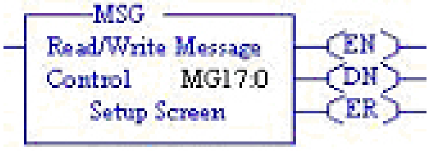
   d. Specify the port-specific statistics file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. The statistics file address must be specified in logical ASCII format. For more information on file addresses, see *3.3.3. DeviceMaster File Addressing* on Page 78.

   e. Set the **MultiHop** option to **Yes**.

6.6.10.11. PLC-5 Typed Write - Set Receive Produced Sequence Number Message

The following screen depicts an *PLC-5 Typed Write - Set Receive Produced Sequence Number* message in ladder logic.

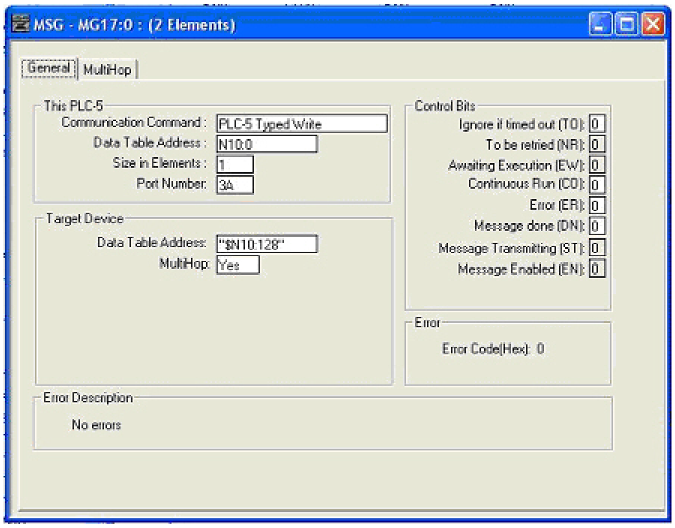
1. Assign a dedicated message file to the **Control** block in the ladder logic.



2. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
3. Make the following changes:
   a. Specify the file address where the receive sequence number resides in the **Data Table Address** box on **This PLC-5** panel. For more information on file addresses, see [3.3.3. DeviceMaster File Addressing]() on Page 78.
   b. Type **1** in the **Size of Elements** box.
   c. Set the **Port Number** parameter to **3A** to use the Ethernet port.

      Note: *The port number for the Ethernet channel may be different on your PLC-5.*

   d. Specify the port-specific receive sequence file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. The receive sequence file address must be specified in logical ASCII format. For more information on file addresses, see [3.3.3. DeviceMaster File Addressing]() on Page 78.
   e. Set the **MultiHop** option to **Yes**.



MSG - MG17:0 : (2 Elements)

This PLC-5
Communication Command : PLC-5 Typed Write
Data Table Address : N10:0
Size in Elements : 1
Port Number: 3A

Target Device
Data Table Address: "$N10:128"
MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0

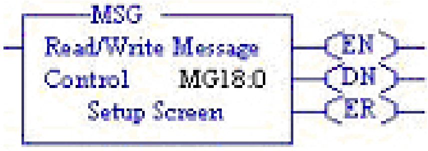
Error
Error Code(Hex): 0

Error Description
No errors

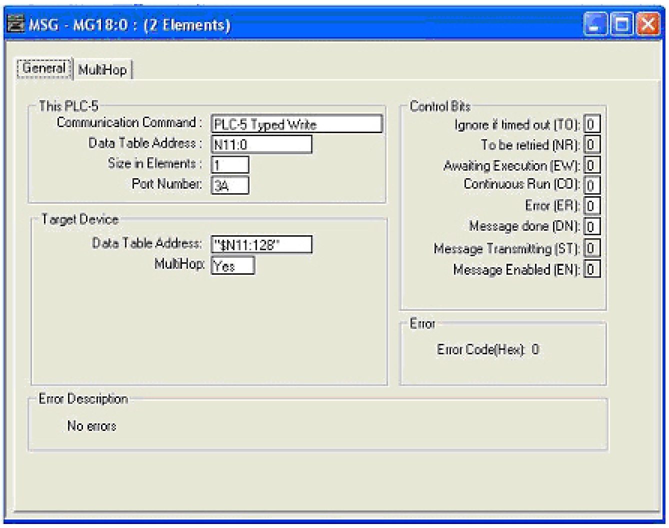
6.6.10.12. PLC-5 Typed Write - Set Transmit Produced Sequence Number Message

The following screen depicts an *PLC-5 Typed Write - Set Transmit Produced Sequence Number* message in ladder logic.

1. Assign a dedicated message file to the **Control** block in the ladder logic.



2. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.


MSG - MG18:0 : (2 Elements)

This PLC-5
Communication Command : PLC-5 Typed Write
Data Table Address : N11:0
Size in Elements : 1
Port Number: 3A

Target Device
Data Table Address: "$N11:128"
MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0

Error
Error Code(Hex): 0

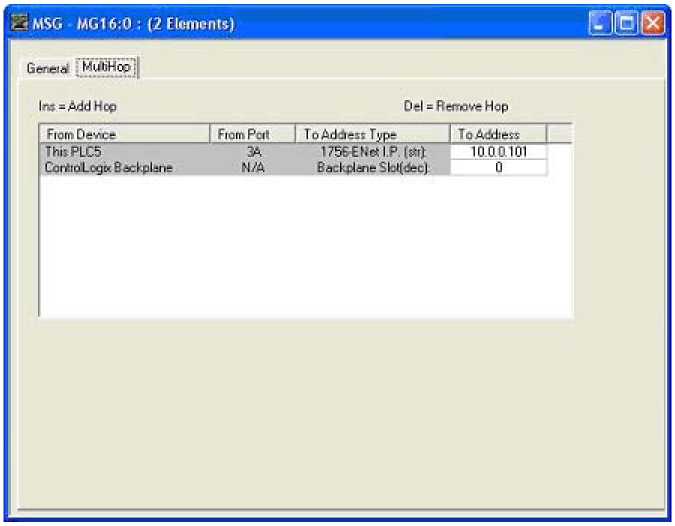
Error Description
No errors

3. Make the following changes:
   a. Specify the file address where the transmit sequence number resides in the **Data Table Address** box on **This PLC-5** panel. For more information on file addresses, see [3.3.3. DeviceMaster File Addressing]() on Page 78.
   b. Type **1** in the **Size of Elements** box.
   c. Set the **Port Number** parameter to **3A** to use the Ethernet port.

      Note: *The port number for the Ethernet channel may be different on your PLC-5.*
   d. Specify the port-specific transmit sequence number file address for the DeviceMaster in the **Data Table Address** box on the **Target Device** panel. The transmit sequence number file address must be specified in logical ASCII format. For more information on file addresses, see [3.3.3. DeviceMaster File Addressing]() on Page 78.
   e. Set the **MultiHop** option to **Yes**.

6.6.10.13. MultiHop Screen

1. Select the **MultiHop** tab on the **MSG** dialog.
2. Make the following changes:
   a. Type the IP address for the DeviceMaster (the designated EtherNet/IP Device) in the **To Address** box.
   b. Type **0** (zero) in the **To Address** box for the ControlLogix Backplane.


MSG - MG16:0 : (2 Elements)

Ins = Add Hop Del = Remove Hop

From Device	From Port	To Address Type	To Address
This PLC5	3A	1756-ENet I.P. (str):	10.0.0.101
ControlLogix Backplane	N/A	Backplane Slot(dec):	0

6.7. EDS Files

You do not need to add DeviceMaster to RSLinx for normal DeviceMaster-to-PLC communications. However, you can easily add the DeviceMaster and its associated Electronic Data Sheet (EDS) files to RSLinx.

6.7.1. Requirements

EDS files and the associated icons are included in the self-installing file (**.MSI**) and are copied to the **Comtrol/EtherNetIP** directory on your computer when you open the **.MSI** file and follow the prompts.

The files named DeviceMaster**_dd_NNNN-x.xx.eds** are ODVA electronic data sheet files where **dd** is the model name, **NNNN** is the product ID number, and **x.xx** is the version number.

File Name	Description
DeviceMaster_1P_9441-**x.xx**.eds	DeviceMaster UP 1-port DB9 panel mount
DeviceMaster_1P_9601-**x.xx**.eds	DeviceMaster EIP-2101 (1-port DB9 DIN rail)
DeviceMaster_1P_9620-**x.xx**.eds	DeviceMaster EIP-2201 (1-port serial terminal DIN rail)
DeviceMaster_2P_9531-**x.xx**.eds	DeviceMaster UP or EIP-2202 (2-port DIN rail with serial terminals and 1 Ethernet port)
DeviceMaster_2P_9541-**x.xx**.eds	DeviceMaster UP or EIP-2402 (2-port DIN rail with serial terminals and 2 Ethernet ports)
DeviceMaster_2P_9551-**x.xx**.eds	DeviceMaster UP or EIP-2102 (2-port DIN rail with DB9 and 1 Ethernet port)
DeviceMaster_2P_9561-**x.xx**.eds	DeviceMaster UP or EIP-2302 (2-port DIN rail with DB9 and 2 Ethernet ports)
DeviceMaster_4P_9447-**x.xx**.eds	DeviceMaster UP 4-port DB9 panel mount
DeviceMaster_4P_9611-**x.xx**.eds	DeviceMaster EIP-2304 (4-port DB9 DIN rail)
† *Models that have EtherNet/IP loaded at the factory are identified with an EtherNet/IP label on the DeviceMaster.*	

6.7.2. Adding DeviceMaster to RSLinx

1. Open **RSLinx**.
2. Under **Communications**, select **Configure Drivers**.
3. Under **Available Drivers**, select **Remote Devices via Linx Gateway**.
4. Select **Add New**.
5. Use the default driver name or type your own driver name and click **OK** to continue.
6. Type the IP address for the device under **Server's IP Address or Hostname** and select **OK**.
7. Select **RSWho** to verify that RSLinx can communicate with the DeviceMaster.

   Note: *A yellow question mark appears by the* DeviceMaster*(s) in the RSWho window when the associated EDS file(s) are not installed.*

6.7.3. Adding EDS Files to RSLinx

1. Open the EDS Hardware Installation Tool. (Select **Start** > **All Programs** > **Rockwell Software** > **RSLinx Tools**.)
2. Click **Add**.
3. Click **Register a directory of EDS files**.
4. Browse to the **Comtrol/EtherNetIP** directory and click **Next** to continue.

5. Verify that there is a green check beside each EDS file name and select **Next** to continue.
6. To change the icons, perform the following tasks.
   a. Select a DeviceMaster.
   b. Select **Change icon**.
   c. Browse to the **Comtrol/EtherNetIP** directory and select the icon associated with your DeviceMaster.
   Note: *You may also select your own icon stored elsewhere.*
7. Click **Next** to continue.
8. Click **Finish** to exit.

6.7.4. Troubleshooting RSLinx

If RSLinx does not display the device after adding DeviceMaster and the EDS files to RSLinx, perform the following procedure:

1. Select **File > Exit and Shutdown** to exit and shutdown RSLinx.
2. Remove the following files from your hard drive:
   - **\Program Files\Rockwell Software\RSCOMMON\Harmony.hrc**
   - **\Program Files\Rockwell Software\RSCOMMON\Harmony.rsh**
3. Restart RSLinx. The DeviceMaster unit or units should now appear with the associated icon or icons.

Chapter 7. Troubleshooting and Technical Support

You should review the *Troubleshooting* chapter in the *EtherNet/IP Hardware Installation and Configuration Guide* before calling Technical Support because they will request that you perform many of the procedures or verifications before they can help you diagnose a problem.

- *7.1. Troubleshooting Checklist* on Page 213
- *7.2. General Troubleshooting* on Page 214

If you cannot diagnose the problem, you can contact *7.3. Technical Support* on Page 214.

7.1. Troubleshooting Checklist

The following checklist may help you diagnose your problem:

- Verify that you are using the correct types of cables on the correct connectors and that all cables are connected securely.

  Note: *Most customer problems reported to Comtrol Technical Support are eventually traced to cabling or network problems.*
- Isolate the DeviceMaster from the network by connecting the device directly to a NIC in a host system.
- Verify that the Ethernet hub and any other network devices between the system and the DeviceMaster are powered up and operating.
- Reset the power on the DeviceMaster and watch the **PWR** or **Status** light activity.

PWR or Status LED	Description
5 sec. off, 3 flashes, 5 sec. off, 3 flashes...	RedBoot™ checksum failure.
5 sec. off, 4 flashes, 5 sec. off, 4 flashes...	SREC load failure.

- If the device has a power switch, turn the device's power switch off and on, while watching the LED diagnostics.
- If the DeviceMaster does not have a power switch, disconnect and reconnect the power cord.
- Verify that the network IP address, subnet mask, and gateway is correct and appropriate for the network. If IP addressing is being used, the system should be able to ping the DeviceMaster.
- Verify that the IP address programmed into the DeviceMaster matches the unique reserved IP configured address assigned by the system administrator.
- If using DHCP, the host system needs to provide the subnet mask and gateway.
- Reboot the system and the DeviceMaster.
- If you have a spare DeviceMaster, try replacing the device.

7.2. General Troubleshooting

This table illustrates some general troubleshooting tips.

Note: *Make sure that you have reviewed the 7.1. Troubleshooting Checklist on Page 213.*

General Condition	Explanation/Action
PWR or **Status** LED flashing	Indicates that boot program has not downloaded to the unit. 1. Reboot the system. 2. Make sure that you have downloaded the most current firmware for your protocol: http://www.comtrol.com/support/download.asp. ***Note:*** *If the PWR or Status LED is still flashing, contact Technical Support.*
PWR or **Status** LED not lit	Indicates that power has not been applied or there is a hardware failure. Contact Technical Support.
Cannot ping the device through Ethernet hub	Isolate the DeviceMaster from the network. Connect the device directly to the NIC in the host system (see Page 213).
Cannot ping or connect to the DeviceMaster	The default IP address is often not accessible due to the subnet masking from another network unless **192.168** is used in the network. In most cases, it is necessary to program in an address that conforms to your network.
DeviceMaster continuously reboots when connected to some Ethernet switches or routers	Invalid IP information may also cause the switch or router to check for a gateway address. Lack of a gateway address is a common cause.

7.3. Technical Support

It contains troubleshooting procedures that you should perform before contacting Technical Support since they will request that you perform, some or all of the procedures before they will be able to help you diagnose your problem. If you need technical support, use one of the following methods.

Comtrol Contact Information	
Downloads	ftp:ftp.comtrol.com/html/up_ethernet_ip_main.htm
Web site	http://www.comtrol.com
Phone	763.957.6000