



EtherNet®/IP User Guide



Trademark Notices

Control, DeviceMaster, and PortVision are registered trademarks of Control Corporation.

ControlLogix, PLC-5 and Rockwell Automation are registered trademarks of Rockwell Automation.

MicroLogix, RSLinx, RSLogix 5, RSLogix 500, RSLogix 5000 and SLC are trademarks of Rockwell Automation.

PLC is a registered trademark of Allen-Bradley Company, Inc.

Ethernet is a registered trademark of Digital Equipment Corporation, Intel, and Xerox Corporation.

Portions of SocketServer are copyrighted by GoAhead Software, Inc. Copyright © 2001. GoAhead Software, Inc. All Rights Reserved.

Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries.

Other product names mentioned herein may be trademarks and/or registered trademarks of their respective owners.

Seventh Edition, August 17, 2006

Copyright © 2005-2006. Control Corporation.

All Rights Reserved.

Control Corporation makes no representations or warranties with regard to the contents of this document or to the suitability of the Control 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 G

Patents Pending

Table of Contents

| | |
|--|-----------|
| Chapter 1. Introduction | 7 |
| 1.1. Audience | 7 |
| 1.2. Product Overview | 7 |
| 1.3. EtherNet/IP Firmware..... | 7 |
| 1.3.1. Traditional EtherNet/IP System Architecture (Firmware V2.x) | 7 |
| 1.3.2. Enhanced EtherNet/IP System Architecture (Firmware 3.x) | 8 |
| 1.3.3. Filtering and Data Extraction Functionality (Patent Pending)..... | 8 |
| 1.4. Locating Updated Software and Documents | 9 |
| 1.5. EtherNet/IP Application Setup..... | 10 |
| | |
| Chapter 2. Programming Interface | 11 |
| 2.1. Overview | 11 |
| 2.1.1. ControlLogix Family..... | 11 |
| 2.1.2. PLC-5/SLC or MicroLogix | 11 |
| 2.1.3. What is EtherNet/IP? | 12 |
| 2.2. Data Type Definitions..... | 12 |
| 2.3. EtherNet/IP Interface Profile (ControlLogix) | 13 |
| 2.3.1. Serial Port Configuration Object Definition (70 Hex) | 13 |
| 2.3.1.1. Class Attributes | 13 |
| 2.3.1.2. Instance Attributes | 13 |
| 2.3.1.3. Common Services | 17 |
| 2.3.1.4. Instance Attribute Definitions | 17 |
| 2.3.2. Serial Port Data Transfer Object Definition (71 Hex)..... | 25 |
| 2.3.2.1. Class Attributes | 25 |
| 2.3.2.2. Instance Attributes | 25 |
| 2.3.2.3. Common Services | 25 |
| 2.3.2.4. Instance Attribute Definitions | 26 |
| 2.3.3. Serial Port Statistics Object Definition (72 Hex)..... | 27 |
| 2.3.3.1. Class Attributes | 27 |
| 2.3.3.2. Instance Attributes | 27 |
| 2.3.3.3. Common Services | 27 |
| 2.3.3.4. Instance Attribute Definitions | 28 |
| 2.3.4. Socket Port Configuration Object Definition (73 Hex) | 29 |
| 2.3.5. Socket Port Data Transfer Definition Object (74 Hex)..... | 37 |
| 2.3.5.1. Class Attributes | 37 |
| 2.3.5.2. Instance Attributes | 37 |
| 2.3.5.3. Common Services | 37 |
| 2.3.5.4. Instance Attribute Definitions | 38 |
| 2.3.6. Informational Objects | 39 |
| 2.3.6.1. Identity Object (01 Hex, 1 instance) | 39 |
| 2.3.6.1.1. Class Attributes | 39 |
| 2.3.6.1.2. Instance Attributes..... | 39 |
| 2.3.6.1.3. Status Word..... | 39 |
| 2.3.6.1.4. Common Services..... | 39 |
| 2.3.6.2. Message Router Object (02 Hex) | 40 |
| 2.3.6.2.1. Class Attributes | 40 |
| 2.3.6.2.2. Instance Attributes..... | 40 |
| 2.3.6.2.3. Common Services..... | 40 |
| 2.3.6.3. Connection Manager Object (06 Hex) | 40 |
| 2.3.6.3.1. Instance Object Specific Services..... | 40 |
| 2.3.6.4. Port Object (F4 Hex - 1 instance) | 41 |
| 2.3.6.4.1. Class Attributes | 41 |
| 2.3.6.4.2. Instance Attributes..... | 41 |

| | |
|---|-----------|
| 2.3.6.4.3. Common Services | 41 |
| 2.3.6.5. TCP Object (F5 Hex - 1 instance)..... | 42 |
| 2.3.6.5.1. Class Attributes..... | 42 |
| 2.3.6.5.2. Instance Attributes | 42 |
| 2.3.6.5.3. Common Services | 42 |
| 2.3.6.6. Ethernet Link Object (F6 Hex)..... | 43 |
| 2.3.6.6.1. Class Attributes..... | 43 |
| 2.3.6.6.2. Instance Attributes | 43 |
| 2.3.6.6.3. Common Services | 43 |
| 2.3.6.7. PCCC Object (67 Hex)..... | 44 |
| 2.3.6.7.1. Class Attributes..... | 44 |
| 2.3.6.7.2. Instance Attributes | 44 |
| 2.3.6.7.3. Instances..... | 44 |
| 2.3.6.7.4. Common Services | 44 |
| 2.3.6.7.5. Message Structure for Execute PCCC | 44 |
| 2.4. PLC-5/SLC Interface | 45 |
| 2.4.1. Requirements..... | 46 |
| 2.4.1.1. SLC 5/05 | 46 |
| 2.4.1.2. PLC-5..... | 46 |
| 2.4.2. Messages | 47 |
| 2.4.2.1. DeviceMaster UP File Addressing | 47 |
| 2.4.3. Receive Data Message | 48 |
| 2.4.4. Transmit Data Message | 49 |
| 2.4.5. Sequence Number Messages | 49 |
| 2.4.6. Retrieve Statistics Message | 50 |
| 2.4.7. Receive Communication Methods..... | 52 |
| 2.4.7.1. Unsolicited - Write to File Receive Method | 52 |
| 2.4.7.2. Unsolicited - Write to File Synced Receive Method | 52 |
| 2.4.7.3. Polling Receive Method..... | 53 |
| Chapter 3. Configuring the DeviceMaster UP | 55 |
| 3.1. Port Configuration Overview | 55 |
| 3.2. Embedded Web Pages Overview | 56 |
| 3.3. Serial Device Configuration Page | 57 |
| 3.3.1. Edit Serial Port Configuration Page..... | 57 |
| 3.3.1.1. Serial Configuration | 57 |
| 3.3.1.2. Serial Packet Identification | 59 |
| 3.4. Ethernet Device Configuration Page..... | 61 |
| 3.5. Edit Socket Port Configuration Page | 61 |
| 3.6. Common Configuration Areas (Serial or Ethernet Device)..... | 63 |
| 3.6.1. EtherNet/IP Settings..... | 63 |
| 3.6.2. Filtering/Data Extraction Configuration | 65 |
| 3.6.3. Application TCP Connection Configuration..... | 68 |
| 3.6.4. Saving Port Options | 69 |
| 3.7. Serial/Ethernet Device Communication Statistics..... | 70 |
| 3.8. PLC Interface Diagnostics..... | 73 |
| 3.9. Edit Network Configuration Page..... | 75 |

| | |
|---|-----------|
| Chapter 4. Programming the PLC | 77 |
| 4.1. Programming Guidelines..... | 77 |
| 4.2. PLC Algorithms..... | 78 |
| 4.2.1. Unsolicited - Write-to-Tag/File PLC Algorithm | 78 |
| 4.2.2. Unsolicited - Write-to-Tag/File-Synced PLC Algorithm | 79 |
| 4.2.3. Polling PLC Algorithm | 80 |
| 4.3. ControlLogix PLC Programming Example Instructions | 81 |
| 4.3.1. What is RSLogix 5000?..... | 81 |
| 4.3.2. Requirements | 81 |
| 4.3.3. loopbackExampleTagWrite.L5K..... | 82 |
| 4.3.4. loopbackExampleTagWriteSynced.L5K..... | 83 |
| 4.3.5. loopbackExamplePolling.L5K..... | 84 |
| 4.3.6. Configuring the DeviceMaster UP for the RSLogix 5000 Example Programs Using the Web Page | 85 |
| 4.3.7. RSLogix 5000 Screen Examples..... | 86 |
| 4.3.7.1. Transmit Data to the DeviceMaster UP | 86 |
| 4.3.7.2. Send Configuration to DeviceMaster UP..... | 87 |
| 4.3.7.3. Request Data from DeviceMaster UP | 89 |
| 4.3.7.4. Send Consumed Sequence Number to DeviceMaster UP | 90 |
| 4.3.7.5. Request Statistics from DeviceMaster UP | 91 |
| 4.3.7.6. Communication Window for all Messages sent to the DeviceMaster UP | 92 |
| 4.3.8. Modifying an RSLogix 5000 PLC Program Example (Older Versions) | 93 |
| 4.4. SLC or MicroLogix PLC Programming Example Instructions..... | 94 |
| 4.4.1. What is RSLogix 500?..... | 94 |
| 4.4.2. Requirements | 94 |
| 4.4.3. Example Program Considerations | 95 |
| 4.4.4. lpbkExampleSlcMsgPollRS500..... | 95 |
| 4.4.5. lpbkExamplePlc5MsgPollRS500..... | 95 |
| 4.4.5.1. lpbkExampleSlcMsgFileRS500 | 95 |
| 4.4.5.2. lpbkExampleSlcMsgFileSyncRS500 | 96 |
| 4.4.6. Configure the DeviceMaster UP for the RSLogix 500 Example Program | 97 |
| 4.4.7. Configure and Run the RSLogix 500 Example Program | 98 |
| 4.4.8. RSLogix 500 Screen Examples..... | 101 |
| 4.4.8.1. Setting up Processor and EtherNet (Channel 1)..... | 101 |
| 4.4.8.2. SLC Typed Read - Receive Data Message | 104 |
| 4.4.8.3. SLC Typed Write - Transmit Data Message | 105 |
| 4.4.8.4. SLC Typed Read - Retrieve Statistics Message | 106 |
| 4.4.8.5. SLC Typed Write - Set Receive Produced Sequence Number Message | 107 |
| 4.4.8.6. SLC Typed Write - Set Transmit Produced Sequence Number Message | 108 |
| 4.4.8.7. PLC-5 Typed Read - Receive Data Message | 109 |
| 4.4.8.8. PLC-5 Typed Write - Transmit Data Message | 110 |
| 4.4.8.9. PLC-5 Typed Read - Retrieve Statistics Message | 111 |
| 4.4.8.10. PLC-5 Typed Write - Set Receive Produced Sequence Number Message..... | 112 |
| 4.4.8.11. PLC-5 Typed Write - Set Transmit Produced Sequence Number Message..... | 113 |
| 4.4.8.12. MultiHop Screen | 113 |

4.5. PLC-5 PLC Programming Example Instructions..... 114

4.5.1. What is RSLogix 5?..... 114

4.5.2. Requirements 114

4.5.3. Example Program Considerations 115

4.5.4. lpbkExampleSlcMsgPollRS5 115

4.5.5. lpbkExamplePlc5MsgPollRS5..... 115

4.5.6. lpbkExamplePlc5MsgFileRS500..... 115

4.5.7. lpbkExamplePlc5MsgFileSyncRS5..... 116

4.5.8. Configure the DeviceMaster UP for the RSLogix 5 Program..... 117

4.5.9. Configure and Run the Example RSLogix 5 Program..... 118

4.5.10. RSLogix 5 Screen Examples 120

4.5.10.1. Requirements 120

4.5.10.2. Setting up Processor and EtherNet Channel 120

4.5.10.3. SLC Typed Read - Receive Data Message 122

4.5.10.4. SLC Typed Write - Transmit Data Message 123

4.5.10.5. SLC Typed Read - Retrieve Statistics Message 124

4.5.10.6. SLC Typed Write - Set Receive Produced Sequence Number Message 125

4.5.10.7. SLC Typed Write - Set Transmit Produced Sequence Number Message 126

4.5.10.8. PLC-5 Typed Read - Receive Data Message 127

4.5.10.9. PLC-5 Typed Write - Transmit Data Message 128

4.5.10.10. PLC-5 Typed Read - Retrieve Statistics Message 129

4.5.10.11. PLC-5 Typed Write - Set Receive Produced Sequence Number Message..... 130

4.5.10.12. PLC-5 Typed Write - Set Transmit Produced Sequence Number Message..... 131

4.5.10.13. MultiHop Screen 131

4.6. EDS Files..... 132

4.6.0.1. Requirements 132

4.6.0.2. Adding DeviceMaster UP to RSLinx..... 132

4.6.0.3. Adding EDS Files to RSLinx 132

4.6.0.4. Troubleshooting RSLinx 133

Index..... 135

Chapter 1. Introduction

1.1. Audience

The primary audience of this document is the person responsible for installing the DeviceMaster UP 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

1.2. Product Overview

The DeviceMaster UP provides an EtherNet/IP interface for 1 and 4-port models.

In order for the DeviceMaster UP to operate as an EtherNet/IP gateway, you must upload the EtherNet/IP firmware over the default application on the DeviceMaster UP. The DeviceMaster UP then provides an EtherNet/IP interface to devices with raw/ASCII serial and Ethernet TCP/IP interfaces.

If you have not installed the hardware, uploaded the EtherNet/IP firmware, and configured the network information on the DeviceMaster UP, you must do so before performing the procedures in this *User Guide*.

For setup procedures, use the *DeviceMaster UP Hardware Installation Guide*. See [1.4. Locating Updated Software and Documents](#) on Page 9 to locate the document and firmware.

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

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

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

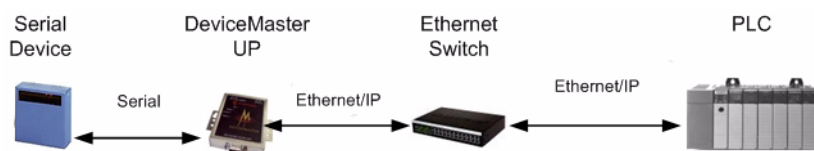
1.3. EtherNet/IP Firmware

The following subsections provides information for existing users who may or may not want to update systems with the enhanced EtherNet/IP firmware V3.x.

For new users, the following subsections provide EtherNet/IP system architecture information.

1.3.1. Traditional EtherNet/IP System Architecture (Firmware V2.x)

EtherNet/IP firmware V2.x provided a traditional EtherNet/IP interface to devices through a raw/ASCII serial interface as illustrated.



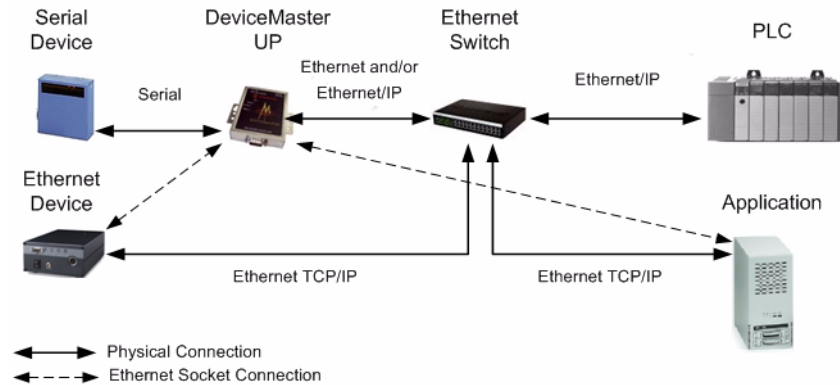
Traditional EtherNet/IP System Architecture (Firmware V2.x)

1.3.2. Enhanced EtherNet/IP System Architecture (Firmware 3.x)

Using the Ethernet TCP/IP firmware V3.x doubles the capacity of the DeviceMaster UP by providing a raw/ASCII interface to both serial and Ethernet TCP/IP devices.

For example:

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



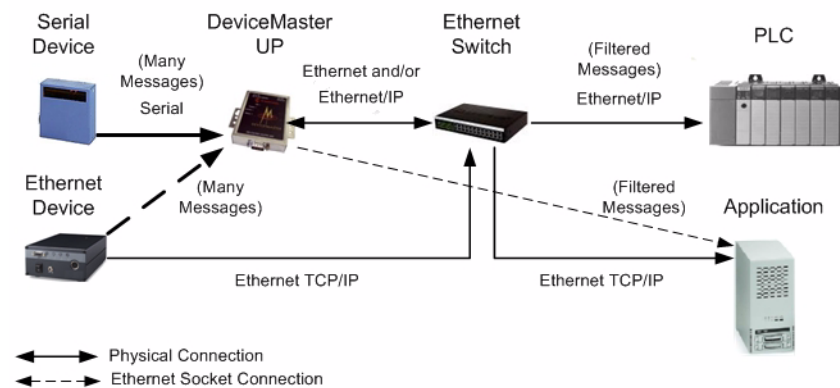
Control EtherNet/IP System Architecture (V3.x)

EtherNet/IP firmware 3.x 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.

1.3.3. Filtering and Data Extraction Functionality (Patent Pending)

EtherNet/IP firmware 3.x 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.



EtherNet/IP Filtering Functionality (V3.x)


If you are running firmware V2.x, you may want to review the following list additional features provided in firmware V3.x.


- Improved PLC interfaces


- Increased PLC interface bandwidth provides improved performance and lower latencies when interfacing to higher bandwidth serial and Ethernet devices such as RFID readers.
- Transfer of large received serial device packets up to 1024 bytes.
- Transfer of large received Ethernet device packets up to 2048 bytes.
- Throttling of received data to the PLC in the *Write-to-Tag/File* receive mode ensures data received by the PLC is not overwritten before it can be processed.
- Disabling of non-filtered receive queue ensures the PLC will only receive the latest received serial/Ethernet device data.
- New embedded web pages
 - *PLC Interface Diagnostics* page, which provides statistics and error messages to monitor and help diagnose PLC interface problems.
 - *Serial/Ethernet Device Communication Statistics* page, which is a comprehensive statistics page for all serial and Ethernet device interfaces. Includes packet, byte, and error counts to the PLC(s) and Application(s) as well as comprehensive filtering statistics.
 - *Ethernet Device Interface Configuration* page, which provides a user interface to the Ethernet device interface configuration.



1.4. Locating Updated Software and Documents


You can access the firmware software assembly, PortVision Plus, and the *DeviceMaster UP* documentation from the CD shipped with the DeviceMaster UP or you can download the latest files using these links.

| DeviceMaster UP | Download from Web |
|---|---|
| <i>Hardware Installation Guide</i> contains hardware installation and firmware updating procedures. |  |

| PortVision Plus | Download from Web |
|---|---|
| <i>PortVision Plus</i> is the application that you use to configure network settings and upload the firmware. |  |

| Firmware | Download from Web |
|---|---|
| EtherNet/IP (.msi) file contains the firmware and supporting files. |  Note: <i>If you are currently running the EtherNet/IP firmware V2.x platform, review 1.3. EtherNet/IP Firmware on Page 7 to see if you want to add the architecture enhancement using EtherNet/IP V3.x or higher.</i> |

| Protocol Documentation | Download from Web |
|---|--|
| <i>EtherNet/IP User Guide</i> contains protocol specific information about configuring the DeviceMaster UP. |  |
| <i>EtherNet/IP Quick Start</i> contains abbreviated installation and configuration procedures for the DeviceMaster UP with links to detailed documentation. |  |

| Filtering and Data Extraction | Download from Web |
|---|--|
| The <i>DeviceMaster UP Filtering and Data Extraction Reference Guide</i> describes the data extraction and filtering processes provided by the DeviceMaster UP. |  |

Optionally, you can check for any updated files using this URL: <http://support.comtrol.com/download.asp?product=DMUP>.

1.5. EtherNet/IP Application Setup

To configure EtherNet/IP protocol on the DeviceMaster UP after installing the hardware, uploading the EtherNet/IP firmware, and configuring the network address, follow these basic steps.

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

| Interfaces | Programming Procedure |
|-----------------------|--|
| ControlLogix PLC | <ul style="list-style-type: none"> • Program the PLC. See the instructions in 4.3. ControlLogix PLC Programming Example Instructions on Page 81. • (Optional) Access the DeviceMaster UP <i>Server Configuration</i> page to configure the serial/socket port settings, if you did not configure the serial/socket port setting in the PLC program (Chapter 3. Configuring the DeviceMaster UP on Page 55). |
| SLC or MicroLogix PLC | <ul style="list-style-type: none"> • Program the SLC or MicroLogix PLC, see the instructions in 4.4. SLC or MicroLogix PLC Programming Example Instructions on Page 94. • Access the DeviceMaster UP <i>Server Configuration</i> page to configure the serial/socket port settings. (Chapter 3. Configuring the DeviceMaster UP on Page 55). |
| PLC-5 PLC | <ul style="list-style-type: none"> • Program the PLC-5 PLC, see the instructions in 4.5. PLC-5 PLC Programming Example Instructions on Page 114. • Access the DeviceMaster UP <i>Server Configuration</i> page to configure the serial/socket port settings. (Chapter 3. Configuring the DeviceMaster UP on Page 55). |

2. Connect your serial device or devices. If necessary, refer to the *DeviceMaster UP Hardware Installation Guide*.

Chapter 2. Programming Interface

2.1. Overview

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

2.1.1. ControlLogix Family

The DeviceMaster UP supports ControlLogix PLCs. You can configure DeviceMaster UP through the PLC program or the DeviceMaster UP *Server Configuration* web page. The DeviceMaster UP interface for the ControlLogix PLC is described in [2.3. EtherNet/IP Interface Profile \(ControlLogix\)](#) on Page 13.

The [EtherNet/IP Quick Start](#) describes the quick start for read-only devices such as barcode scanners and read/write devices such as printers.

[4.3. ControlLogix PLC Programming Example Instructions](#) on Page 81 provides descriptions of the RSLogix 5000 PLC program examples intended to aid the PLC programmer.

You can also use the *Server Configuration* page on the DeviceMaster UP when you do not want to use the PLC programming interface to configure the DeviceMaster UP. [Chapter 3. Configuring the DeviceMaster UP](#) on Page 55 describes the web pages on the DeviceMaster UP and provides instructions for configuring the serial/socket port settings

2.1.2. PLC-5/SLC or MicroLogix

The DeviceMaster UP supports PLC-5, SLC and MicroLogix PLCs. You must configure the DeviceMaster UP through the web pages on the DeviceMaster UP. The DeviceMaster UP interface for the PLC-5/SLC or MicroLogix PLC is described in [2.4. PLC-5/SLC Interface](#) on Page 45.

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 [4.4. SLC or MicroLogix PLC Programming Example Instructions](#) on Page 94. There you will find 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 [4.5. PLC-5 PLC Programming Example Instructions](#) on Page 114. There you will find descriptions of the RSLogix 5 PLC program examples intended to aid the PLC programmer.

2.1.3. 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 protocols, it uses long established EtherNet hardware and software to define an application layer protocol for configuring access and controlling industrial automation devices. EtherNet/IP classifies EtherNet nodes as predefined device types with specific behaviors. 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.

2.2. 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) |

2.3. EtherNet/IP Interface Profile (ControlLogix)

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

2.3.1. 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 UP over EtherNet/IP.
- An optional application can communicate with a serial device through the DeviceMaster UP 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 UP. (Port numbers are numbered from one to N.)*

You can disregard this object definition if you configure the DeviceMaster UP using the Server Configuration web page. See [Chapter 3. Configuring the DeviceMaster UP](#) on Page 55 to configure the DeviceMaster UP using the embedded web page.

2.3.1.1. 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 UP | Get |
| 3 | Num Instances | UINT | Number of ports on the DeviceMaster UP | Get |

2.3.1.2. Instance Attributes

| Attribute ID | Name | Data Type | Data Value(s) | Access Rule |
|--------------|-------------------------|-----------|--|---|
| 1 | Serial Port Device Type | UDINT | 0=Raw Data Device (Default=0) | Set/Get |
| 2 | Serial Port Commands | DWORD | 1=Reset Serial Port 2=Save in Flash 4=Clear Sequence Counters 8=Clear Statistics Counters (Default=0) | Set/Get (Get returns last command sent) |
| 3 | Baud Rate | UDINT | Valid rates: 300, 600, 1200, 2400, 4800, 9600 (Default), 19200, 38400, 57600, 115200, 230400 | Set/Get |
| 4 | Interface Mode | USINT | 0=RS-232 (Default) 1=RS-422 2=RS-485 | Set/Get |
| 5 | Parity | USINT | 0=None (Default) 1=Even 2=Odd | Set/Get |
| 6 | Data Bits | USINT | Valid Values: 5-8 (Default=8) | Set/Get |
| 7 | Stop Bits | USINT | Valid Values: 1 or 2 (Default=1) | Set/Get |
| 8 | Flow Control | USINT | 0=None (Default) 1=RTS/CTS 2=XON/XOFF 3=Half Duplex | Set/Get |
| 9 | DTR Control | USINT | 0=Off (Default) 1=On | Set/Get |

| Attribute ID | Name | Data Type | Data Value(s) | Access Rule |
|--------------|--|-------------------------|--|-------------|
| 10 | PLC Transmit STX Append 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 STX) (Default=0) 0 to 255 0 to 255 | Set/Get |
| 11 | PLC Transmit ETX Append 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=0) 0 to 255 0 to 255 | Set/Get |
| 12 | Reserved | UINT | 0 | Get |
| 13 | Receive STX 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 STX) (Default=1) 0 to 255 0 to 255 (Default: Value1 = 2) | Set/Get |
| 14 | 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 | 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 | 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) | Set/Get |

| Attribute ID | Name | Data Type | Data Value(s) | Access Rule |
|--------------|---|----------------------------|---|-------------|
| 17 | Receive (DeviceMaster UP to PLC) Ethernet Data Transfer Method | USINT | 0=OFF 1=Unsolicited - Write-to-Tag 2=Unsolicited - Write-to-Tag-Synced 3=Polling (Default=3) | Set/Get |
| 18 | Reserved | USINT | 0 | Get |
| 19 | Reserved | UINT | 0 | Get |
| 20 | Maximum 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 | 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 UP to PLC) Produced Data Tag Name | STRING (Array of 40 SINTs) | ASCII string | Set/Get |
| 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 |

| Attribute ID | Name | Data Type | Data Value(s) | Access Rule |
|--------------|---|-----------------------------|--|-------------|
| 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 |
| 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 | Application Transmit STX Append 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 STX) (Default=0) 0 to 255 0 to 255 | Set/Get |
| 43 | Application Transmit ETX Append 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=0) 0 to 255 0 to 255 | Set/Get |
| 44 | Reserved | USINT[2] | 0 | Get |

2.3.1.3. 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 |

2.3.1.4. Instance Attribute Definitions

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: DeviceMaster UP supports the following commands:

- **Reset serial port** - This option resets the serial port hardware and statistics counters. You must reset the DeviceMaster UP 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 UP.
- **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 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 UP 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 UP finds an STX byte it collects the data. If the first byte is not the STX byte, the DeviceMaster UP discards the byte. The DeviceMaster UP continues to discard the bytes until the DeviceMaster UP finds an STX byte.
- **2 (two STX bytes)** - Scans serial data for two STX bytes. If the DeviceMaster UP finds two STX bytes it collects the data. If the first two bytes are not the STX bytes, the DeviceMaster UP discards the bytes. Bytes continue to be discarded until the DeviceMaster UP finds two STX bytes.
- **Value1** - Specifies the character that represents the first STX byte. DeviceMaster UP 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 UP 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 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 UP 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 UP 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 UP 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 attribute specifies:

- How long DeviceMaster UP 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:

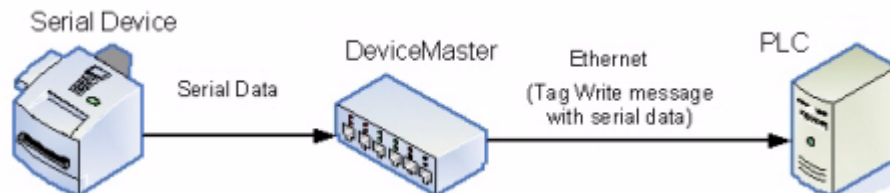
- **01 Hexadecimal (Bit 0)** - DeviceMaster UP strips received STX/ETX characters from received packets before sending the packet to the PLC.
- **02 Hexadecimal (Bit 1)** - DeviceMaster UP discards received packets with errors.
- **04 Hexadecimal (Bit 2)** - (PLC-5/SLC only) DeviceMaster UP receives 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 UP 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 UP 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 UP 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 UP strips received STX/ETX characters from received packets before sending the packet to the application.

Attribute 17 - Receive (DeviceMaster UP to PLC) EtherNet Data Transfer Method - This attribute specifies the EtherNet data transfer method used by the DeviceMaster UP. There are three methods that DeviceMaster UP can use to transfer data received from a serial device to the PLC. These methods are:

- Off - DeviceMaster UP will not allow any data to be sent to the PLC.
- *Unsolicited - Write-to-Tag* receive method - DeviceMaster UP writes the received serial data directly into tag(s) on the PLC.

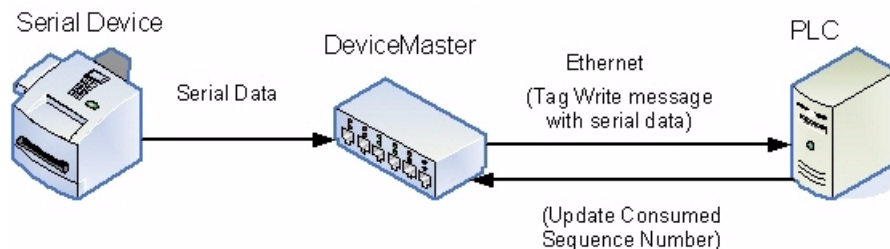
The following diagram shows the Write-to-Tag receive data flow:



The following restrictions apply to this method:

- **The Receive Data Tag Name** (Attribute 24) must have the same name as the tag defined on the PLC. (For more information, see Attribute 24 on Page 20.)
- The tag on the PLC must be of type SINT and must be large enough to contain the sequence number, length, and data field associated with the received data structure. (For more information, see [2.3.2. Serial Port Data Transfer Object Definition \(71 Hex\)](#) on Page 25.)
- An incremented sequence number indicates new data.
- The PLC program must scan and consume new data faster than the data can be received on the serial port to ensure that no data is lost.
- *Unsolicited - Write-to-Tag-Synced* receive method - DeviceMaster UP writes the serial data into tag(s) on the PLC and provides a mechanism to synchronize the data flow between the PLC and DeviceMaster UP.

In this method, DeviceMaster UP does not write the serial packet to the tag on the PLC until the PLC updates the consumed sequence number (**Serial Port Data Transfer** object Attribute 4) to match the produced data sequence number. Then the DeviceMaster UP writes the data into the “tag” data location on the PLC in the same way as the *Unsolicited - Write-to-Tag* receive method. For more information, see the description of Attribute 4 in [2.3.2. Serial Port Data Transfer Object Definition \(71 Hex\)](#) on Page 25. The following diagram shows the *Write-to-Tag-Synced* received data flow:



The following restrictions apply to this method:

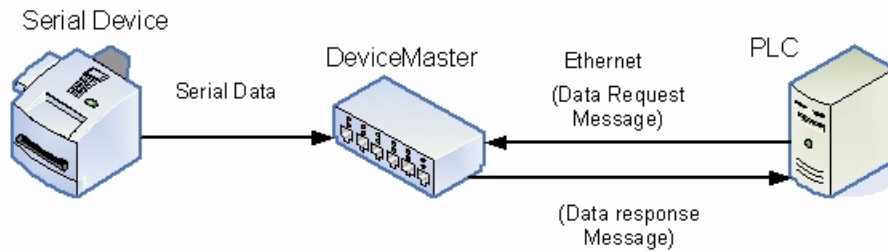
- **The Receive Data Area Tag Name** (Attribute 24) must have the same name as the tag defined on the PLC.
- The tag on the PLC must be an SINT type and large enough to contain the sequence number, length, and data field associated with the received data structure. (For more information, see the [2.3.2. Serial Port Data Transfer Object Definition \(71 Hex\)](#) on Page 25.)
- An incremented sequence number indicates new data.
- The DeviceMaster UP will not write new data to the tag on the PLC until the consumed sequence number has been incremented to match the last produced sequence number.
- While the DeviceMaster UP queues received serial 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 UP do not overflow. (For example: If the serial port can receive two serial packets per

second, then the consumption rate must be at least one packet every 500 microsecond.)

- *Polling* receive method - The PLC requests data on a periodic basis.

In this method, DeviceMaster UP returns the serial data in the response to the data request message. The PLC requests data by accessing the **Serial Port Data Transfer Object Attribute 2**. For more information, see the description of Attribute 2 in [2.3.2. Serial Port Data Transfer Object Definition \(71 Hex\)](#) on Page 25.

The following diagram shows the polling receive data flow:



The following restrictions apply to this method:

- The data tag to receive the data on the PLC must be large enough to contain the sequence number, length, and data field associated with the received data structure. (For more information, see the [2.3.2. Serial Port Data Transfer Object Definition \(71 Hex\)](#) on Page 25.)
- An incremented sequence number indicates new data.
- The same data may be returned more than once. However, the same data packet will also return the same sequence number.
- A length of zero (0) indicates no data.
- While the DeviceMaster UP 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 UP do not overflow. (For example: If the serial port can receive two packets per second, then the polling rate must be at least once every 500 microseconds.)

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 UP 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 method does not use this attribute.*

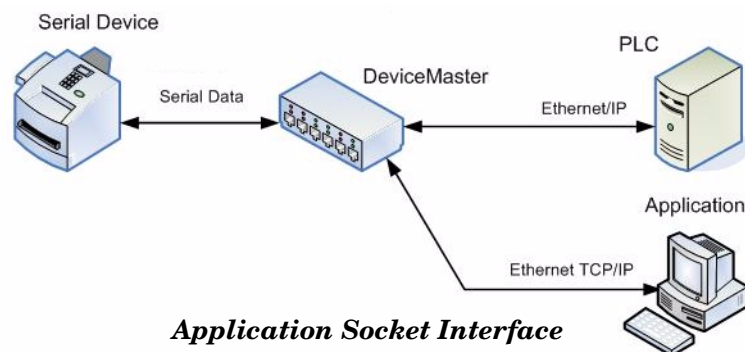
Attribute 23 - PLC IP Address: This attribute specifies the IP address in hexadecimal format for the PLC EtherNet/IP card. For example, an IP address of 10.1.2.100 is 0A010264 in hexadecimal.

Note: *The Polling method does not use this attribute.*

Attribute 24 - Receive (DeviceMaster UP to PLC) Produced Data Tag Name - This attribute 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.



Attribute 26 - Application Listen Enable: Enabling this setting allows the application to connect to the DeviceMaster UP via an Ethernet TCP/IP socket.

- 0 = Disables listening - The DeviceMaster UP will not accept connection attempts.
- 1 = Enables listening - The DeviceMaster UP will accept connection attempts from the application socket port.

Attribute 27 - Application Connect Mode: This setting controls if and how the DeviceMaster UP attempts to connect to the application at the application connection IP address and application connection socket port.

- 0 = Never - The DeviceMaster UP will not attempt to connect to the application.
- 1 = Connect Always - The DeviceMaster UP will attempt to connect to the application socket port until a connection is made.
- 2 = Connect On Data - The DeviceMaster UP will 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 UP will attempt to connect to the application until a connection is made.

Attribute 28 - Application Disconnect Mode: This setting controls if and how the DeviceMaster UP disconnects from an application.

- 0 = Never - The DeviceMaster UP will not disconnect from the application socket port.
- 1 = Disconnect On Idle - The DeviceMaster UP will disconnect 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 UP the application will connect to if **Application Listen Enable** is enabled.

Attribute 30 - Application Connection Socket Port: This is the application socket port number the DeviceMaster UP will connect 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 UP will connect 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 23.

See the [DeviceMaster UP Filtering and Data Extraction Reference Guide](#) (Page 10) for further details.

Attribute 34 - 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 23.

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 10) for further details.

Attribute 35 - 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 36 - RFID Antenna Grouping: This attribute is applicable only to RFID filtering and only if the **Antenna** filtering option is enabled. It allows the DeviceMaster UP 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 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 10) 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 10) for further details.

Attribute 39 - Filter Age Time: This attribute defines the time a filter string, RFID tag, or barcode will continue 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 will not be sent to the PLC and/or application. However, if the **Filter Age Time** has passed, it will pass 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](#) (Page 10) for further details.

Attribute 41 - Barcode Formats: 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.

The standard and eight digit formats are selected independently and each operates independently. It is important to note that the barcode filtering/data extraction will not function if no format is selected.

| Format | Numbering Digits | Company Digits | Product Digits | Check Digit |
|----------------------------|------------------|----------------|----------------|-------------|
| Standard Formats | | | | |
| None | N/A | N/A | N/A | N/A |
| Company-5/ Product-5 | 1-3 | 5 | 5 | 1 |
| Company-6/ Product-4 | 1-3 | 6 | 4 | 1 |
| Company-7/ Product-3 | 1-3 | 7 | 3 | 1 |
| Company-8/ Product-2 | 1-3 | 8 | 2 | 1 |
| Company-9/ Product-1 | 1-3 | 9 | 1 | 1 |
| Eight Digit Formats | | | | |
| EAN-8 Number-2/Product 5 | 2 | 0 | 5 | 1 |
| EAN-8 Number-3/Product 4 | 3 | 0 | 4 | 1 |
| UPC-E | 1 | Variable | Variable | 1 |

See the [DeviceMaster UP Filtering and Data Extraction Reference Guide](#) (Page 10) for further details.

Attribute 42 - 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 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 43 - 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 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.

2.3.2. 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 UP 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 UP. (Port numbers are numbered from one to N.)*

2.3.2.1. 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 UP | Get |
| 3 | Num Instances | UINT | Number of ports on the DeviceMaster UP | Get |

2.3.2.2. Instance Attributes

| Attribute ID | Name | Data Type | Data Value(s) | Access Rule |
|--------------|---|--------------------------------|---------------------------------------|-------------|
| 1 | Transmit (PLC to DeviceMaster UP) message data. Structure of: Produced data sequence number Data length Data array | UINT UINT Array of USINT | 0-65535 (FFFF hex) 1-440 0-255 | Set/Get |
| 2 | Receive (DeviceMaster UP to PLC) message data. Structure of: Produced data sequence number Data length Data array | UINT UINT Array of USINT | 0-65535 (FFFF hex) 0-440* 0-255 | Get |
| 3 | Receive (DeviceMaster UP to PLC) produced data sequence number - Normally sent to the PLC from the DeviceMaster UP in data transfer. | UINT | 0-65535 (FFFF hex) | Set/Get |
| 4 | Receive (DeviceMaster UP to PLC) consumed sequence number - normally updated by the PLC in the <i>Write-to-Tag-Synced</i> receive method. | UINT | 0-65535 (FFFF hex) | Set/Get |
| 5 | Transmit (PLC to DeviceMaster UP) produced data sequence number - normally sent to the DeviceMaster UP 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 *Write-to-Tag* and *Write-to-Tag-Synced* receive method is 1024 bytes.

2.3.2.3. 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 |

2.3.2.4. Instance Attribute Definitions

Attribute 1 - Transmit (PLC to DeviceMaster UP) Message Data: This attribute transmits data out of a serial port on the DeviceMaster UP.

- 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 18.)
- 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 UP to PLC) Message Data: This attribute provides the receive data while operating in the Polling communication method.

- DeviceMaster UP 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.
- Serial packets up to 1024 bytes may be received while operating in the *Write-To-Tag* or *Write-To-Tag-Synced* received methods. For serial packets over 440 bytes, the DeviceMaster UP will place the data into a sequence of tags. These tags must meet the following criteria:
 - All must be of type SINTs.
 - The entire sequence 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 will not be numbered (i.e. Com1_RxData), the second tag will have a “2” appended (i.e. Com1_RxData2), the third will have 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 will have been received and the PLC can process the data.

Attribute 3 - Receive (DeviceMaster UP 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 UP 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 19.

Attribute 5 - Transmit (PLC to DeviceMaster UP) 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 UP in the *Transmit Message data*.

2.3.3. Serial Port Statistics Object Definition (72 Hex)

The *Serial Port Statistics* object defines the statistics gathered by the DeviceMaster UP 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 UP. (Port numbers are numbered from one to N.)*

2.3.3.1. 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 UP | Get |
| 3 | Num Instances | UINT | Number of ports on the DeviceMaster UP | Get |

2.3.3.2. 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 |

2.3.3.3. 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 |

2.3.3.4. Instance Attribute Definitions

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 UP 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 UP 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 UP 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 18 for additional information.)
- DeviceMaster UP receives a transmit message with a sequence number equal to the previous sequence number. (The DeviceMaster UP expects this sequence number to be incremented by one from the sequence number in the previous transmit message.)

Attribute 11 - Unexpected Transmit Sequence Number Error Count: This attribute counts the number of *Unexpected Transmit Sequence Number* errors. The DeviceMaster UP 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 18 for additional information.)
- DeviceMaster UP 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 UP 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

2.3.4. 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 UP over Ethernet/IP.
- An optional application can communicate with the Ethernet device through the DeviceMaster UP 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 UP. (Socket port numbers can be numbered form 1 to N.)

You can disregard this object definition if you configure the DeviceMaster UP using the embedded *Server Configuration* web page. Use [Chapter 3. Configuring the DeviceMaster UP](#) on Page 55 to configure the DeviceMaster UP using the embedded web pages.

| Attribute ID | Name | Data Type | Data Value(s) | Access Rule |
|--------------|--|--------------|---|--|
| 1 | Socket Port Device | UDINT | 0=Raw/ASCII Data Device | Set/Get |
| 2 | Socket Port Commands | DWORD | 1=Reset Socket Port 2=Save in Flash 4=Clear Sequence Counters 8=Clear Statistics Counters | Set/Get (Get returns last command sent) |
| 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 | Reserved | UDINT | 0 | Get |
| 12 | PLC IP Address | UDINT | 00000000 Hex to FFFFFFFF Hex (Mask = 255.255.255.255) | Set/Get |
| 13 | Receive (DeviceMaster UP to PLC) Ethernet Data Transfer Method | USINT | 0=OFF 1=Unsolicited - Write-to-Tag 2=Unsolicited - Write-to-Tag-Synced 3=Polling (Default=3) | Set/Get |
| 14 | PLC Controller Slot Number | USINT | 0 to Max Slot Number on PLC (Default = 0) | Set/Get |

| Attribute ID | Name | Data Type | Data Value(s) | Access Rule |
|--------------|---|----------------------------|--|-------------|
| 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 | Set/Get |
| 16 | Maximum PLC Update Rate (No more than one message per time period.) | UINT (msec) | 10-65535 (Default = 40) | Set/Get |
| 17 | Reserved | UINT | 0 | Get |
| 18 | Received (DeviceMaster UP to PLC) Produced Data Tag Name | STRING (Array of 40 SINTS) | ASCII String | |
| 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 |
| 36 | Reserved | USINT[16] | 0 | Get |

| 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 |
| 10 Hex | No | Yes (Attribute 2 Only) | Set_Attribute_Single |

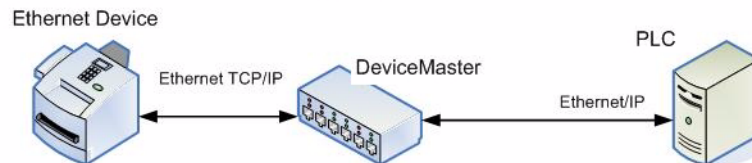
Attribute 1 - Socket Port Device Type: This attribute indicates the socket port device type. **Raw/ASCII Data Device** is the only currently supported option for EtherNet/IP.

Attribute 2 - Socket Port Commands: DeviceMaster UP 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 UP will detect 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 UP.
- **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 UP via an Ethernet TCP/IP socket.

Device Socket Interface



Attribute 4 - Device Listen Enable: Enabling this setting allows the device to connect to the DeviceMaster UP via an Ethernet TCP/IP socket.

- 0 = Disables listening - The DeviceMaster UP will not accept connection attempts.
- 1 = Enables listening - The DeviceMaster UP will accept connection attempts from the specified socket.

Attribute 5 - Device Listen Socket Port: This is the socket port number on the DeviceMaster UP the device will connect to if **Device Listen Enable** is enabled.

Attribute 6 - Device Connect Mode: This setting controls if and how the DeviceMaster UP attempts to connect to the device at the specified IP Address and socket.

- 0 = Never - The DeviceMaster UP will not attempt to connect to the device.
- 1 = Connect Always - Will attempt to connect to the device until a connection is made.
- 2 = Connect On Data - The DeviceMaster UP will 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 UP will attempt to connect to the device until a connection is made.

Attribute 7 - Device Disconnect Mode: This setting controls if and how the DeviceMaster UP disconnects from a socket device.

- 0 = Never - The DeviceMaster UP will not disconnect from the device.
- 1 = Disconnect On Idle - The DeviceMaster UP will disconnect 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 UP will connect 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 UP will connect 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 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 UP to PLC) Ethernet Data Transfer Method: This attribute specifies Ethernet data transfer method used by the DeviceMaster UP for the specified socket port. Please refer to Attribute 17 of the *Serial Port Configuration Object* 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 15 - Socket Port Data Transfer Options: This attribute specifies the socket port transfer options. The following options are supported:

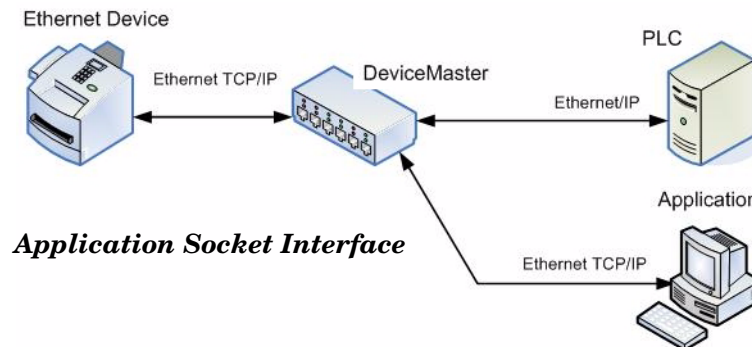
- **01 Hexadecimal (Bit 0)** = (PLC-5/SLC only) Receive Most Significant (MS) byte of 16 bit integer first. The default is transmit least significant (LS) byte first.
- **02 Hexadecimal (Bit 1)** = (PLC-5/SLC only) Transmit Most Significant (MS) byte of 16 bit integer first. The default is transmit least significant (LS) byte first.
- **04 Hexadecimal (Bit 2)** = Tx Sequence Number Checking
 - DeviceMaster UP 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 UP 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.

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 UP to space the messages to the PLC to prevent overrunning of data before the PLC can process it.

Attribute 17 - Receive (DeviceMaster UP 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 method.

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.



Attribute 20 - Application Listen Enable: Enabling this setting allows the application to connect to the DeviceMaster UP via an Ethernet TCP/IP socket.

- 0 = Disables listening - The DeviceMaster UP will not accept connection attempts.
- 1 = Enables listening - The DeviceMaster UP will accept connection attempts from the specified socket.

Attribute 21 - Application Connect Mode: This setting controls if and how the DeviceMaster UP attempts to connect to the application at the specified IP address and socket.

- 0 = Never - The DeviceMaster UP will not attempt to connect to the application.
- 1 = Connect Always - Will attempt to connect to the application until a connection is made.
- 2 = Connect On Data - The DeviceMaster UP will 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 UP will attempt to connect to the application until a connection is made.

Attribute 22 - Application Disconnect Mode: This setting controls if and how the DeviceMaster UP disconnects from an application.

- 0 = Never - The DeviceMaster UP will not disconnect from the application.
- 1 = Disconnect On Idle - The DeviceMaster UP will disconnect 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 UP the application will connect to if **Application Listen Enable** is enabled.

Attribute 24 - Application Connection Socket Port: This is the application socket port number the DeviceMaster UP will connect 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 UP will connect 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 23.

See the [DeviceMaster UP Filtering and Data Extraction Reference Guide](#) (Page 10) for further details.

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 23.

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 10) 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 UP 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 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 10) 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 10) for further details.

Attribute 33 - Filter Age Time: This attribute defines the time a filter string, RFID tag, or barcode will continue 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 will not be sent to the PLC and/or application. However, if the **Filter Age Time** has passed, it will pass 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 10) for further details.

Attribute 35 - Barcode Formats - 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.

The standard and eight digit formats are selected independently and each operates independently. It is important to note that the barcode filtering/data extraction will not function if no format is selected.

| Format | Numbering Digits | Company Digits | Product Digits | Check Digit |
|----------------------------|------------------|----------------|----------------|-------------|
| Standard Formats | | | | |
| None | N/A | N/A | N/A | N/A |
| Company-5/ Product-5 | 1-3 | 5 | 5 | 1 |
| Company-6/ Product-4 | 1-3 | 6 | 4 | 1 |
| Company-7/ Product-3 | 1-3 | 7 | 3 | 1 |
| Company-8/ Product-2 | 1-3 | 8 | 2 | 1 |
| Company-9/ Product-1 | 1-3 | 9 | 1 | 1 |
| Eight Digit Formats | | | | |
| EAN-8 Number-2/Product 5 | 2 | 0 | 5 | 1 |
| EAN-8 Number-3/Product 4 | 3 | 0 | 4 | 1 |
| UPC-E | 1 | Variable | Variable | 1 |

See the [DeviceMaster UP Filtering and Data Extraction Reference Guide](#) (Page 10) for further details.

2.3.5. 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 UP 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 UP. (Port numbers are numbered from one to N.)*

2.3.5.1. 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 UP | Get |
| 3 | Num Instances | UINT | Number of ports on the DeviceMaster UP | Get |

2.3.5.2. Instance Attributes

| Attribute ID | Name | Data Type | Data Value(s) | Access Rule |
|--------------|---|--------------------------------|---------------------------------------|-------------|
| 1 | Transmit (PLC to DeviceMaster UP) message data. Structure of: Produced data sequence number Data length Data array | UINT UINT Array of USINT | 0-65535 (FFFF hex) 1-440 0-255 | Set/Get |
| 2 | Receive (DeviceMaster UP to PLC) message data. Structure of: Produced data sequence number Data length Data array | UINT UINT Array of USINT | 0-65535 (FFFF hex) 0-440* 0-255 | Get |
| 3 | Receive (DeviceMaster UP to PLC) produced data sequence number - Normally sent to the PLC from the DeviceMaster UP in data transfer. | UINT | 0-65535 (FFFF hex) | Set/Get |
| 4 | Receive (DeviceMaster UP to PLC) consumed sequence number - normally updated by the PLC in the <i>Write-to-Tag-Synced</i> receive method. | UINT | 0-65535 (FFFF hex) | Set/Get |
| 5 | Transmit (PLC to DeviceMaster UP) produced data sequence number - normally sent to the DeviceMaster UP 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 *Write-to-Tag* and *Write-to-Tag-Synced* receive method is 2048 bytes.

2.3.5.3. 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 |

2.3.5.4. Instance Attribute Definitions

Attribute 1 - Transmit (PLC to DeviceMaster UP) Message Data: This attribute transmits data out of a socket port on the DeviceMaster UP.

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 18.)

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 UP to PLC) Message Data: This attribute provides the receive data while operating in the Polling communication method.

- DeviceMaster UP 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.
- Socket packets up to 2048 bytes may be received while operating in the *Write-To-Tag* or *Write-To-Tag-Synced* received methods. For socket packets over 440 bytes, the DeviceMaster UP will place the data into a sequence of tags. These tags must meet the following criteria:
 - All must be of type SINTs.
 - The entire sequence 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 will not be numbered (i.e. **Com1_RxData**), the second tag will have a 2 appended (i.e. **Com1_RxData2**), the third will have 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 socket packet will have been received and the PLC can process the data.

Attribute 3 - Receive (DeviceMaster UP 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 UP 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 19.

Attribute 5 - Transmit (PLC to DeviceMaster UP) 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 UP in the *Transmit Message data*.

2.3.6. 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.

2.3.6.1. Identity Object (01 Hex, 1 instance)

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

2.3.6.1.1. Class Attributes

| Attribute ID | Name | Data Type | Data Value(s) | Access Rule |
|--------------|----------|-----------|---------------|-------------|
| 1 | Revision | UINT | 1 | Get |

2.3.6.1.2. Instance Attributes

| Attribute ID | Name | Data Type | Data Value(s) | Access Rule |
|--------------|--|-----------------|--|-------------|
| 1 | Vendor ID | UINT | 909 (Comtrol) | Get |
| 2 | Device Type | UINT | 70 hex (vendor specific device) | Get |
| 3 | Product Code | UINT | 1-65535 | 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 2.3.6.1.3. Status Word , below. | Get |
| 6 | Serial Number | UDINT | 1-65535 | Get |
| 7 | Product Name Structure of: Name Length Name String | USINT STRING | Length of string: "DeviceMaster UP x P" (where " x " is 1 or 4 depending on the model) | Get |

2.3.6.1.3. Status Word

Please refer to Pages 5-8 in Volume 1 of the *CIP Common Specification*.

2.3.6.1.4. 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 |

2.3.6.2. 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.

2.3.6.2.1. Class Attributes

| Attribute ID | Name | Data Type | Data Value(s) | Access Rule |
|--------------|----------|-----------|---------------|-------------|
| 1 | Revision | UINT | 1 | Get |

2.3.6.2.2. 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 |

2.3.6.2.3. 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) |

2.3.6.3. Connection Manager Object (06 Hex)

The *Connection Manager Object* provides services for connection and connection-less communications. This object has no supported attributes.

2.3.6.3.1. Instance Object Specific Services

DeviceMaster UP supports the following instance object specific services:

| Service Code | Service Name | Service Description |
|--------------|------------------|---------------------------|
| 4E Hex | Forward_Close | Closes a connection. |
| 52 Hex | Unconnected_Send | Unconnected Send Service. |
| 54 Hex | Forward_Open | Opens a connection. |

2.3.6.4. Port Object (F4 Hex - 1 instance)

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

2.3.6.4.1. Class Attributes

| Attribute ID | Name | Data Type | Data Value(s) | Access Rule |
|--------------|---------------|---------------|---|-------------|
| 1 | Revision | UINT | 1 | Get |
| 2 | Max Instance | UINT | 1 | Get |
| 3 | Num Instances | UINT | 1 | Get |
| 8 | Entry Port | UINT | 1 | Get |
| 9 | All Ports | Array of UINT | [0]=0 [1]=0 [2]=TCP_IP_PORT_TYPE (4) [3]=TCP_IP_PORT_NUMBER(2) | Get |

2.3.6.4.2. Instance Attributes

| 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]=0212 Hex [1]=0 | Get |

2.3.6.4.3. 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 |

2.3.6.5. TCP Object (F5 Hex - 1 instance)

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

2.3.6.5.1. Class Attributes

| Attribute ID | Name | Data Type | Data Value(s) | Access Rule |
|--------------|---------------|-----------|---------------|-------------|
| 1 | Revision | UINT | 1 | Get |
| 2 | Max Instance | UINT | 1 | Get |
| 3 | Num Instances | UINT | 1 | Get |

2.3.6.5.2. Instance Attributes

| Attribute ID | Name | Data Type | Data Value(s) | Access Rule |
|--------------|---|---|--|-------------|
| 1 | Status | DWORD | 0=Not configured 1=Valid configuration | 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 |
| 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> | Get |
| 6 | Host Name Structure of: Host Name Length Host Name | UINT STRING | 7 to 15 <IP address> | Get |

2.3.6.5.3. 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 |

2.3.6.6. Ethernet Link Object (F6 Hex)

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

2.3.6.6.1. Class Attributes

| Attribute ID | Name | Data Type | Data Value(s) | Access Rule |
|--------------|---------------|-----------|---------------|-------------|
| 1 | Revision | UINT | 2 | Get |
| 2 | Max Instance | UINT | 1 | Get |
| 3 | Num Instances | UINT | 1 | Get |

2.3.6.6.2. 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) | Get |
| 3 | Physical Address | Array of 6 USINT | MAC address | Get |

2.3.6.6.3. 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 |

2.3.6.7. 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.

2.3.6.7.1. Class Attributes

Not supported.

2.3.6.7.2. Instance Attributes

Not supported.

2.3.6.7.3. Instances

Supports Instance 1.

2.3.6.7.4. Common Services

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

2.3.6.7.5. 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 |

2.4. PLC-5/SLC Interface

EtherNet/IP firmware versions 2.01 or later supports EtherNet/IP connections to the PLC-5 and SLC PLCs. As of EtherNet/IP version 3.02, 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 UP only through the embedded web page. There is no DeviceMaster UP configuration available through the PLC-5 and SLC PLCs. See the following topics for more information:
 - [4.4. SLC or MicroLogix PLC Programming Example Instructions](#) on Page 94
 - [4.5. PLC-5 PLC Programming Example Instructions](#) on Page 114
- 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 1024 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 UP to operate in *Write-to-File* or *Write-to-File-Synced*, enter the file name starting with a "\$" (i.e. \$N10:0).
- 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, Control recommends the Write-to-File receive method used with the Maximum PLC Update Rate option*

2.4.1. 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 UP. If you need to update your PLC firmware, contact your Rockwell distributor.

2.4.1.1. SLC 5/05

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

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

2.4.1.2. 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, 1785-L30B, 1785-L40B, 1785-L40L, 1785-L60B, 1785-L60L, 1785-L80B | Series B: Revision N.1 or later Series C: Revision N or later Series D: Revision E or later Series E: Revision D or later |
| ControlNet PLC-5 attached to EtherNet Module | 1785-L30C15 1785-L40C15 1785-L60C15 1785-L80C15 | Series C: Revision N or later Series D: Revision E or later Series E: Revision D or later 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

2.4.2. 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) |

2.4.2.1. DeviceMaster UP File Addressing

The following tables display the DeviceMaster UP 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 |

Note: There are no statistics available for the Ethernet socket ports.

2.4.3. 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 UP to PLC) message data. Structure of: Produced data sequence number Data length (in bytes) Data array | UINT UINT Array of UINT | 0-65535 (FFFF hex) 0-222 (SLC) 0-248 (PLC-5)* 0-65535 | Read only |

* *Polling mode only.*

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 [3.3.1.2. Serial Packet Identification](#) on Page 59.
- The DeviceMaster UP supports serial packets of up 1024 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 will automatically be 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 will have been transferred to the PLC when the sequence number is updated.

2.4.4. 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 UP) message data. Structure of: Produced data sequence number Data length (in bytes) Data array | UINT UINT Array of UINT | 0-65535 (FFFF hex) 1-202 (SLC) 1-222 (PLC-5) 0-65535 | Read/Write |

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 [3.3.1.2. Serial Packet Identification](#) on Page 59.
- A **Get** returns the last successfully transmitted serial/socket packet.

2.4.5. 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 UP 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 UP 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.

2.4.6. 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: <ul style="list-style-type: none"> • No STX byte(s) found • No ETX byte(s) found • Time-outs • Too large of packet • Receive buffer queue overflows |
| 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. |

| Retrieve Statistics Message | Description |
|--|--|
| Received Consumed Sequence Error Count | <p>This attribute counts the number of received consumed sequence number errors. The DeviceMaster UP only increments this number when all of the following statements are true:</p> <ul style="list-style-type: none"> • You selected the <i>Unsolicited - Write-to-Tag-Synced</i> method of receive communication. • DeviceMaster UP 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 | <p>This attribute counts the number of Duplicate Transmit Sequence Number errors. The DeviceMaster UP increments this number when the following statements are true:</p> <ul style="list-style-type: none"> • You enabled the Transmit Sequence Number Checking configuration option. See <i>Transmit Sequence Number Checking</i> under 3.6.1. EtherNet/IP Settings on Page 63 for additional information. • DeviceMaster UP receives a transmit message with a sequence number equal to the previous sequence number. (The DeviceMaster UP expects this sequence number to be incremented by one from the sequence number in the previous transmit message.) |
| Unexpected Transmit Sequence Number Error Count | <p>This attribute counts the number of Unexpected Transmit Sequence Number errors. The DeviceMaster UP increments this number when the following statements are true.</p> <ul style="list-style-type: none"> • You enabled the <i>Transmit Sequence Number Checking</i> configuration option. See Transmit Sequence Number Checking under 3.6.1. EtherNet/IP Settings. • DeviceMaster UP 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 UP increments this sequence number by one with each new transmit message.) |
| Dropped Packet to Application Count | <p>This attribute counts the number of dropped receive packets on the serial port intended for the application due to:</p> <ul style="list-style-type: none"> • No STX byte(s) found • No ETX byte(s) found • Time-outs • Too large of packet • Receive buffer queue overflows |

2.4.7. Receive Communication Methods

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

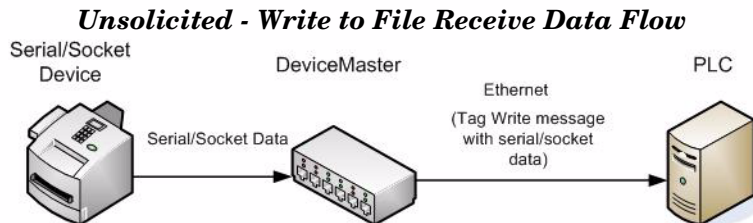
2.4.7.1. Unsolicited - Write to File Receive Method

Note: This is the recommended receive method.

When a serial/socket packet is received on the DeviceMaster UP, the data packet is immediately written to a file data location on the PLC. The following diagram shows the 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 [2.4.3. Receive Data Message](#) on Page 48 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 will allow your PLC to process the data. The default of 40 milliseconds may or may not need to be increased for your PLC application.



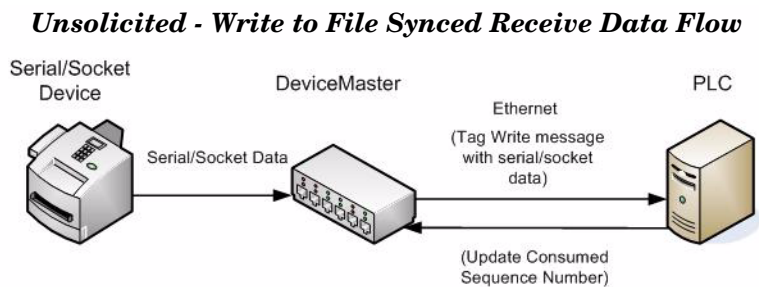
2.4.7.2. Unsolicited - Write to File Synced Receive Method

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.

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 [2.4.3. Receive Data Message](#) on Page 48 for more information.
- New data is indicated with an incremented sequence number.
- New data will not be 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 UP 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 UP 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.)



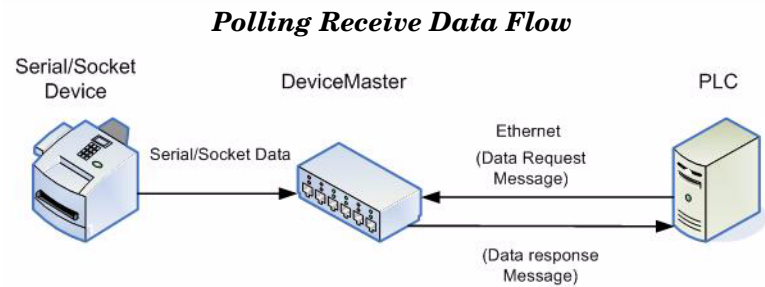
2.4.7.3. Polling Receive Method

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

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 [2.4.3. Receive Data Message](#) on Page 48 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 will also return the same sequence number.
- No data is indicated with a length of zero.
- While the DeviceMaster UP 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 UP 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 3. Configuring the DeviceMaster UP

This chapter discusses the embedded web pages for serial and Ethernet device configuration. Ethernet devices are configured via an Ethernet TCP/IP socket connection. 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 [DeviceMaster UP Hardware Installation Guide](#) or the PortVision Plus help system.

You may want to refer to the *EtherNet/IP Installation Quick Start* (Page 9) and use this chapter as a reference if you need information about specific fields. The *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 (2.3.1. Serial Port Configuration Object Definition (70 Hex) on Page 13) or Socket Port Configuration (2.3.4. Socket Port Configuration Object Definition (73 Hex) on Page 29) objects.*

3.1. Port Configuration Overview

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

If you have not configured the network information into the DeviceMaster UP during initial setup, you must configure the network information before configuring serial/socket port characteristics. See the [DeviceMaster UP Hardware Installation Guide](#) or the PortVision Plus help system for help configuring the network settings.

1. From PortVision Plus, highlight the DeviceMaster UP that you want to configure and select **Web Manager**.

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

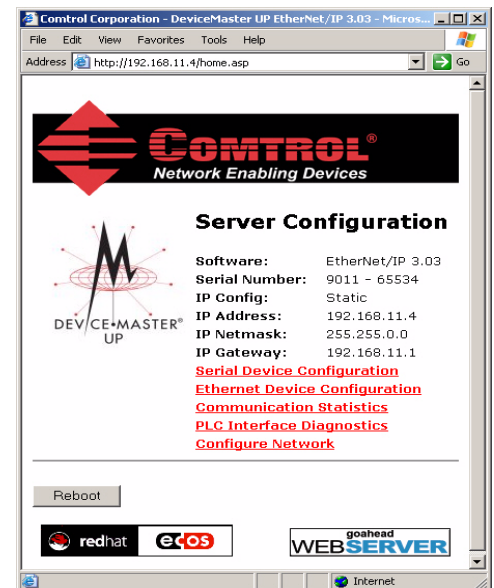
2. Select the appropriate procedure for your environment.

Serial Device

- a. Select **Serial Device Configuration**.
- b. Select the appropriate port to access the *Edit Serial Port Configuration* page for that port.
- c. Change the [serial port configuration properties](#) (Page 57) as required for your site.

Ethernet Device

- a. Select **Ethernet Device Configuration**.
 - b. Select the appropriate socket to access the *Edit Socket Port Configuration* page for that port.
 - c. Change the [socket port configuration properties](#) (Page 61) as required for your site.
3. Select **Submit** to commit the changes and repeat for each port.
 4. Use the appropriate procedure for your environment to complete the DeviceMaster UP installation.
 - **ControlLogix PLC:** [4.3. ControlLogix PLC Programming Example Instructions](#) on Page 81 describes how to use RSLogix 5000 to configure and run the DeviceMaster UP.
 - **SLC or MicroLogix PLC:** [4.4. SLC or MicroLogix PLC Programming Example Instructions](#) on Page 94 describes how to use RSLogix 500 to configure and run the DeviceMaster UP.



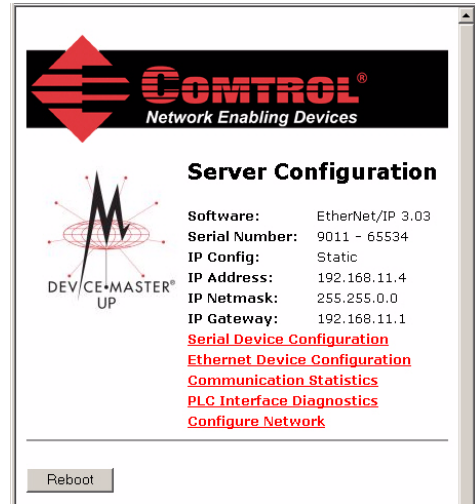
- **PLC-5 PLC:** [4.5. PLC-5 PLC Programming Example Instructions](#) on Page 114 describes how to use RSLogix 5 to configure and run the DeviceMaster UP.

3.2. Embedded Web Pages Overview

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

Access the main DeviceMaster UP web page (*Server Configuration*) from PortVision Plus or enter the IP address of the DeviceMaster UP in the **Address** box of your web browser.

The *Server Configuration* page displays the software version and current network configuration for the DeviceMaster UP. In addition, the *Server Configuration* page links to the configuration, statistics, and diagnostics pages, which are discussed in the table below.



| Server Configuration Page | |
|---|---|
| Software | EtherNet/IP firmware version currently running on the DeviceMaster UP. |
| Serial Number | DeviceMaster UP serial number. |
| IP Config | Type of IP configuration currently in use (static or DHCP). |
| IP Address, IP Netmask, and IP Gateway | IP address, netmask, and gateway configured in the DeviceMaster UP. |
| Serial Device Configuration | Opens the <i>Serial Device Configuration</i> page (3.3. Serial Device Configuration Page on Page 57), which provides an overview of the serial device interface settings and access to the <i>Edit Serial Port Configuration</i> page for serial port configuration on the selected port. |
| Ethernet Device Configuration | Opens the <i>Ethernet Device Configuration</i> page (3.4. Ethernet Device Configuration Page on Page 61), which provides an overview of the Ethernet device interface settings and access to the <i>Edit Socket Port Configuration</i> page for Ethernet device configuration on the selected socket port. |
| Communication Statistics | Opens the <i>Communication Statistics</i> page (3.7. Serial / Ethernet Device Communication Statistics on Page 70), which is an overview of serial and Ethernet device interface statistics. |
| PLC Interface Diagnostics | Opens the <i>PLC Interface Diagnostics</i> page (3.8. PLC Interface Diagnostics on Page 73), which is an overview of statistics and error reporting for the EtherNet/IP PLC interface. |
| Configure Network | Opens the <i>Configure Network</i> page (3.9. Edit Network Configuration Page on Page 75), which can be used to modify DeviceMaster UP network configuration after initial configuration using PortVision Plus. |
| Reboot | Reboots the DeviceMaster UP. |

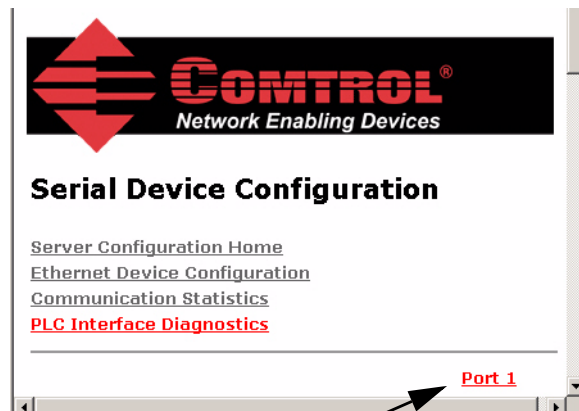
3.3. Serial Device Configuration Page

The *Serial Device Configuration* page provides:

- Links to other pages
- Access to the *Edit Serial Port Configuration* page for each port (Port #)
- An overview of serial device configuration settings

The overview area for each port displays the current settings.

To change these settings for a port, select the corresponding **Port #** link, which opens the *Edit Serial Port Configuration* page. See [3.3.1. Edit Serial Port Configuration Page](#) on Page 57 to locate information for each setting area.



3.3.1. Edit Serial Port Configuration Page

Use the *Edit Serial Port Configuration* page to change a serial port’s configuration parameters.

To access the *Edit Serial Port Configuration* page, select the appropriate port number link (for example, **Port 1**) on the *Serial Device Configuration* page.

The next two subsections discuss the *Serial Port* and *Serial Packet ID Settings* areas on this page. The remainder of the page is discussed in the following subsections, which are located under the 3.6. *Common Configuration Areas (Serial or Ethernet Device)* section:

- [3.6.1. EtherNet/IP Settings](#) on Page 63
- [3.6.2. Filtering/Data Extraction Configuration](#) on Page 65
- [3.6.3. Application TCP Connection Configuration](#) on Page 68
- [3.6.4. Saving Port Options](#) on Page 69

3.3.1.1. Serial Configuration

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

| | |
|-----------------------------|---------------------------------------|
| Serial Configuration | |
| Mode: | <input type="text" value="RS-232"/> |
| Baud: | <input type="text" value="9600"/> |
| Parity: | <input type="text" value="none"/> |
| Data Bits: | <input type="text" value="8"/> |
| Stop Bits: | <input type="text" value="1"/> |
| Flow: | <input type="text" value="none"/> |
| DTR: | <input type="text" value="off"/> |
| Rx Timeout Between Packets: | <input type="text" value="200"/> (ms) |

| Serial Configuration | |
|----------------------|---|
| 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 | Select a baud rate from the list. The baud rate that you select determines how fast information is transferred through a port. |

| Serial Configuration | |
|-----------------------------------|--|
| Parity | <p>Select a method for error checking.</p> <ul style="list-style-type: none"> • None - When the parity is set to none, there is no parity bit, and DeviceMaster UP 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 | <p>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:</p> <ul style="list-style-type: none"> • 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 | <p>Select the state of Data Terminal Ready (DTR).</p> <ul style="list-style-type: none"> • on - Enables DTR. • off - Disables DTR. • WhenEnabled - Select this option when enabling the serial port through the PLC. |
| Rx Timeout Between Packets | <p>Specifies the following information:</p> <ul style="list-style-type: none"> • How long the DeviceMaster UP 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. |

3.3.1.2. Serial Packet Identification

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

| | |
|--|---|
| Serial Packet Identification | |
| STX (Start of Transmission) Rx Detect: | <input type="text" value="one byte"/> Byte 1: <input type="text" value="2"/> Byte 2: <input type="text"/> |
| ETX (End of Transmission) Rx Detect: | <input type="text" value="one byte"/> Byte 1: <input type="text" value="3"/> Byte 2: <input type="text"/> |
| Discard Rx Packets With Errors: | <input checked="" type="checkbox"/> |
| PLC Specific Settings | |
| STX (Start of Transmission) Tx Append: | <input type="text" value="none"/> Byte 1: <input type="text"/> Byte 2: <input type="text"/> |
| ETX (End of Transmission) Tx Append: | <input type="text" value="none"/> Byte 1: <input type="text"/> Byte 2: <input type="text"/> |
| Strip Rx STX/ETX: | <input checked="" type="checkbox"/> |
| Application Specific Settings | |
| STX (Start of Transmission) Tx Append: | <input type="text" value="none"/> Byte 1: <input type="text"/> Byte 2: <input type="text"/> |
| ETX (End of Transmission) Tx Append: | <input type="text" value="none"/> Byte 1: <input type="text"/> Byte 2: <input type="text"/> |
| Strip Rx STX/ETX: | <input type="checkbox"/> |

For more information on serial packet ID settings, see [2.3.1. Serial Port Configuration Object Definition \(70 Hex\)](#) on Page 13.

| Serial Packet Identification | |
|--|---|
| STX (Start of Transmission) Rx Detect | <p>When enabled, the DeviceMaster UP detects an STX (start of transmission) byte sequence which is configured as one byte or two bytes when it receives a serial packet. The length indicates the number of STX bytes, valid values for length are:</p> <ul style="list-style-type: none"> none - Disables this function and the DeviceMaster UP accepts the first byte received after the last ETX byte(s) as the start of the next data packet. one byte - Scans serial data for one STX byte and when the DeviceMaster UP finds an STX byte it collects the data. If the first byte is not the STX byte, it discards the byte. The DeviceMaster UP continues to discard the bytes until it finds an STX byte. two bytes - Scans serial data for two STX bytes and when the DeviceMaster UP finds two STX bytes it collects the data. If the STX bytes cannot be found, it discards the bytes. The DeviceMaster UP continues to discard the bytes until it finds the two STX bytes. <p>Byte 1 - Specifies the character that represents the first STX byte. The DeviceMaster UP 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.</p> <p>Byte 2 - Specifies the character that represents the second STX byte. The DeviceMaster UP 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.</p> |

| Serial Packet Identification | |
|---|---|
| ETX (End of Transmission) Rx Detect | <p>When enabled, the DeviceMaster UP 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:</p> <ul style="list-style-type: none"> • none - Disables this function and the DeviceMaster UP uses the <i>Rx Timeout Between Packets</i> to indicate the end of data packet. • one byte - Scans serial data for one ETX byte and when the DeviceMaster UP 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 UP finds the ETX bytes, it identifies the data as a serial packet. <p>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.</p> <p>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.</p> |
| Discard Rx Pkts with Errors | <p>By default, this box is checked and the DeviceMaster UP discards serial packets with errors. Clear the check box when you need to receive a serial packet with errors to troubleshoot an issue.</p> |
| PLC Specific Settings and Application Specific Settings STX Tx Append | <p>When enabled, the DeviceMaster UP 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:</p> <ul style="list-style-type: none"> • none - Disables this function. • one byte - Inserts one STX byte before the data. • two bytes - Inserts two STX bytes before the data. <p>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.</p> <p>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.</p> |
| PLC Specific Settings and Application Specific Settings ETX Tx Append | <p>When enabled, the DeviceMaster UP 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:</p> <ul style="list-style-type: none"> • 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. <p>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.</p> <p>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.</p> |
| PLC Specific Settings and Application Specific Settings Strip Rx STX/ETX | <p>When you select this check box, the DeviceMaster UP strips STX/ETX characters from received serial packets. Clear the check box when you do not want the DeviceMaster UP to strip STX/ETX characters from received serial packets.</p> <p>Serial Packets sent from the PLC or application to the DeviceMaster UP (over Ethernet), and then sent out the serial port, are not checked for STX/ETX.</p> <p>No STX/ETX character stripping occurs in these serial packets, and framing/parity/overrun error checking does not apply.</p> |

3.4. Ethernet Device Configuration Page

The *Ethernet Device Configuration* page provides:

- Links to other pages
- Access to the *Edit Socket Port Configuration* page for each port (Port #)
- An overview of Ethernet device configuration settings

The overview area for each port displays the current settings.

To change these settings for a port, select the corresponding **Port #** link, which opens the *Edit Socket Port Configuration* page. See [3.3.1. Edit Serial Port Configuration Page](#) on Page 57 to locate information for each setting area.



Ethernet Device Configuration

- [Server Configuration Home](#)
- [Serial Device Configuration](#)
- [Communication Statistics](#)
- [PLC Interface Diagnostics](#)

Select the appropriate socket port number to configure the socket port characteristics. → [Socket 1](#)

3.5. Edit Socket Port Configuration Page

Access to the *Edit Socket Port Configuration* page is provided by selecting the corresponding socket number on the *Ethernet Device Configuration* page (for example, **Socket 1**).

The the remainder of this subsection discusses the *Device TCP Connection Configuration* area on this page.

The other areas of this page are discussed in the following subsections, which are located under the *3.6. Common Configuration Areas (Serial or Ethernet Device)* section:

- [3.6.1. EtherNet/IP Settings](#) on Page 63
- [3.6.2. Filtering/Data Extraction Configuration](#) on Page 65
- [3.6.3. Application TCP Connection Configuration](#) on Page 68
- [3.6.4. Saving Port Options](#) on Page 69

Device TCP Connection Configuration

| | |
|---------------------|---------------------------------------|
| Enable: | <input type="checkbox"/> |
| Listen: | <input type="checkbox"/> |
| Listen Port: | <input type="text" value="8000"/> |
| Connect To Mode: | <input type="text" value="Never"/> |
| Connect Port: | <input type="text" value="8010"/> |
| Connect IP Address: | <input type="text" value="0.0.0.0"/> |
| Disconnect Mode: | <input type="text" value="Never"/> |
| Idle Timer: | <input type="text" value="0"/> (msec) |

The following table provides information about configuring the *Device TCP Connection Configuration* area.

| Device TCP Connection Configuration | |
|-------------------------------------|---|
| Enable | <p>This setting enables/disables the <i>Device Ethernet Device</i>. 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.</p> <div style="text-align: center;"> <pre> graph LR ED[Ethernet Device] <--> Ethernet TCP/IP DS[DeviceSocket Interface / DeviceMaster] DS <--> Ethernet/IP PLC[PLC] </pre> </div> |

| Device TCP Connection Configuration | |
|--|--|
| Listen | <p>Enabling this setting allows the device to connect to the DeviceMaster UP via an Ethernet TCP/IP socket.</p> <ul style="list-style-type: none"> • Not selected - Disables listening; the DeviceMaster UP will not accept connection attempts. • Selected - Enables listening; the DeviceMaster UP will accept connection attempts from the specified Listen Port. |
| Listen Port | <p>This is the socket port number on the DeviceMaster UP the application will connect to if the Device Listen Enable is selected.</p> |
| Connect To Mode | <p>This setting specifies if and how the DeviceMaster UP attempts to connect to the device at the specified Connect IP Address and Connect Port.</p> <ul style="list-style-type: none"> • Never - The DeviceMaster UP will not attempt to connect to the device. • Connect-Always - The DeviceMaster UP will attempt to connect to the device until a connection is made. • Connect-On-Data - The DeviceMaster UP will 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 UP will attempt to connect to the device until a connection is made. |
| Connect Port | <p>The device socket port number the DeviceMaster UP will connect to if the Device Connect To Mode is set to either Connect-Always or Connect-On-Data.</p> |
| Connect IP Address | <p>The device IP address the DeviceMaster UP will connect to if the Device Connect To Mode is set to either Connect-Always or Connect-On-Data.</p> |
| Disconnect Mode | <p>This setting specifies if and how the DeviceMaster UP disconnects from the device.</p> <ul style="list-style-type: none"> • Never - The DeviceMaster UP will not disconnect from the device. • Idle - The DeviceMaster UP will disconnect when there has been no transmit or received data between the device and PLC/application for a specified Idle Timer period. |
| Idle Timer | <p>The idle timeout period in milliseconds that is used if the Device Disconnect Mode is set to Idle.</p> |

3.6. Common Configuration Areas (Serial or Ethernet Device)

The *Edit Serial Port Configuration* and *Edit Socket Port Configuration* pages have the following areas in common. This section discusses the following:

- [3.6.1. EtherNet/IP Settings](#) on Page 63
- [3.6.2. Filtering/Data Extraction Configuration](#) on Page 65
- [3.6.3. Application TCP Connection Configuration](#) on Page 68

3.6.1. EtherNet/IP Settings

Use this area to set up the EtherNet/IP settings for a serial or socket port using the appropriate *Edit Port Configuration* page.

For more information on EtherNet/IP settings, see [2.3.1. Serial Port Configuration Object Definition \(70 Hex\)](#) on Page 13.

EtherNet/IP Settings

| | |
|---|--|
| Rx (To PLC) Ethernet Transfer Method: | <input type="text" value="Polling"/> |
| PLC IP Address: | <input type="text" value="0.0.0.0"/> |
| PLC Controller Slot Number (ControlLogix Family): | <input type="text" value="0"/> |
| Maximum PLC Update Rate (Write-To-Tag/File): | <input type="text" value="40"/> (msec) |
| Rx (To PLC) Produced Data Tag/File Name: | <input type="text"/> |
| Note: File names for SLC/PLC-5 must begin with a "\$" (i.e. \$N10:0). | |
| Tx Sequence Number Checking: | <input type="checkbox"/> |
| Disable Non-Filtered To PLC Rx Queue: | <input type="checkbox"/> |
| (PLC-5/SLC) Rx MS Byte First: | <input type="checkbox"/> |
| (PLC-5/SLC) Tx MS Byte First: | <input type="checkbox"/> |

| EtherNet/IP Settings (Serial or Socket Port) | |
|--|---|
| Rx (To PLC) Ethernet Transfer Method | <p>Specifies the EtherNet data transfer method used by the DeviceMaster UP. There are three methods that the DeviceMaster UP can use to transfer data received from a serial or Ethernet device to the PLC. These methods are:</p> <ul style="list-style-type: none"> • <i>Write-to-Tag/File</i> - The DeviceMaster UP writes the serial data directly into a tag or file on the PLC. However, it should only be used if the PLC can scan and consume the serial data faster than the serial device can produce it. • <i>Write-to-Tag/File-Synced</i> - The DeviceMaster UP writes the serial data into a tag or file on the PLC and provides a mechanism to synchronize the data flow between the PLC and the DeviceMaster UP. Use this method when you want to ensure that the tag or file is not overwritten before the PLC can consume the data. • <i>Polling</i> - 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 port receive queues on the DeviceMaster UP do not overflow. |
| PLC IP Address | <p>Specifies the IP address for the PLC EtherNet/IP card.</p> <p>Note: <i>The Polling method does not use this attribute.</i></p> |
| PLC Controller Slot Number (ControlLogix Family) | <p>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.</p> <p>Note: <i>The Polling method does not use this attribute.</i></p> |
| Maximum PLC Update Rate (Write-To-Tag/File) | <p>The maximum rate (or minimum time interval) in milliseconds, that messages is sent to the PLC tag in the <i>Write-To-Tag/File</i> receive method. This setting configures the DeviceMaster UP to space the messages to the PLC in order to prevent overrunning of data before the PLC can process it.</p> |

| EtherNet/IP Settings (Serial or Socket Port) | |
|---|--|
| Rx (To PLC) Produced Data Tag/ File Name | <p>Specifies the PLC tag or file name. It indicates where to write received data while operating in the <i>Write-to-Tag/File</i> or <i>Write-to-Tag/File-Synced</i> receive method. This column supports a name containing up to 40 characters.</p> <p>Note: <i>The Polling method does not use this attribute.</i></p> <p>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).</p> |
| Tx Sequence Number Checking | <p>When you select this check box, the DeviceMaster UP checks the transmit sequence number and performs the following tasks:</p> <ul style="list-style-type: none"> • 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 <i>Duplicate Transmit Sequence Error Count</i>. • 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 <i>Unexpected Transmit Sequence Error Count</i>. <p>This check box is clear by default. Clear the check box when you do not want DeviceMaster UP to check the transmit sequence number.</p> |
| Disable Non-Filtered TO PLC Rx Queue | <p>If filtering is disabled, only the last message received is sent to the PLC.</p> <p>This box is clear by default.</p> |
| (PLC-5/SLC) Rx MS Byte First | <p>When you select this check box, the DeviceMaster UP 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.</p> |
| (PLC-5/SLC) Tx MS Byte First | <p>When you select this check box, DeviceMaster UP 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.</p> |

3.6.2. Filtering/Data Extraction Configuration

Use this area to configure filtering or data extraction settings for a serial or socket port using the appropriate *Edit Port Configuration* page.

Filtering/Data Extraction Configuration

To PLC Filter Mode:

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:

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:

RFID Reader Interface Type:

Barcode UPC/EAN Standard 12-14 Digit Format:

Barcode UPC/EAN Eight Digit Format:

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

Discard Unrecognized Data (RFID/Barcode):

| Filtering/Data Extraction Configuration (Serial or Socket Port) | |
|---|--|
| To PLC Filter Mode | <p>Defines the filter/data extraction mode to be employed on data to be sent to the PLC.</p> <ul style="list-style-type: none"> 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 23. |
| To PLC Filtering Options (RFID Only) | <p>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.</p> <ul style="list-style-type: none"> 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. |
| To PLC Filtering Options (RFID/Barcode) | <p>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.</p> <ul style="list-style-type: none"> 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. |

| Filtering/Data Extraction Configuration (Serial or Socket Port) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|------|---|---|---|---|----------------|-----|-----|-----|------|------------------|-------|-------|-------|------|-----------------|---------|---------|------------|------|----------------|-----|---|---|-----|------------------|-------|---|---|-----|
| To Application Filter Mode | <p>The filter/data extraction mode to be employed on data to be sent to the application.</p> <ul style="list-style-type: none"> • 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 23. <p><i>Note: The application filter mode can be set independently of the PLC filtering mode. The only exceptions are:</i></p> <ul style="list-style-type: none"> • <i>If the PLC filter mode is set to RFID, the application filter mode cannot be set to Barcode.</i> • <i>If the PLC filter mode is set to Barcode, the application filter mode cannot be set to RFID.</i> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| To Application Filtering Options (RFID Only) | <p>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.</p> <ul style="list-style-type: none"> • 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) | <p>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.</p> <ul style="list-style-type: none"> • 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. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RFID Antenna Grouping | <p>This setting is applicable only to RFID filtering and only if the antenna filtering option is enabled. It allows the DeviceMaster UP to filter RFID tags based on antenna groupings. The possible groupings are:</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Setting</th> <th style="text-align: center;">Group 1 Antennas</th> <th style="text-align: center;">Group 2 Antennas</th> <th style="text-align: center;">Group 3 Antennas</th> <th style="text-align: center;">Group N Antennas</th> </tr> </thead> <tbody> <tr> <td>None</td> <td style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> <td style="text-align: center;">4</td> </tr> <tr> <td>Groups of Twos</td> <td style="text-align: center;">1,2</td> <td style="text-align: center;">3,4</td> <td style="text-align: center;">5,6</td> <td style="text-align: center;">Etc.</td> </tr> <tr> <td>Groups of Threes</td> <td style="text-align: center;">1,2,3</td> <td style="text-align: center;">4,5,6</td> <td style="text-align: center;">7,8,9</td> <td style="text-align: center;">Etc.</td> </tr> <tr> <td>Groups of Fours</td> <td style="text-align: center;">1,2,3,4</td> <td style="text-align: center;">5,6,7,8</td> <td style="text-align: center;">9,10,11,12</td> <td style="text-align: center;">Etc.</td> </tr> <tr> <td>First Two Only</td> <td style="text-align: center;">1,2</td> <td style="text-align: center;">3</td> <td style="text-align: center;">4</td> <td style="text-align: center;">N+1</td> </tr> <tr> <td>First Three Only</td> <td style="text-align: center;">1,2,3</td> <td style="text-align: center;">4</td> <td style="text-align: center;">5</td> <td style="text-align: center;">N+2</td> </tr> </tbody> </table> | 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 |
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Filtering/Data Extraction Configuration (Serial or Socket Port) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|-----------------------|-------------------------|-----------------------|-----------------------|--------------------|-------------------------|--|--|--|--|------|-----|-----|-----|-----|------------------------------|----------------------|-----|---|---|---------------------|----------------------|-----|---|---|--|----------------------|-----|---|---|--|----------------------|-----|---|---|--|----------------------|-----|---|---|----------------------------|--|--|--|--|--|--------------------------|---|---|---|--|--------------------------|---|---|---|--|-------|---|----------|----------|
| RFID Reader Interface Type | <p>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.</p> <ul style="list-style-type: none"> • Unspecified - The DeviceMaster UP will assume a HEX ASCII format and will attempt to locate the antenna number. • Alien (Text Mode) - Specifies the Alien RFID reader Text Mode. • Alien (Terse Mode) - Specifies the Alien RFID reader Terse Mode. • Intermec (Hex ASCII Mode) - Specifies the Intermec reader returning data in the Hex ASCII Mode. <p>See the DeviceMaster UP Filtering and Data Extraction Reference Guide (Page 10) for further details.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Barcode Formats: | <p>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. It is important to note that the barcode filtering/data extraction will not function if no format is selected.</p> <table border="1"> <thead> <tr> <th>Format</th> <th>Numbering Digits</th> <th>Company Digits</th> <th>Product Digits</th> <th>Check Digit</th> </tr> </thead> <tbody> <tr> <td colspan="5">Standard Formats</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>UPC/EAN Standard 12-14 Digit</td> <td>Company-5/ Product-5</td> <td>1-3</td> <td>5</td> <td>1</td> </tr> <tr> <td>UPD/EAN Eight Digit</td> <td>Company-6/ Product-4</td> <td>1-3</td> <td>6</td> <td>1</td> </tr> <tr> <td></td> <td>Company-7/ Product-3</td> <td>1-3</td> <td>7</td> <td>1</td> </tr> <tr> <td></td> <td>Company-8/ Product-2</td> <td>1-3</td> <td>8</td> <td>1</td> </tr> <tr> <td></td> <td>Company-9/ Product-1</td> <td>1-3</td> <td>9</td> <td>1</td> </tr> <tr> <td colspan="5">Eight Digit Formats</td> </tr> <tr> <td></td> <td>EAN-8 Number-2/Product 5</td> <td>2</td> <td>0</td> <td>5</td> </tr> <tr> <td></td> <td>EAN-8 Number-3/Product 4</td> <td>3</td> <td>0</td> <td>4</td> </tr> <tr> <td></td> <td>UPC-E</td> <td>1</td> <td>Variable</td> <td>Variable</td> </tr> </tbody> </table> <p>See the DeviceMaster UP Filtering and Data Extraction Reference Guide (Page 10) for further details.</p> | Format | Numbering Digits | Company Digits | Product Digits | Check Digit | Standard Formats | | | | | None | N/A | N/A | N/A | N/A | UPC/EAN Standard 12-14 Digit | Company-5/ Product-5 | 1-3 | 5 | 1 | UPD/EAN Eight Digit | Company-6/ Product-4 | 1-3 | 6 | 1 | | Company-7/ Product-3 | 1-3 | 7 | 1 | | Company-8/ Product-2 | 1-3 | 8 | 1 | | Company-9/ Product-1 | 1-3 | 9 | 1 | Eight Digit Formats | | | | | | EAN-8 Number-2/Product 5 | 2 | 0 | 5 | | EAN-8 Number-3/Product 4 | 3 | 0 | 4 | | UPC-E | 1 | Variable | Variable |
| Format | Numbering Digits | Company Digits | Product Digits | Check Digit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Standard Formats | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| None | N/A | N/A | N/A | N/A | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UPC/EAN Standard 12-14 Digit | Company-5/ Product-5 | 1-3 | 5 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UPD/EAN Eight Digit | Company-6/ Product-4 | 1-3 | 6 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Company-7/ Product-3 | 1-3 | 7 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Company-8/ Product-2 | 1-3 | 8 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Company-9/ Product-1 | 1-3 | 9 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Eight Digit Formats | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | EAN-8 Number-2/Product 5 | 2 | 0 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | EAN-8 Number-3/Product 4 | 3 | 0 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | UPC-E | 1 | Variable | Variable | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Filter Age Time (Time filtered after last read) | <p>Defines the time a filter string, RFID tag, or barcode will continue to be filtered after the last time it was received.</p> <p>If an entry is received before the Filter Age Time has passed, the entry is filtered and the data will not be sent to the PLC and/or application. However, if the Filter Age Time has passed, it will pass filtering and be sent to the PLC and/or application.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Discard Unrecognized Data Mode (RFID/Barcode) | <p>Specifies what to do with unrecognized RFID or barcode data.</p> <ul style="list-style-type: none"> • 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. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

3.6.3. Application TCP Connection Configuration

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

Application TCP Connection Configuration

Enable:

Listen:

Listen Port:

Connect To Mode:

Connect Port:

Connect IP Address:

Disconnect Mode:

Idle Timeout: (msec)

| Application TCP Connection Configuration (Serial or Socket Port) | |
|--|--|
| Enable | <p>Enables/disables the <i>Application Socket Interface</i>. Enabling this function allows an application to be connected to the device serial/socket port.</p> <p>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.</p> |
| Listen | <p>Enabling this setting allows the application to connect to the DeviceMaster UP via an Ethernet TCP/IP socket.</p> <ul style="list-style-type: none"> • Not selected - Disables listening and the DeviceMaster UP will not accept connection attempts. • Selected - Enables listening and the DeviceMaster UP will accept connection attempts from the specified Listen Port. |
| Listen Port | <p>The socket port number on the DeviceMaster UP the application will connect to if the Application Listen Enable is selected.</p> |
| Connect To Mode | <p>Specifies if and how the DeviceMaster UP attempts to connect to the application at the specified Connect IP Address and Connect Port.</p> <ul style="list-style-type: none"> • Never - The DeviceMaster UP will not attempt to connect to the application. • Connect-Always - The DeviceMaster UP will attempt to connect to the application until a connection is made. • Connect-On-Data - The DeviceMaster UP will 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 UP will attempt to connect to the application until a connection is made. |
| Connect Port | <p>The application socket port number the DeviceMaster UP will connect to if the Application Connect To Mode is set to either Connect-Always or Connect-On-Data.</p> |
| Connect IP Address | <p>The application IP address the DeviceMaster UP will connect to if the Application Connect To Mode is set to either Connect-Always or Connect-On-Data.</p> |

| Application TCP Connection Configuration (Serial or Socket Port) | |
|--|--|
| Disconnect Mode | Controls if and how the DeviceMaster UP disconnects from an application. <ul style="list-style-type: none"> • Never – The DeviceMaster UP will not disconnect from the application. • Idle - The DeviceMaster UP will disconnect 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 . |

3.6.4. Saving Port Options

After configuring the serial/socket and protocol characteristics for the port, scroll to the bottom of the *Edit Serial Port Configuration* or *Edit Socket Port Configuration* page to save the changes.

The following options are available.

Reset Statistics
 Reset Port
 Save in Flash

| Option | Description |
|-------------------------|---|
| Reset Statistics | Selecting this check box, clears the statistics counters for this port when you select Submit . |
| Reset Port | When you select this check box, the DeviceMaster UP resets the serial port hardware and statistics counters for this port when you click Submit . You must reset the port after modifying the serial port configuration options, including: baud rate, interface mode, parity, data bits, stop bits, flow control, or DTR control. |
| Save in Flash | When you select this check box, the DeviceMaster UP saves changes to port configuration settings in flash memory. These settings are restored when you reboot the DeviceMaster UP. |
| Undo Changes | Restores modified port settings to current values. |
| Submit | Saves changes to port in RAM. If Save in Flash was not selected when you clicked Submit , the changes will revert to original settings when you reboot the DeviceMaster UP. |

3.7. Serial/Ethernet Device Communication Statistics

The top portion of this page provides links to other pages. The displayed counters are the same as those specified in [2.3.3. Serial Port Statistics Object Definition \(72 Hex\)](#) on Page 27. Refer to the following table for definitions of the fields.

Note: The refresh rate on this page is set to 20 seconds. To stop the page refresh, select **Refresh** in your browser. To restart refreshing the page display; exit and return to this page.

| Statistics Unique to Serial Device Interfaces (Top) | |
|---|---|
| Yes/No | Toggles to display or not to display serial statistics. |
| Reset Statistics | 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: <ul style="list-style-type: none"> No STX byte(s) found No ETX byte(s) found Time-outs Packet to large Receive buffer queue overflows |
| <i>(Continued)</i> | |



Serial/Ethernet Device Communication Statistics

- [Server Configuration Home](#)
- [Serial Device Configuration](#)
- [Ethernet Device Configuration](#)
- [PLC Interface Diagnostics](#)

Serial Device Interface Statistics
Yes/No
Reset Statistics

Port - 1

TX Byte Count: 0

TX Pkt Count: 0

RX Byte Count: 0

RX Pkt Count: 0

Parity Error Count: 0

Framing Error Count: 0

Overrun Error Count: 0

To PLC Dropped Packet Count: 0

Rx Con Seq Error Count: 0

Tx Duplicate Seq Errors: 0

Tx Unexpected Seq Errors: 0

Filtering Statistics

Valid Data Items Sent to PLC Interface: 0

Valid Data Items Filtered From PLC: 0

Invalid Data Items Discarded From PLC: 0

Valid Data Items Sent to App Interface: 0

Valid Data Items Filtered From App: 0

Invalid Data Items Discarded From App: 0

RFID Tags With Unknown Formats: 0

Application Connection Statistics

TX Byte Count: 0

TX Pkt Count: 0

To Application Dropped Packet Count: 0

RX Byte Count: 0

RX Pkt Count: 0

To Device Dropped Packet Count: 0

Ethernet Device Interface Statistics
Yes/No
Reset Statistics

Socket-1

Device Connection Statistics

TX Byte Count: 0

TX Pkt Count: 0

RX Byte Count: 0

RX Pkt Count: 0

To PLC Dropped Packet Count: 0

Rx Con Seq Error Count: 0

Tx Duplicate Seq Errors: 0

Tx Unexpected Seq Errors: 0

Filtering Statistics

Valid Data Items Sent to PLC Interface: 0

Valid Data Items Filtered From PLC: 0

Invalid Data Items Discarded From PLC: 0

Valid Data Items Sent to App Interface: 0

Valid Data Items Filtered From App: 0

Invalid Data Items Discarded From App: 0

RFID Tags With Unknown Formats: 0

Application Connection Statistics

TX Byte Count: 0

TX Pkt Count: 0

To Application Dropped Packet Count: 0

RX Byte Count: 0

RX Pkt Count: 0

To Device Dropped Packet Count: 0

Reboot



| Statistics Unique to Serial Device Interfaces (continued) | |
|--|---|
| Rx Con Seq Errors Count | <p>Displays the number of <i>received consumed sequenced number</i> errors. The DeviceMaster UP only increments this number when all of the following statements are true:</p> <ul style="list-style-type: none"> You selected the <i>Write-to-Tag-Sync</i> method of receive communication. DeviceMaster UP receives a serial packet. The <i>Consumed Sequence</i> number is out of sync. (It is not equal to the <i>Produced Sequence</i> number or equal to the <i>Produced Sequence</i> number minus one.) |
| Tx Duplicate Seq Errors | <p>Displays the number of <i>Duplicate Transmit Sequence Number</i> errors. The DeviceMaster UP increments this number when the following statements are true:</p> <ul style="list-style-type: none"> You enabled the <i>Transmit Sequence Number Checking configuration</i> option. (See <i>Attribute 16 - Serial Port Transfer Options</i> on Page 18 for additional information.) DeviceMaster UP receives a transmit message with a sequence number equal to the previous sequence number. (The DeviceMaster UP expects this sequence number to be incremented by one from the sequence number in the previous transmit message.) |
| Tx Unexpected Seq Errors | <p>Displays the number of <i>Unexpected Transmit Sequence Number</i> errors. The DeviceMaster UP increments this number when the following statements are true:</p> <ul style="list-style-type: none"> You enabled the <i>Transmit Sequence Number Checking configuration</i> option. (See <i>Attribute 16 - Serial Port Transfer Options</i> on Page 18 for additional information.) DeviceMaster UP 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 UP expects this sequence number to be incremented by one with each new transmit message.) |

| Filtering Statistics (Serial or Ethernet Device Interfaces) | |
|--|---|
| Valid Data Items Sent To PLC | 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 Application | Displays the number of valid string, RFID, or barcode data sent to the application. Applies when filtering is enabled. |
| Valid Data Items Filtered From Application | 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 (Serial or Ethernet Device Interfaces) | |
|---|--|
| 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: <ul style="list-style-type: none"> • No STX byte(s) found • No ETX byte(s) found • Time-outs • Packet too 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. |

| Statistics Unique to Ethernet Device Interfaces | |
|--|--|
| Yes/No | Toggles to display or not to display socket statistics. |
| 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 Application Dropped Packet Count | Displays the number of dropped packets that were intended for the application. |
| Rx Con Seq Error Count | Same as the serial port statistics, except the packet was received over a socket port (Page 71). |
| Tx Duplicate Seq Errors | Same as the serial port statistics (Page 71). |
| Tx Unexpected Seq Errors | Same as the serial port statistics (Page 71). |

3.8. PLC Interface Diagnostics

The *PLC Interface Diagnostics* page provides detailed statistics and error reporting for the EtherNet/IP PLC interface. It is intended to help with debugging PLC programs, monitoring the PLC interface, and solving configuration problems.

| PLC Interface Diagnostics | |
|---|---|
| Messages/Responses Received From PLC | Displays the number of messages and responses received from the PLC(s). |
| Broadcasts Received From PLC | Displays the number of broadcast messages received from the PLC(s). |
| Messages/Responses Sent To PLC | Displays the number of messages and responses sent to the PLC(s). |
| Request Messages From PLC | Displays the number of request messages received from the PLC(s). |
| Bad Responses To Msgs Sent To PLC | <p>Displays the number of bad responses from messages sent to the PLC(s). Bad responses are typically returned for such errors as:</p> <ul style="list-style-type: none"> • 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 |
| Invalid Network Path Errors | Displays the number of network path errors on messages sent to the PLC(s). These are typically caused by incorrect IP address settings. |
| No Response From PLC Errors | <p>Displays the number of no responses from messages sent to the PLC(s). No responses are typically returned for such errors as:</p> <ul style="list-style-type: none"> • Incorrect IP address • Incorrect PLC configuration • PLC malfunction • PLC is overloaded and cannot handle the amount of Ethernet traffic |
| Pending Request Limit Errors | Displays the number of pending request limit errors. These errors occur when the PLC is sending a continuous stream of messages to the DeviceMaster UP faster than the DeviceMaster UP can process them. |
| <i>(Continued)</i> | |



PLC Interface Diagnostics

- [Server Configuration Home](#)
- [Serial Device Configuration](#)
- [Ethernet Device Configuration](#)
- [Communication Statistics](#)

EtherNet/IP Interface Statistics

| | |
|--|-------------------|
| Messages/Responses Received From PLC: | 0 |
| Broadcasts Received From PLC: | 0 |
| Messages/Responses Sent To PLC: | 0 |
| Request Messages From PLC: | 0 |
| Bad Responses to Msgs Sent To PLC: | 0 |
| Invalid Network Path Errors: | 0 |
| No Response From PLC Errors: | 0 |
| Pending Request Limit Errors: | 0 |
| Unexpected Event Errors: | 0 |
| Unsupported CIP Request Instance Errors: | 0 |
| Unsupported CIP Request Service Errors: | 0 |
| Unsupported CIP Request Class Errors: | 0 |
| Unsupported CIP Request Attribute Errors: | 0 |
| Improper Configuration Errors: | 0 |
| Invalid Message Data Errors: | 0 |
| System Resource Errors: | 0 |
| Writes To Offline Ethernet Device on Socket 1: | 0 |
| First Error Description: | No Error Detected |
| Last Error Description: | |



| PLC Interface Diagnostics (cont.) | |
|--|---|
| Unexpected Event Errors | Displays the number of unexpected event errors. Unexpected event errors occur when the DeviceMaster UP receives an unexpected message from the PLC such as an unexpected response or unknown message. |
| Unsupported CIP Request 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 UP. |
| Unsupported CIP Request 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 UP. |
| Unsupported CIP Request 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 UP. |
| Unsupported CIP Request 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 UP. |
| Improper Configuration Errors | Displays the number of improper configuration errors. These errors occur when the DeviceMaster UP receives a message that cannot be performed due to an invalid configuration. |
| Invalid Message Data Errors | Displays the number of invalid message data errors. These errors occur when the DeviceMaster UP 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 UP 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 UP faster than the DeviceMaster UP can process it. |
| 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. |

3.9. Edit Network Configuration Page

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

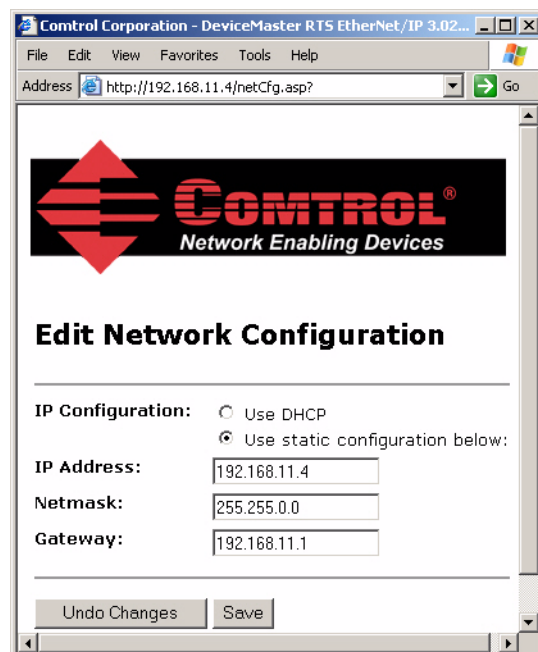
Use the following procedure to change the network configuration.

1. Select the IP configuration type (**DHCP** or **Static**).
2. 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.*

3. Select **Save** or **Undo Changes** to close the page.
4. If you selected **Save**, select **Reboot** to program the network information into the DeviceMaster UP or **Continue** if you want to reboot later.

Note: *Changed network settings will not take affect until the DeviceMaster UP is rebooted.*



Chapter 4. Programming the PLC

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

4.1. Programming Guidelines

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

| Communications Methods | Description |
|--|---|
| Unsolicited - Write-to-Tag/File (<i>Recommended Method</i>) | DeviceMaster UP 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 UP can send it |
| Unsolicited - Write-to-Tag/File-Synced | DeviceMaster UP 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 UP. 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 UP do not overflow. |

- *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 UP 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 1024 serial bytes and 2048 socket bytes), the DeviceMaster UP will send the data to a series of tags.

See [2.3.2. Serial Port Data Transfer Object Definition \(71 Hex\)](#) on Page 25 and [2.3.5. Socket Port Data Transfer Definition Object \(74 Hex\)](#) on Page 37 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 UP 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 1024 serial bytes and 2048 socket bytes), the DeviceMaster UP will send the data to a series of files in sequential order.

See [2.4.3. Receive Data Message](#) on Page 48 for details.

- Only one PLC generated EtherNet/IP message may be active at any one time in a PLC program.

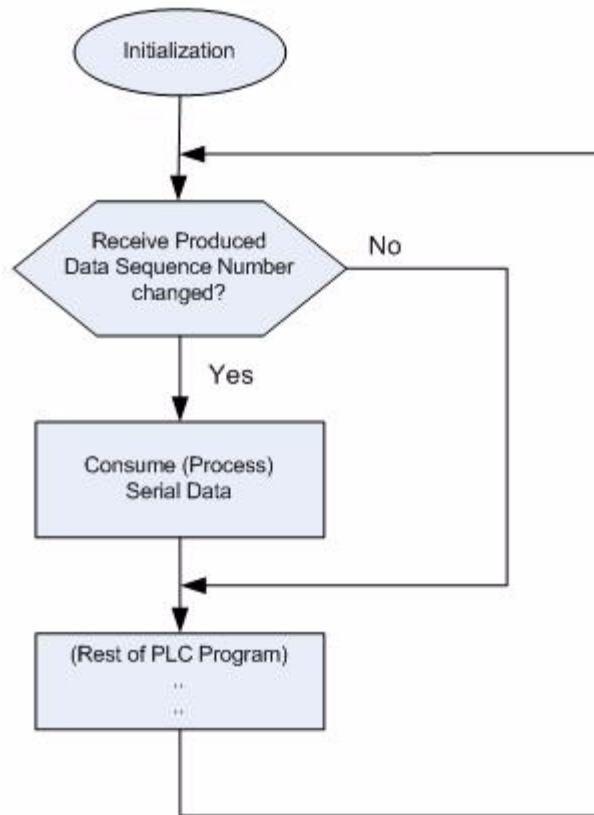
4.2. PLC Algorithms

This section displays the following PLC algorithms:

- [Unsolicited - Write-to-Tag/File PLC Algorithm](#) on Page 78
- [Unsolicited - Write-to-Tag/File-Synced PLC Algorithm](#) on Page 79
- [Polling PLC Algorithm](#) on Page 80

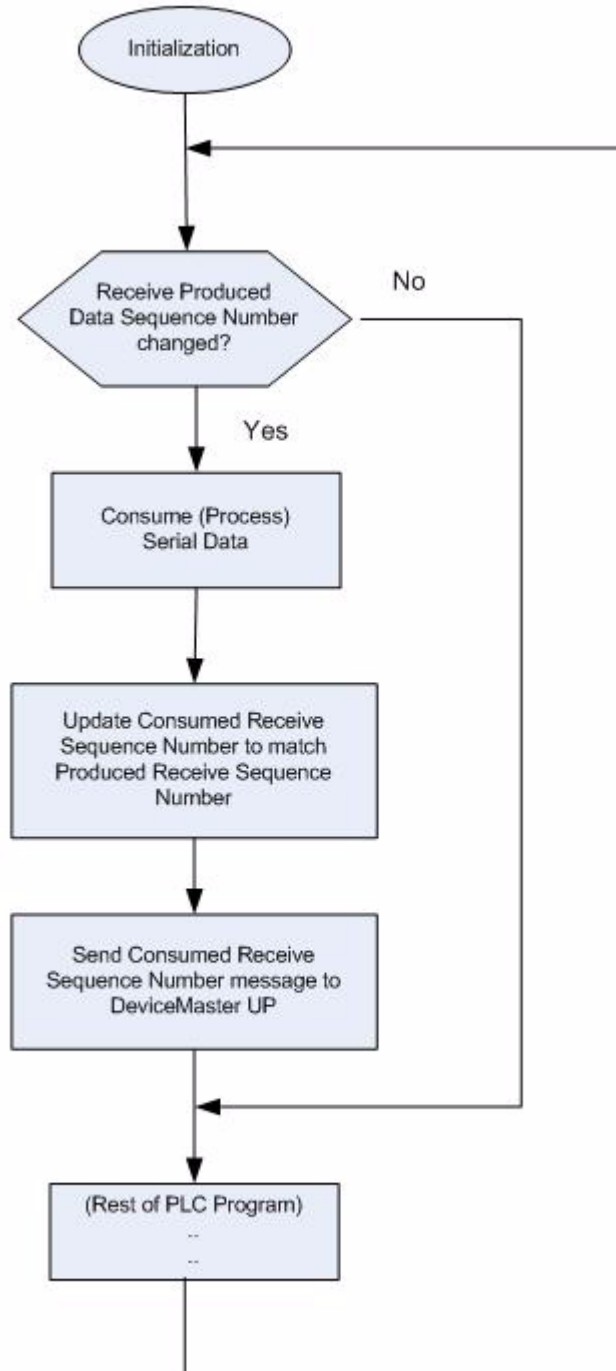
4.2.1. Unsolicited - Write-to-Tag/File PLC Algorithm

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



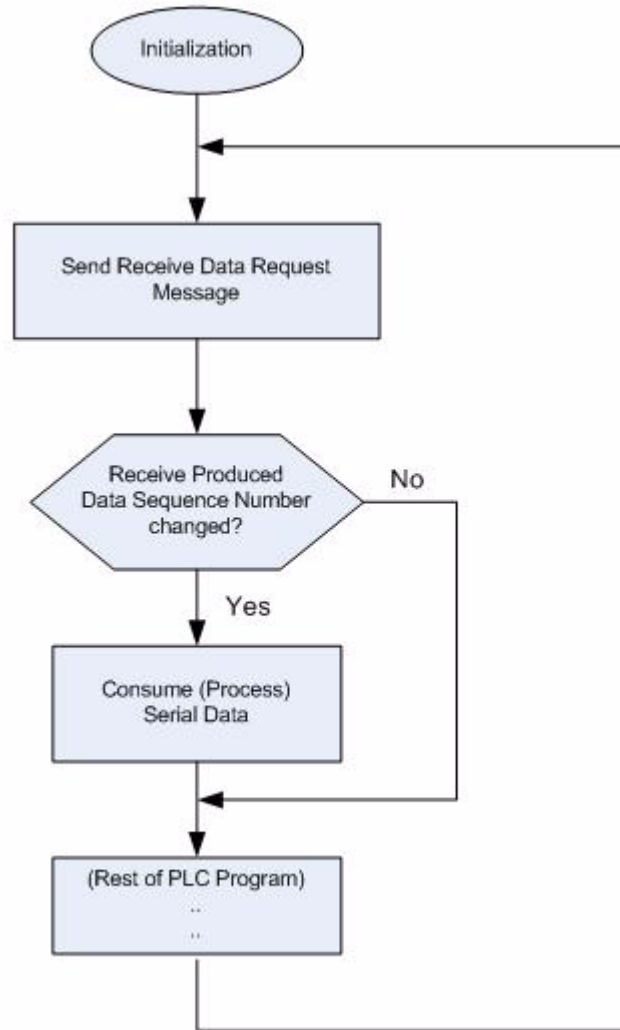
4.2.2. Unsolicited - Write-to-Tag/File-Synced PLC Algorithm

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



4.2.3. Polling PLC Algorithm

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



4.3. ControlLogix PLC Programming Example Instructions

This topic describes how to use RSLogix 5000 to configure and run the DeviceMaster UP 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 **Control/EtherNet/IP** 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:

- [4.3.3. loopbackExampleTagWrite.L5K](#) on Page 82
- [4.3.4. loopbackExampleTagWriteSynced.L5K](#) on Page 83
- [4.3.5. loopbackExamplePolling.L5K](#) on Page 84

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 UP 1-port or on Port 1 of a DeviceMaster UP 4-port. Additional programming is required to use all ports on a DeviceMaster UP 4-port.*



Disclaimer: Control 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. Control 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.

4.3.1. 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.*

4.3.2. Requirements

- The DeviceMaster UP must be installed and configured as described in the [DeviceMaster UP Hardware Installation Guide](#).
- The DeviceMaster UP 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 UP. See the [DeviceMaster UP Hardware Installation 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. Locating Updated Software and Documents](#) on Page 9 for the location of the PLC program examples.

4.3.3. 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 UP 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 [4.3.7. RSLogix 5000 Screen Examples](#) on Page 86.

1. Attach a loopback plug to the serial port. (If necessary, see the [DeviceMaster UP Hardware Installation Guide](#).)
2. Open **RSLogix 5000**.
3. Import the **loopbackExampleTagWrite.L5K** file.

Note: *If you have problems loading the PLC program example, see [4.3.8. Modifying an RSLogix 5000 PLC Program Example \(Older Versions\)](#) on Page 93 for a solution.*

4. Modify the PLC program (**loopbackExampleTagWrite.L5K**) for your system. [2.3. EtherNet/IP Interface Profile \(ControlLogix\)](#) on Page 13 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 ControlCLX** 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 UP IP Address>`

Where `<Ethernet Interface Name>` specifies the name of your Ethernet interface and `<DeviceMaster UP 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 [4.3.6. Configuring the DeviceMaster UP for the RSLogix 5000 Example Programs Using the Web Page](#) on Page 85.

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.

4.3.4. 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 UP 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 UP 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 [4.3.7. RSLogix 5000 Screen Examples](#) on Page 86.

1. Attach a loopback plug to the serial port. (If necessary, see the [DeviceMaster UP Hardware Installation Guide](#).)

2. Open **RSLogix 5000**.

3. Import the **loopbackExampleTagWriteSynced.L5K** file.

Note: If you have problems loading the PLC program example, see [4.3.8. Modifying an RSLogix 5000 PLC Program Example \(Older Versions\)](#) on Page 93 for a solution.

4. Modify the PLC program (**loopbackExampleTagWriteSynced.L5K**) for your system.

[2.3. EtherNet/IP Interface Profile \(ControlLogix\)](#) on Page 13 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 ControlCLX** 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 UP IP Address>

Where **<Ethernet Interface Name>** specifies the name of your Ethernet interface and **<DeviceMaster UP 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 [4.3.6. Configuring the DeviceMaster UP for the RSLogix 5000 Example Programs Using the Web Page](#) on Page 85.

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.

4.3.5. loopbackExamplePolling.L5K

This example program demonstrates an RSLogix 5000 loopback PLC program using the *Polling* receive method. This program configures a DeviceMaster UP 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 [4.3.7. RSLogix 5000 Screen Examples](#) on Page 86.

1. Attach a loopback plug to the serial port. (If necessary, see the [DeviceMaster UP Hardware Installation Guide](#).)
2. Open **RSLogix 5000**.
3. Import the **loopbackExamplePolling.L5K** file.

Note: *If you have problems loading the PLC program example, see [4.3.8. Modifying an RSLogix 5000 PLC Program Example \(Older Versions\)](#) on Page 93 for a solution.*

4. Modify the PLC program (**loopbackExamplePolling.L5K**) for your system.

[2.3. EtherNet/IP Interface Profile \(ControlLogix\)](#) on Page 13 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 ControlCLX** 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 UP IP Address>`

Where `<Ethernet Interface Name>` specifies the name of your Ethernet interface and `<DeviceMaster UP 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 [4.3.6. Configuring the DeviceMaster UP for the RSLogix 5000 Example Programs Using the Web Page](#) on Page 85.

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.

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

The following procedure configures the DeviceMaster UP 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 *Server Configuration* web pages, see [Chapter 3. Configuring the DeviceMaster UP](#) on Page 55.

1. Attach a loopback plug to the serial port.
2. Access the *Server Configuration* web page, using one of these methods.
 - Open PortVision Plus, right-click on the DeviceMaster UP for which you want to program network information and select **Web Manager**.
 - Open a browser and type the IP address for the DeviceMaster UP in the **Address** box.
3. Click **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. Set the serial port settings under **Ethernet/IP Settings** to the following values:

| Field | Selection |
|---|---|
| TX Sequence Number Checking | Select. |
| Rx (To PLC) Ethernet Transfer Method | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • Leave blank for Polling. • Set to Com1_RxData, the PLC receive filename for Write-to-Tag and Write-to-Tag-Synced. |
| Reset Port | Select. |
| Save in Flash | Select. |

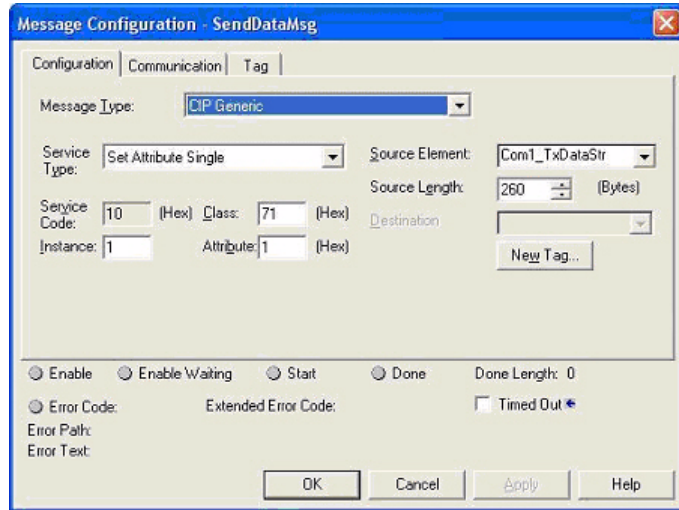
7. Click **Submit**.

4.3.7. RSLogix 5000 Screen Examples

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

4.3.7.1. Transmit Data to the DeviceMaster UP

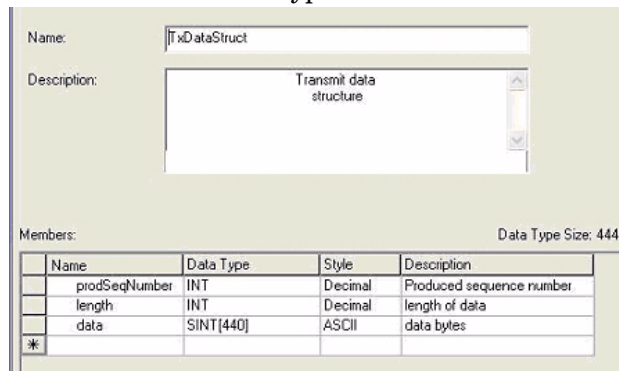
Use the **Configuration** tab on the **Message Configuration** dialog to set options for transmitting data through a specified port on the DeviceMaster UP.



Provide the following information.

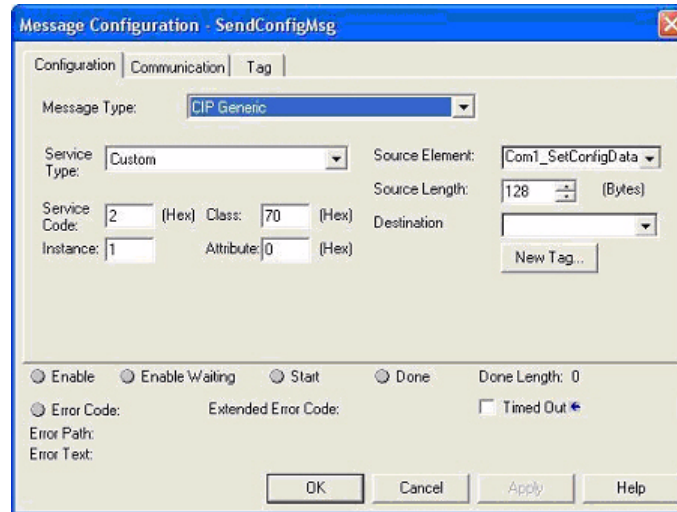
| Field | Selection |
|----------------|--|
| Message Type | CIP Generic |
| Service Type | Set Attribute Single |
| Service Code | 10 Hex (Set Attribute Single) |
| Class | 71 Hex (Serial Port Data Transfer object) |
| Instance | 1 (Port 1) |
| Attribute | 1 (Transmit message data attribute) |
| Source Element | Com1_TxDataStr The transmit data structure includes: <ul style="list-style-type: none"> • 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.



4.3.7.2. Send Configuration to DeviceMaster UP

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



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:

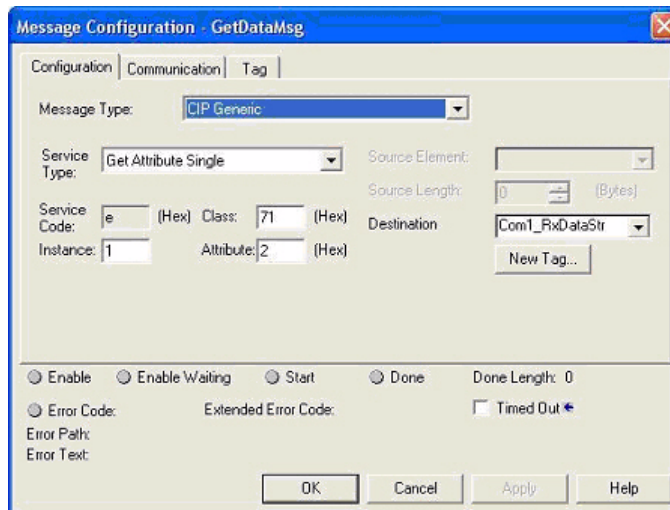
Description:

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 | |
| dtControl | SINT | Decimal | |
| txStLength | SINT | Decimal | |
| txStValue1 | SINT | Decimal | |
| txStValue2 | SINT | Decimal | |
| txEtLength | SINT | Decimal | |
| txEtValue1 | SINT | Decimal | |
| txEtValue2 | SINT | Decimal | |
| reserved1 | INT | Decimal | |
| rxStLength | SINT | Decimal | |
| rxStValue1 | SINT | Decimal | |
| rxStValue2 | SINT | Decimal | |
| rxEtLength | SINT | Decimal | |
| rxEtValue1 | SINT | Decimal | |
| rxEtValue2 | SINT | Decimal | |
| rxTimeBetweenP | INT | Decimal | |
| serPrtXferOptions | INT | Decimal | |
| rxNetDataVerM | SINT | Decimal | |
| reserved2 | SINT | Decimal | |
| reserved3 | INT | Decimal | |
| reserved4 | INT | Decimal | |
| reserved5 | SINT | Decimal | |
| plcSlotNumber | SINT | Decimal | |
| plcIPAddress | DINT | Decimal | |
| rxDataTagName | SINT[40] | ASCII | |
| reservedTagName | SINT[40] | ASCII | |

4.3.7.3. Request Data from DeviceMaster UP

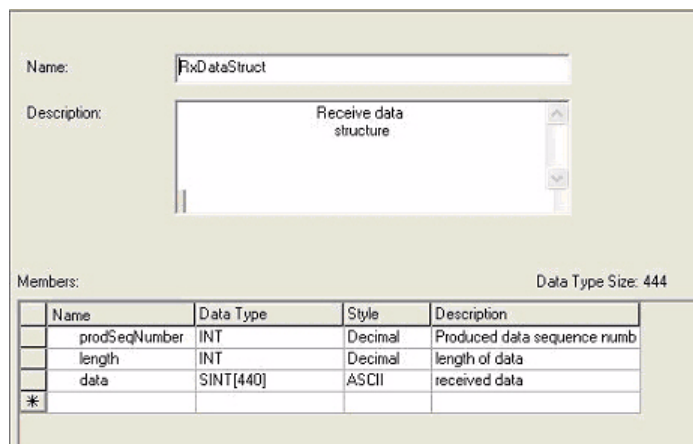
Use the **Configuration** tab on the **Message Configuration** dialog to request data from a specified serial port on the DeviceMaster UP. Only the *Polling* receive method uses this feature.



Provide the following information.

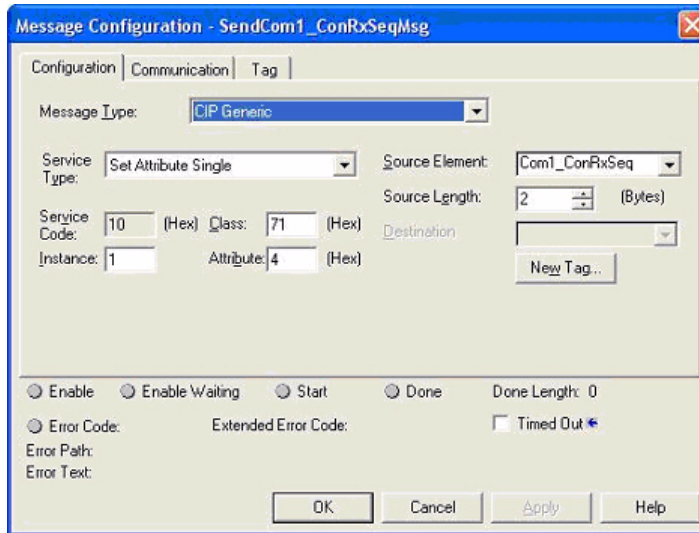
| Field | Selection |
|---------------|--|
| Message Type | CIP Generic |
| Service Type | Get Attribute Single |
| Service Code | 0E Hex (Get Attribute Single) |
| Class | 71 Hex (Serial Port Data Transfer object) |
| Instance | 1 (Port 1) |
| Attribute | 2 (Receive message data attribute) |
| Destination | Com1_RxDataStr (Receive data structure) The Receive data structure includes: <ul style="list-style-type: none"> 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.



4.3.7.4. Send Consumed Sequence Number to DeviceMaster UP

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

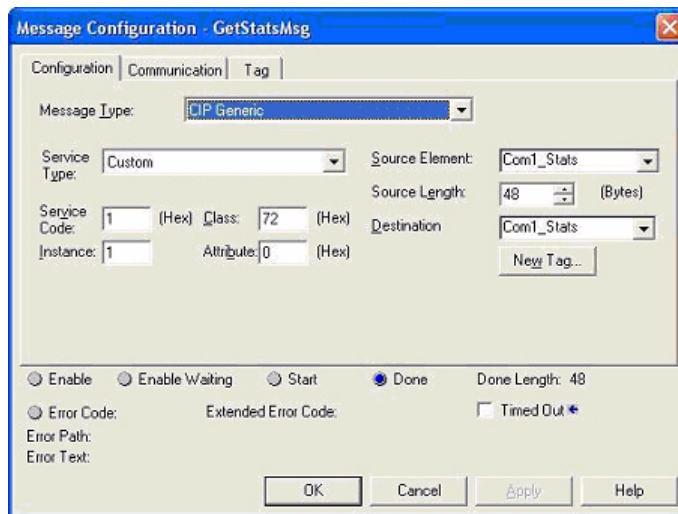


Provide the following information:

| 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) |

4.3.7.5. Request Statistics from DeviceMaster UP

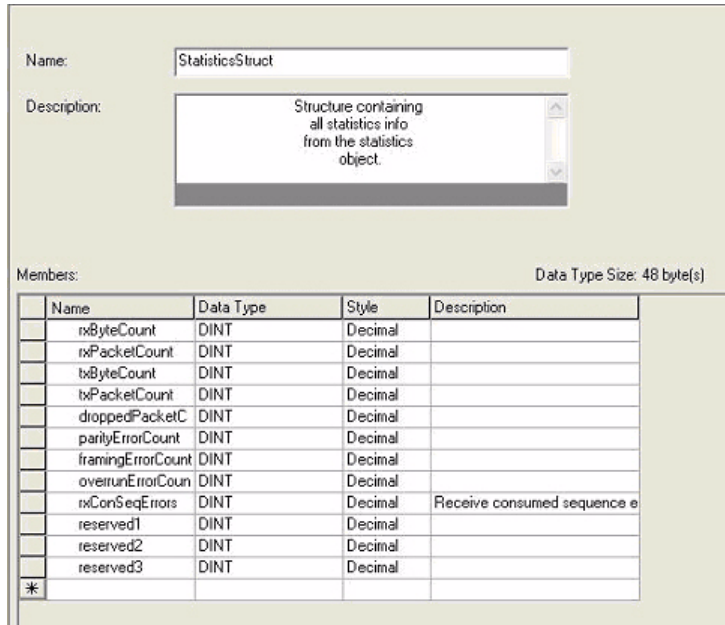
Use the **Configuration** tab on the **Message Configuration** dialog for requesting statistics for a specified port from the DeviceMaster UP.



Provide the following information.

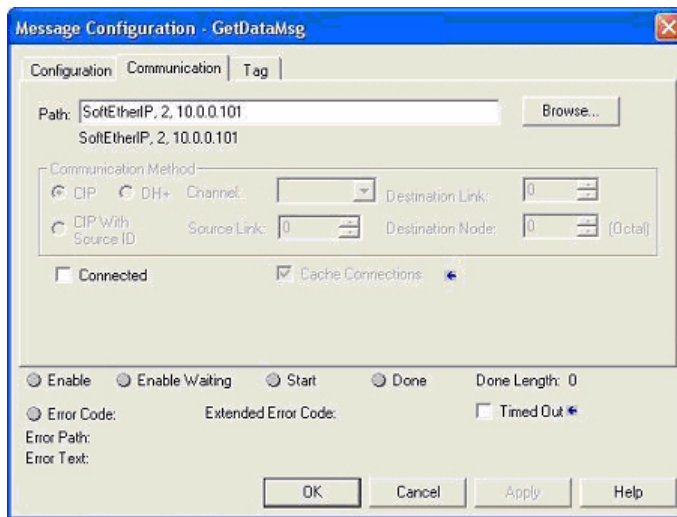
| Field | Selection |
|----------------|--|
| Message Type | CIP Generic |
| Service Type | Custom |
| Service Code | 1 Hex (Get Attribute All) |
| Class | 72 Hex (Serial Port Statistics object) |
| Instance | 1 (Port 1) |
| Attribute | 0 (Unused) |
| Source Element | Com1_Stats (not actually used) |
| 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.



4.3.7.6. Communication Window for all Messages sent to the DeviceMaster UP

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



Provide the following information.

| Field | Selection |
|-------|---|
| Path | <p>Provide the following information in this box:</p> <ul style="list-style-type: none"> • 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 UP used to create the example programs. |

4.3.8. 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:
 - [4.3.3. loopbackExampleTagWrite.L5K](#) on Page 82
 - [4.3.4. loopbackExampleTagWriteSynced.L5K](#) on Page 83
 - [4.3.5. loopbackExamplePolling.L5K](#) on Page 84
 - 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:
 - [4.3.3. loopbackExampleTagWrite.L5K](#) on Page 82
 - [4.3.4. loopbackExampleTagWriteSynced.L5K](#) on Page 83
 - [4.3.5. loopbackExamplePolling.L5K](#) on Page 84
 - 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.

4.4. SLC or MicroLogix PLC Programming Example Instructions

This topic describes how to use RSLogix 500 to configure and run the DeviceMaster UP 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 **Control/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:

- [4.4.4. *lpbkExampleSlcMsgPollRS500*](#)
- [4.4.5. *lpbkExamplePlc5MsgPollRS500*](#)

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 UP 1-port or on Port 1 of a DeviceMaster UP 4-port. Additional programming is required to use all ports on a DeviceMaster UP 4-port.*



Caution

Disclaimer: Control 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. Control 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.

4.4.1. 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.*

4.4.2. Requirements

- The DeviceMaster UP must be installed and configured as described in the [DeviceMaster UP Hardware Installation Guide](#).
- The DeviceMaster UP 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 UP when running an example PLC program. If necessary, see the [DeviceMaster UP Hardware Installation 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. Locating Updated Software and Documents](#) on Page 9, for the location of the PLC program examples.

4.4.3. 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 UP at the start of the programs, the only requirement is that the sequence numbers be in sync between the PLC and DeviceMaster UP.
- Statistics retrieval is not included in the example programs, but you can easily add it by inserting a request statistics message.

4.4.4. lpbkExampleSlcMsgPollRS500

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 UP 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.

4.4.5. lpbkExamplePlc5MsgPollRS500

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 UP 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.

4.4.5.1. lpbkExampleSlcMsgFileRS500

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 UP. 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.

4.4.5.2. lpbkExampleSlcMsgFileSyncRS500

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 UP. The consumed receive sequence number is updated to match the produced receive sequence number and sent to the DeviceMaster UP 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.

4.4.6. Configure the DeviceMaster UP for the RSLogix 500 Example Program

The following procedure configures the DeviceMaster UP 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 *Server Configuration* web pages, see [Chapter 3. Configuring the DeviceMaster UP](#) on Page 55.

1. Attach a loopback plug to the serial port.
2. Access the *Server Configuration* web page, using one of these methods.
 - Open PortVision Plus, right-click on the DeviceMaster UP for which you want to program network information and select **Web Manager**.
 - Open a browser and type the IP address for the DeviceMaster UP in the **Address** box.
3. Click **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 |
| (PLC-5/SLC) Rx MS Byte First | Optionally, select |
| (PLC-5/SLC) Tx MS Byte First | Optionally, select |

6. Set the serial port settings under **Ethernet/IP Settings** to the following values:

| Field | Selection |
|---|---|
| TX Sequence Number Checking | Select. |
| Rx (To PLC) Ethernet Transfer Method | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • Leave blank for Polling. • Set to \$N10:0, the PLC receive filename for Write-to-File and Write-to-File-Synced. |
| Reset Port | Select. |
| Save in Flash | Select. |

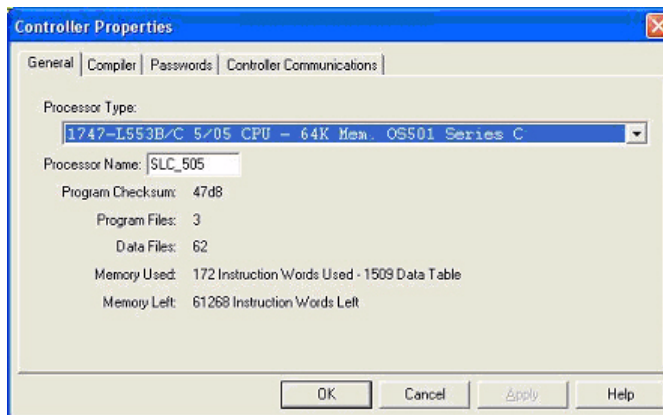
7. Click **Submit**.

4.4.7. Configure and Run the RSLogix 500 Example Program

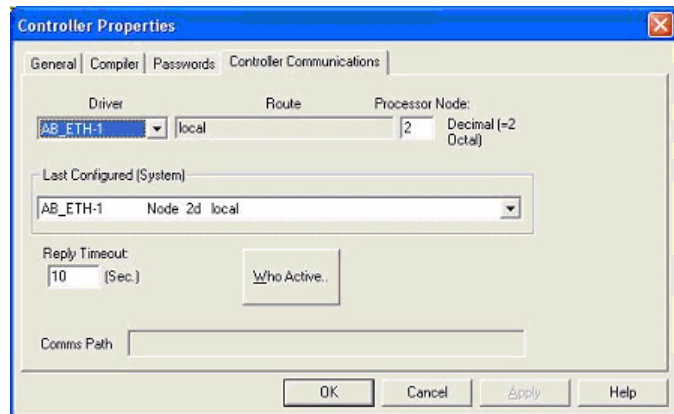
You can configure and run the RSLogix 500 example programs through RSLogix 500. For additional information on the RSLogix 500, see [4.4.8. RSLogix 500 Screen Examples](#) on Page 101.

Note: Configure the DeviceMaster UP before you configure and download the RSLogix 500 example program. For instructions on configuring DeviceMaster UP, see [4.4.6. Configure the DeviceMaster UP for the RSLogix 500 Example Program](#), 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**.

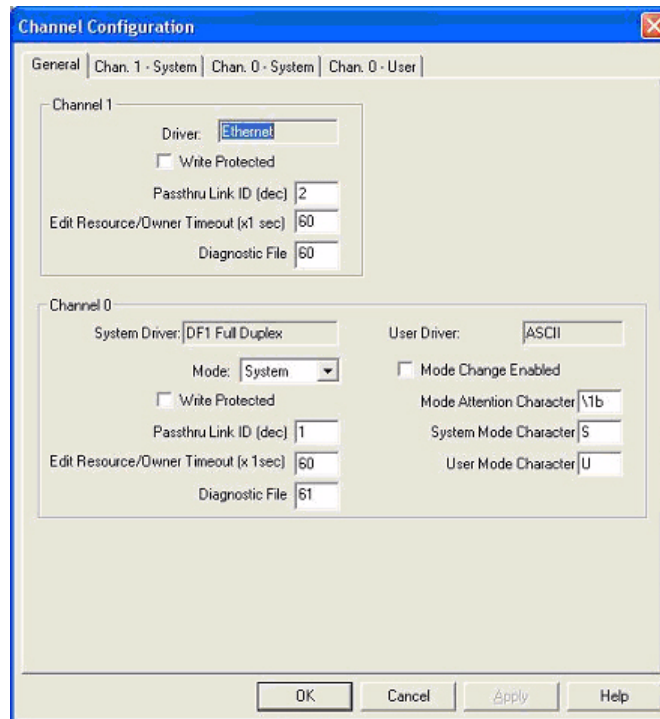
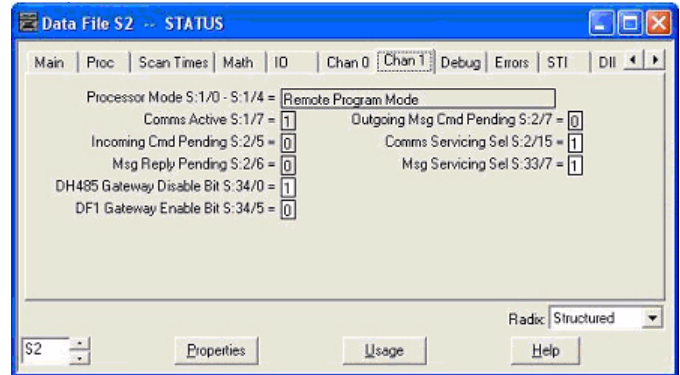


5. Select the **Controller Communications** tab and select the following options:
 - a. Set **Driver** to the appropriate type to allow RSLogix 500 to communicate with the SLC processor.
 - b. Type the processor node number in the **Processor Node** box. (You may reference the **Last Configured (System)** node or select **Who Active**.)

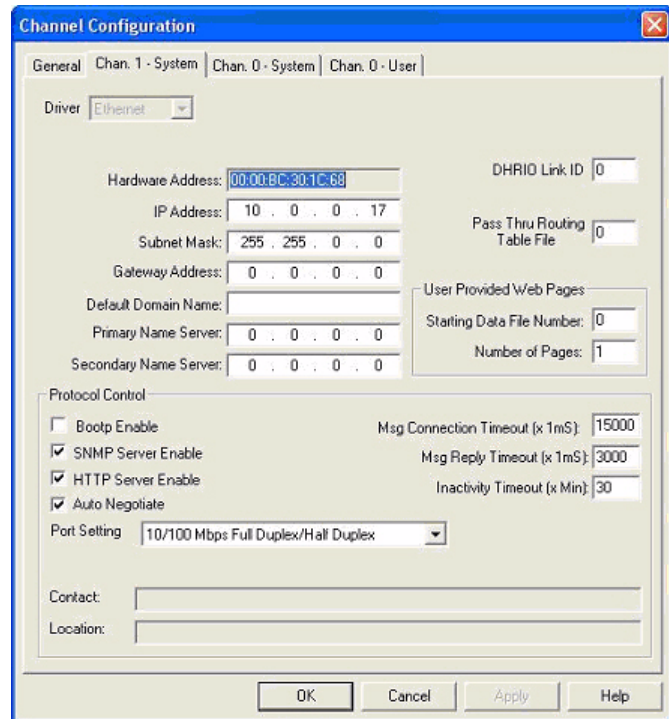


6. Click **OK** to apply your changes and close the **Controller Properties** dialog.
7. Double-click **Processor Status**, select the **Chan 1** tab on the **Data File S2 -- STATUS** dialog and verify the following:
 - a. **DH485 Gateway Disable Bit S:34/0** option is set to **1**.
 - b. **DF1 Gateway 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.



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.
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 UP 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.



4.4.8. RSLogix 500 Screen Examples

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

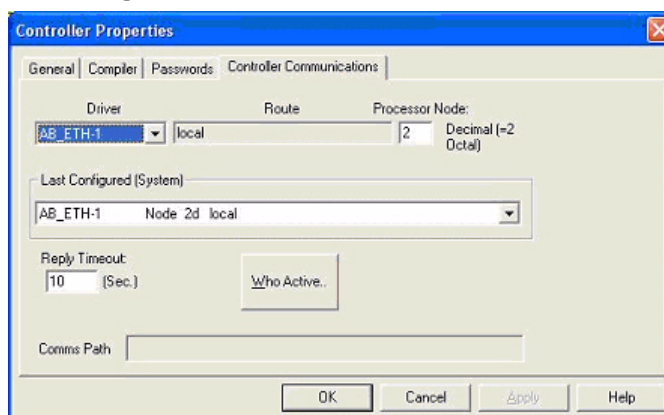
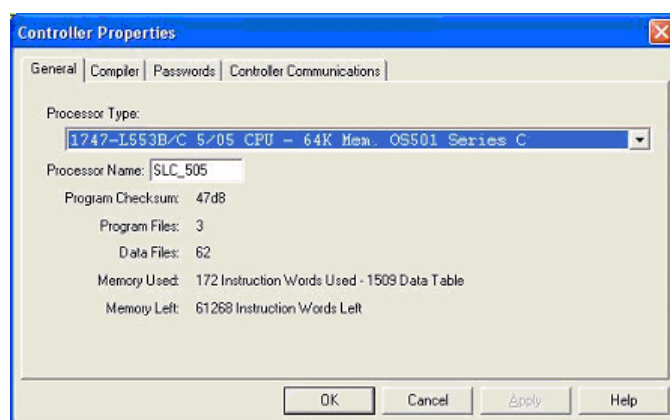
4.4.8.1. Setting up Processor and EtherNet (Channel 1)

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).

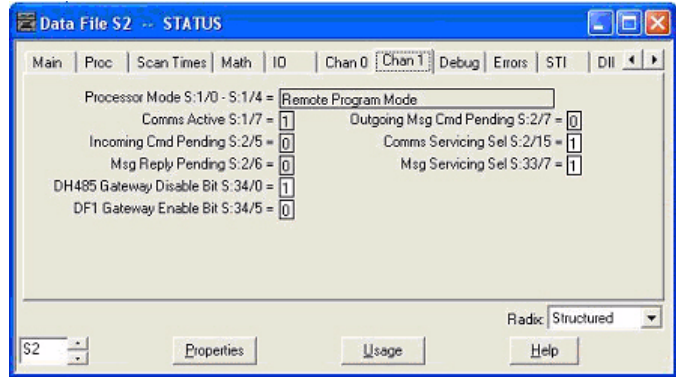
The following screens show the recommended settings that allow EtherNet/IP to function properly on an SLC or MicroLogix PLC.

1. Start **RSLogix 500**.
2. Double-click **Controller Properties** and select the correct processor type from the **General** tab on the **Controller Properties** dialog.
3. Select the **Controller Communications** tab and select the proper driver for RSLogix 500.

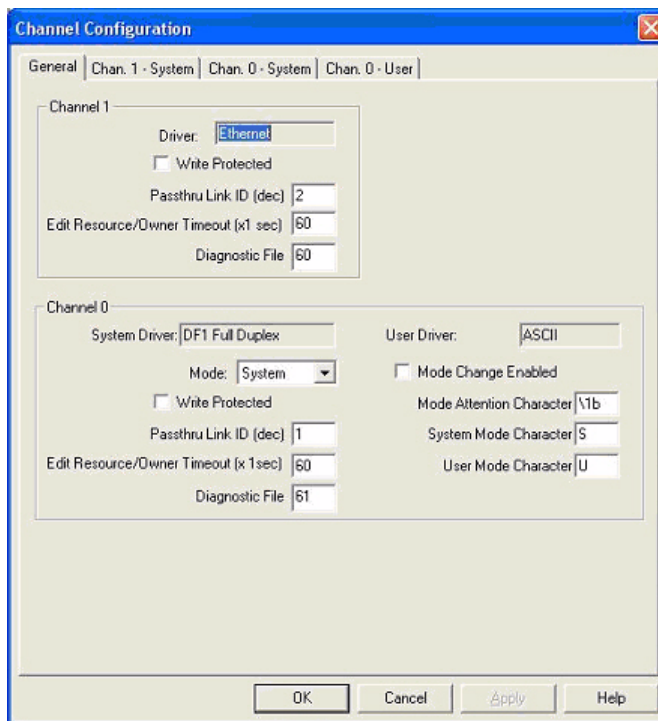


4. Click **OK** to apply your changes and close the **Controller Properties** dialog.

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.)



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.



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.

Channel Configuration

General | **Chan. 1 - System** | Chan. 0 - System | Chan. 0 - User

Driver: Ethernet

Hardware Address: 00:00:8C:30:1C:68 | DHRIO Link ID: 0

IP Address: 10 . 0 . 0 . 17 | Pass Thru Routing Table File: 0

Subnet Mask: 255 . 255 . 0 . 0

Gateway Address: 0 . 0 . 0 . 0

Default Domain Name: | User Provided Web Pages

Primary Name Server: 0 . 0 . 0 . 0 | Starting Data File Number: 0

Secondary Name Server: 0 . 0 . 0 . 0 | Number of Pages: 1

Protocol Control

Bootp Enable | Msg Connection Timeout (x 1mS): 15000

SNMP Server Enable | Msg Reply Timeout (x 1mS): 3000

HTTP Server Enable | Inactivity Timeout (x Min): 30

Auto Negotiate

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

Contact: |

Location: |

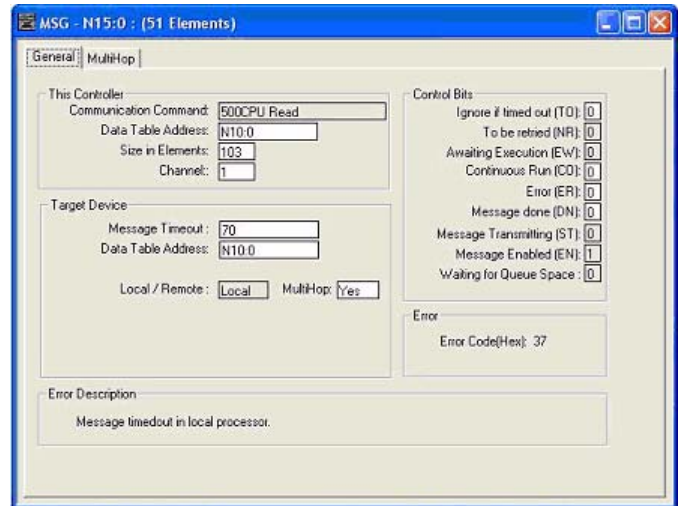
OK Cancel Apply Help

4.4.8.2. SLC Typed Read - Receive Data Message

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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - 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.
 - d. Note that the **Message Timeout** parameter is not actually configurable. RSLogix 500 will set the value in this box based on the EtherNet timeout settings.
 - e. Specify the port-specific read file address for the DeviceMaster UP in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - f. Set the **MultiHop** option to **Yes**.

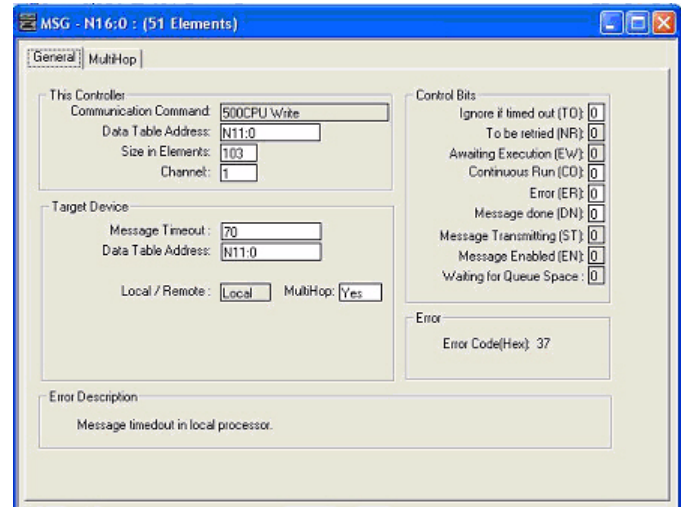
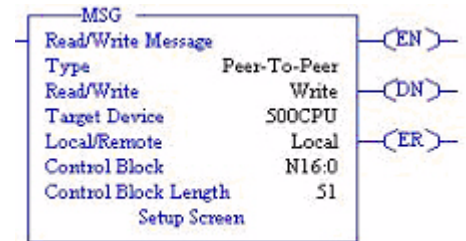


4.4.8.3. SLC Typed Write - Transmit Data Message

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

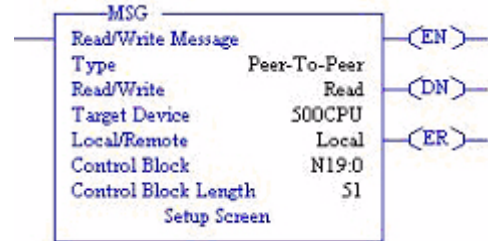
Make the following changes to the ladder logic.

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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - 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.
 - d. Note that the **Message Timeout** parameter is not actually configurable. RSLogix 500 will set the value in this box based on the EtherNet timeout settings.
 - e. Specify the port-specific transmit file address for the DeviceMaster UP in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - f. Set the **MultiHop** option to **Yes**.



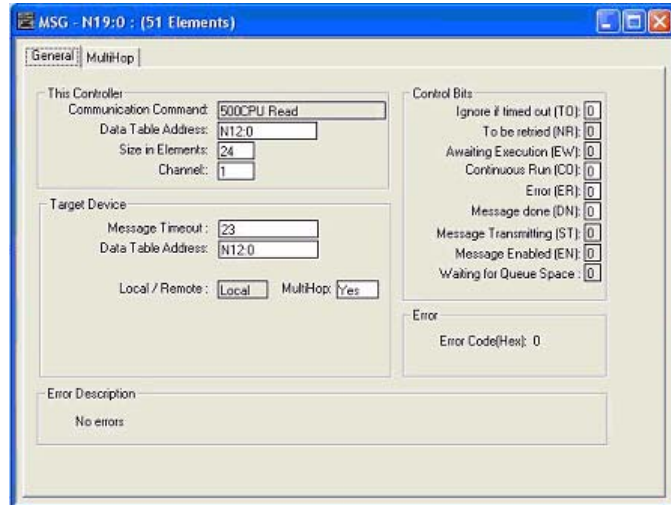
4.4.8.4. SLC Typed Read - Retrieve Statistics Message

The following screen depicts an *SLC Typed Read - Retrieve Statistics* 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 statistics data information in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - 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.
 - d. Note that the **Message Timeout** parameter is not actually configurable. RSLogix 500 will set the value in this box based on the EtherNet timeout settings.
 - e. Specify the port-specific statistics file address for the DeviceMaster UP in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - f. Set the **MultiHop** option to **Yes**.

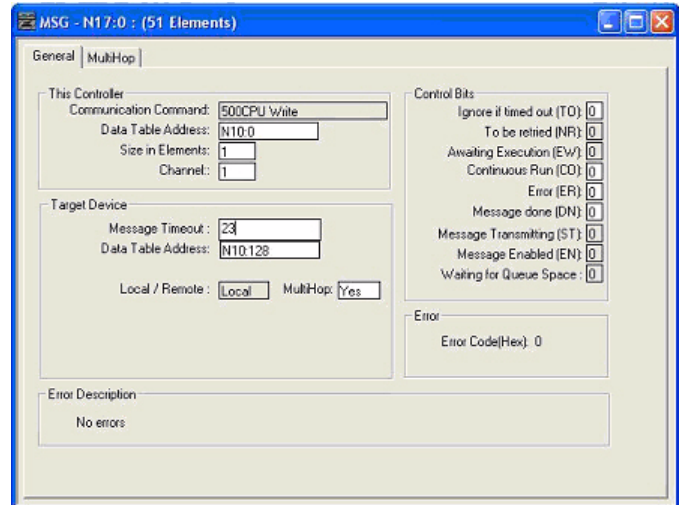


4.4.8.5. SLC Typed Write - Set Receive Produced Sequence Number Message

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

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 receive sequence number resides in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - b. Type **1** in the **Size of Elements** box.
 - c. Set the **Channel** parameter to **1** to use the EtherNet port.
 - d. Note that the **Message Timeout** parameter is not actually configurable. RSLogix 500 will set the value in this box based on the EtherNet timeout settings.
 - e. Specify the port-specific receive sequence file address for the DeviceMaster UP in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - f. Set the **MultiHop** option to **Yes**.

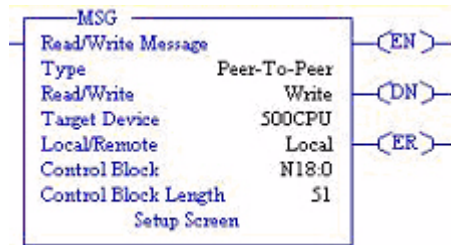


4.4.8.6. SLC Typed Write - Set Transmit Produced Sequence Number Message

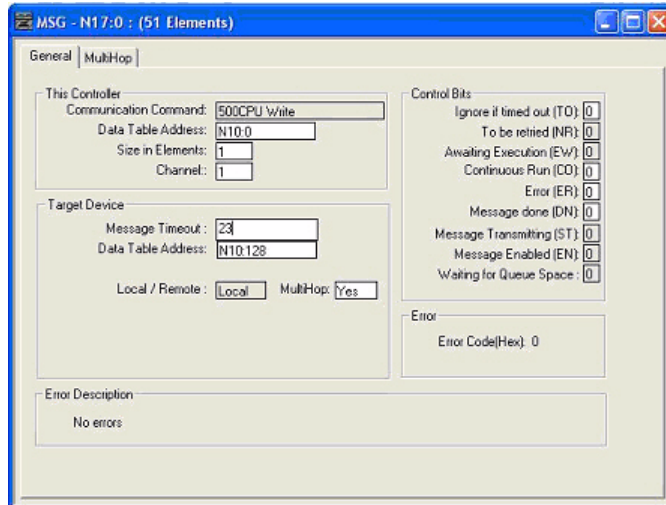
The following screen depicts an *SLC Typed Write - Set Transmit Produced Sequence Number* message in ladder logic.

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:

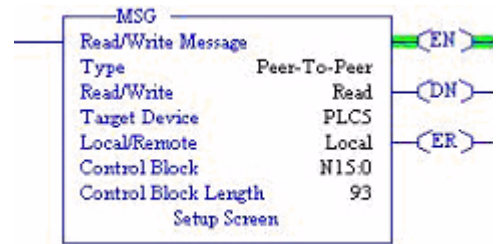


- a. Specify the file address where the transmit sequence number resides in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
- b. Type **1** in the **Size of Elements** box.
- c. Set the **Channel** parameter to **1** to use the EtherNet port.
- d. Note that the **Message Timeout** parameter is not actually configurable. RSLogix 500 will set the value in this box based on the EtherNet timeout settings.
- e. Specify the port-specific transmit sequence number file address for the DeviceMaster UP in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
- f. Set the **MultiHop** option to **Yes**.



4.4.8.7. PLC-5 Typed Read - Receive Data Message

The following screen depicts a *PLC-5 Typed Read - Receive Data* message in ladder logic.

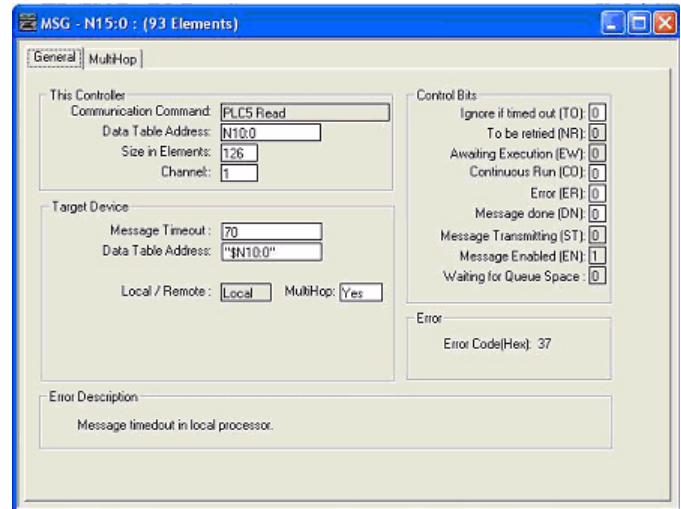


Make the following changes to the ladder logic.

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.

In the ladder logic, double-click the **Setup** 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - 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.
 - d. Note that the **Message Timeout** parameter is not actually configurable. RSLogix 500 will set the value in this box based on the EtherNet timeout settings.
 - e. Specify the port-specific receive file address for the DeviceMaster UP 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - f. Set the **MultiHop** option to **Yes**.

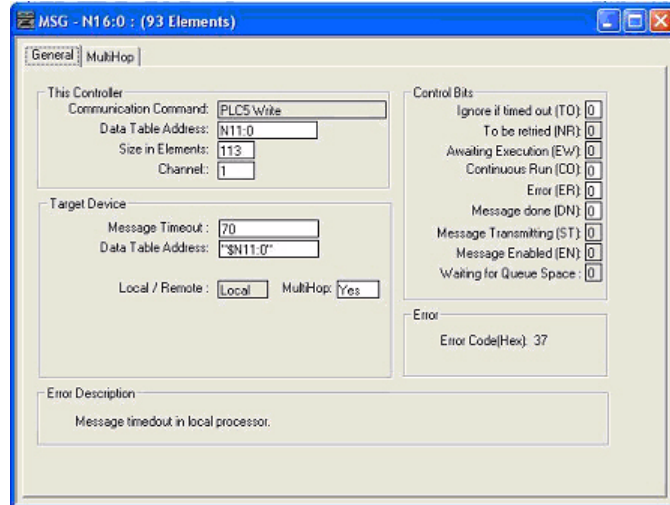


4.4.8.8. PLC-5 Typed Write - Transmit Data Message

The following screen depicts a PLC-5 Typed Write - Transmit Data 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.
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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - 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.
 - d. Note that the **Message Timeout** parameter is not actually configurable. RSLogix 500 will set the value in this box based on the EtherNet timeout settings.
 - e. Specify the port-specific transmit file address for the DeviceMaster UP 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - f. Set the **MultiHop** option to **Yes**.

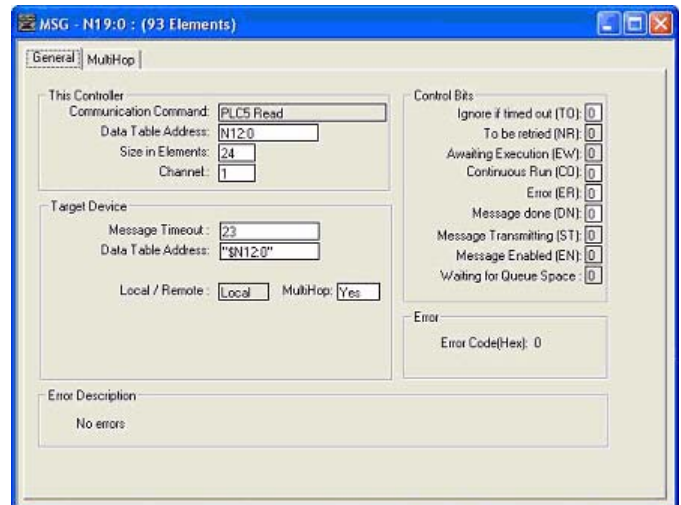
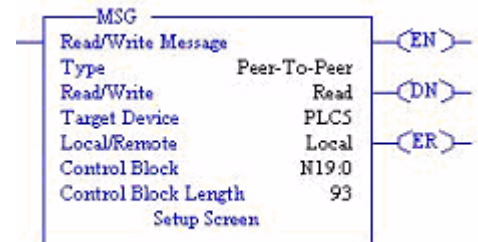


4.4.8.9. PLC-5 Typed Read - Retrieve Statistics Message

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.
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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - 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.
 - d. Note that the **Message Timeout** parameter is not actually configurable. RSLogix 500 will set the value in this box based on the EtherNet timeout settings.
 - e. Specify the port-specific statistics file address for the DeviceMaster UP 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - f. Set the **MultiHop** option to **Yes**.

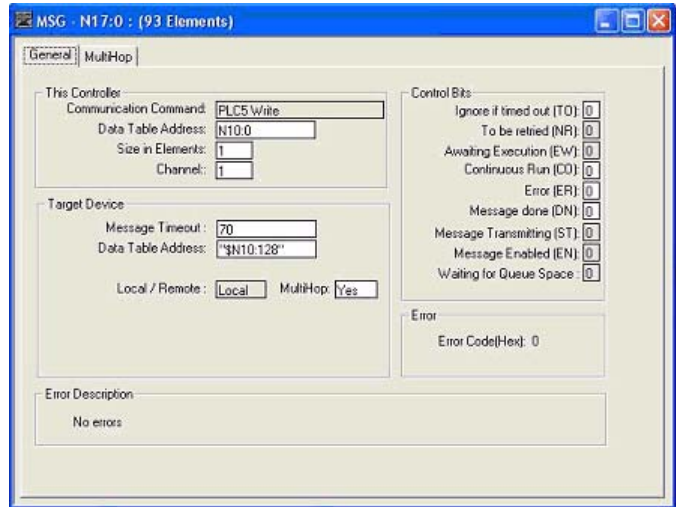
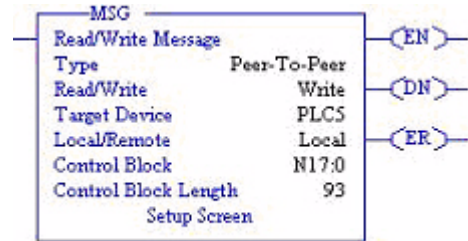


4.4.8.10. PLC-5 Typed Write - Set Receive Produced Sequence Number Message

The following screen depicts an *PLC-5 Typed Write - Set Receive Produced Sequence Number* message in ladder logic.

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.
 - 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - b. Type **1** in the **Size of Elements** box.
 - c. Set the **Channel** parameter to **1** to use the EtherNet port.
 - d. Note that the **Message Timeout** parameter is not actually configurable. RSLogix 500 will set the value in this box based on the EtherNet timeout settings.
 - e. Specify the port-specific receive sequence file address for the DeviceMaster UP 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - f. Set the **MultiHop** option to **Yes**.

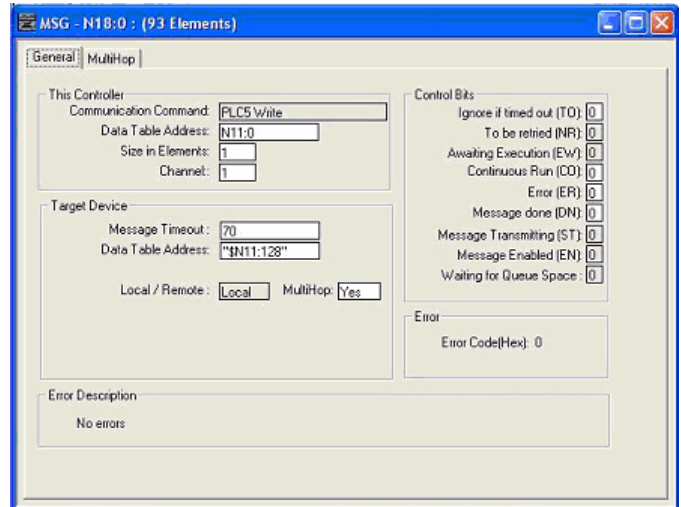
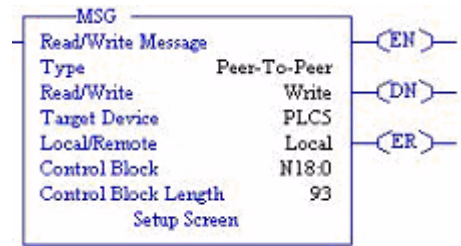


4.4.8.11. PLC-5 Typed Write - Set Transmit Produced Sequence Number Message

The following screen depicts an *PLC-5 Typed Write - Set Transmit Produced Sequence Number* message in ladder logic.

Make the following changes to the ladder logic.

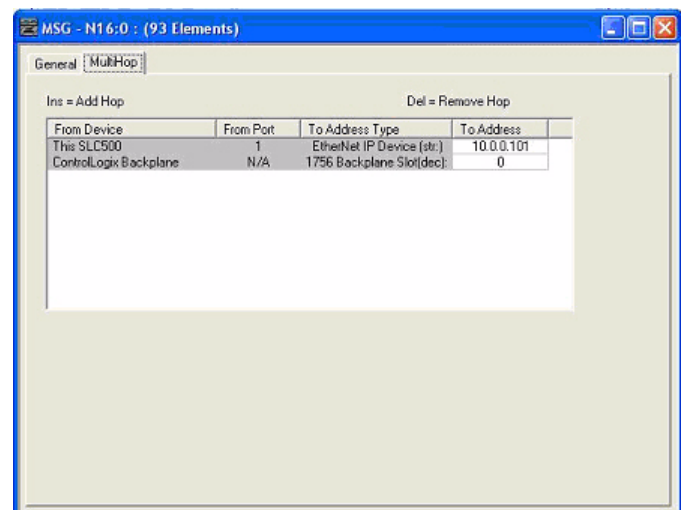
1. Select the **Write** option.
2. Select the **PLC5** option.
3. Select **Local**.
4. Assign a dedicated integer file of 93 integers to the **Control** block.
5. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
6. Make the following changes.
 - a. Specify the file address where the transmit sequence number resides in the **Data Table Address** box on **This Controller** panel. For more information on file addresses, see [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - b. Type **1** in the **Size of Elements** box.
 - c. Set the **Channel** parameter to **1** to use the EtherNet port.
 - d. Note that the **Message Timeout** parameter is not actually configurable. RSLogix 500 will set the value in this box based on the EtherNet timeout settings.
 - e. Specify the port-specific transmit sequence number file address for the DeviceMaster UP 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - f. Set the **MultiHop** option to **Yes**.



4.4.8.12. MultiHop Screen

1. Select the **MultiHop** tab on the **MSG** dialog.
2. Make the following changes.
 - a. Type the IP address for the DeviceMaster UP (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.*



4.5. PLC-5 PLC Programming Example Instructions

This topic describes how to use RSLogix 5 to configure and run the DeviceMaster UP 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 **Control/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:

- [4.5.4. *lpbkExampleSlcMsgPollRS5*](#)
- [4.5.5. *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 UP 1-port or on Port 1 of a DeviceMaster UP 4-port. Additional programming is required to use all ports on a DeviceMaster UP 4-port.*



Disclaimer: Control 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. Control 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.

4.5.1. 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.*

4.5.2. Requirements

- The EtherNet/IP firmware must be installed on the DeviceMaster UP and configured as described in the [DeviceMaster UP Hardware Installation Guide](#).
- The DeviceMaster UP 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 UP when running an example PLC program. See the [DeviceMaster UP Hardware Installation 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. Locating Updated Software and Documents](#) on Page 9, for the location of the PLC program examples.

4.5.3. 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 UP when you start the programs. However, the sequence numbers must be in sync between the PLC and DeviceMaster UP 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 UP file addresses.

4.5.4. 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 UP 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.

4.5.5. 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 UP 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.

4.5.6. 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 UP. 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.

4.5.7. 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 UP. The consumed receive sequence number is updated to match the produced receive sequence number and sent to the DeviceMaster UP 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.

4.5.8. Configure the DeviceMaster UP for the RSLogix 5 Program

The following procedure configures the DeviceMaster UP 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 3. Configuring the DeviceMaster UP](#) on Page 55.

1. Attach a loopback plug to the serial port.
2. Access the *Server Configuration* web page, using one of these methods:
 - Open PortVision Plus, right-click the DeviceMaster UP and select **Web Manager**.
 - Open a browser and type the IP address for the DeviceMaster UP in the **Address** box.
3. Click **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. Set the serial port settings under **EtherNet/IP Settings** to the following values.

| Field | Settings |
|---|---|
| TX Sequence Number Checking. | Select |
| Rx (To PLC) Ethernet Transfer Method | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • Leave blank for Polling. • Set to \$N10:0, the PLC receive filename for Write-to-File and Write-to-File-Synced. |
| Reset Port | Select |
| Save in Flash | Select |

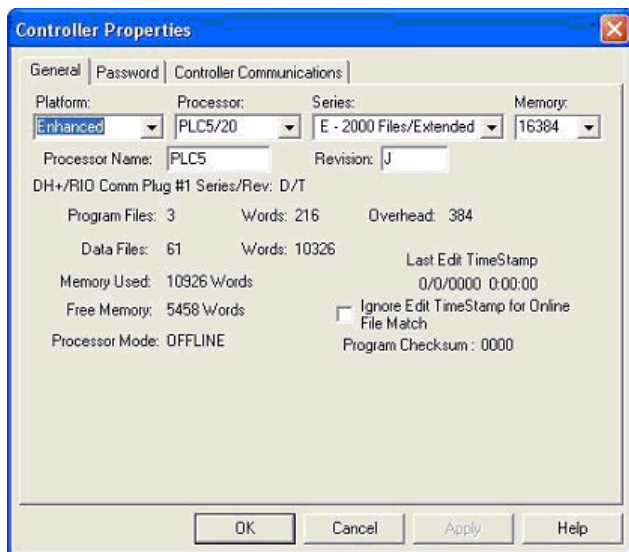
7. Click **Submit**.

4.5.9. Configure and Run the Example RSLogix 5 Program

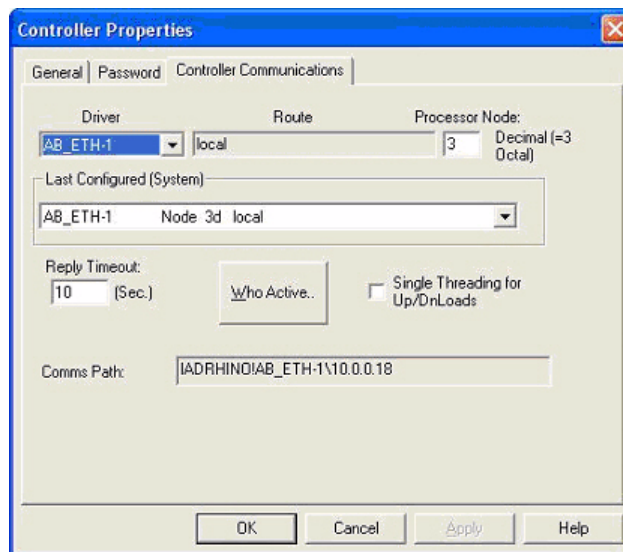
You can configure and run the RSLogix 5 example programs through RSLogix 5. For additional information on the RSLogix 5, see [4.5.10. RSLogix 5 Screen Examples](#) on Page 120.

Note: *The DeviceMaster UP must be configured for PLC-5/SLC before you can configure and download the example RSLogix 5 program. For instructions on configuring DeviceMaster UP, see [4.5.8. Configure the DeviceMaster UP 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.



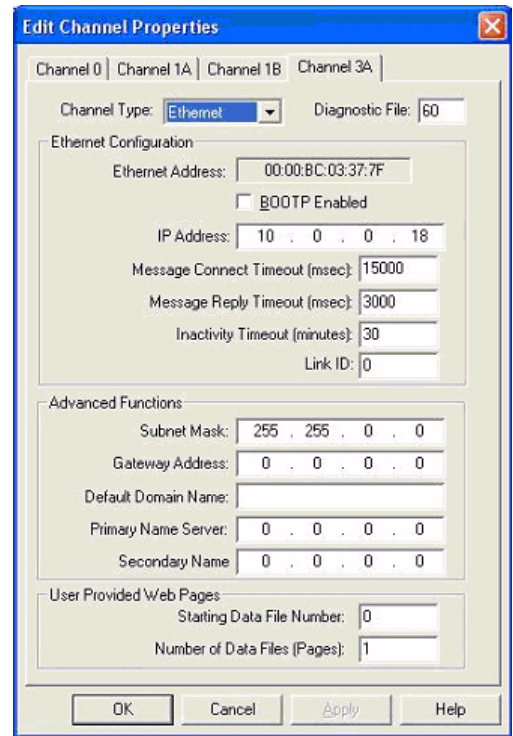
6. Select the **Controller Communications** tab and select the following options.
 - a. Set **Driver** to the appropriate type to allow RSLogix 5 to communicate with the PLC-5 processor.
 - b. Type the processor node number in the **Processor Node** box. (You may reference the **Last Configured (System)** 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.



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.

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 UP** 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.



4.5.10. RSLogix 5 Screen Examples

The following subsections explain how to configure the DeviceMaster UP through RSLogix 5. Use these screens to set up PLC and program the various messages.

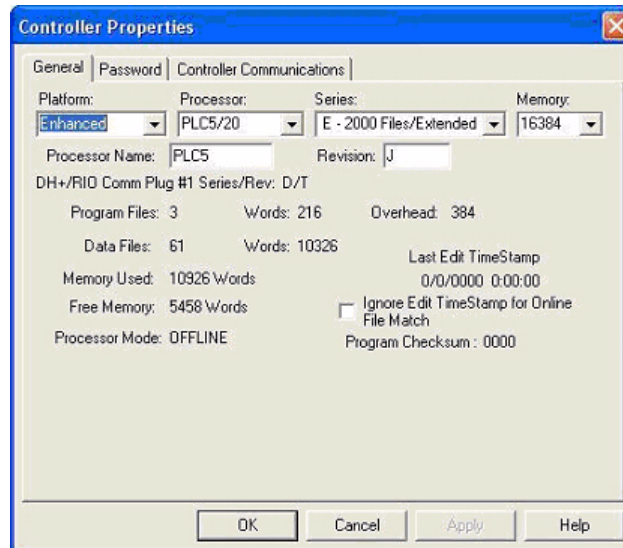
4.5.10.1. Requirements

- PLC-5 PLCs require EtherNet/IP firmware 2.01 or later running on the DeviceMaster UP.
- The PLC-5 PLC firmware must support MultiHop, ControlLogix devices and EtherNet/IP. The tables in [2.4.1. Requirements](#) on Page 46 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. Locating Updated Software and Documents](#) on Page 9 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

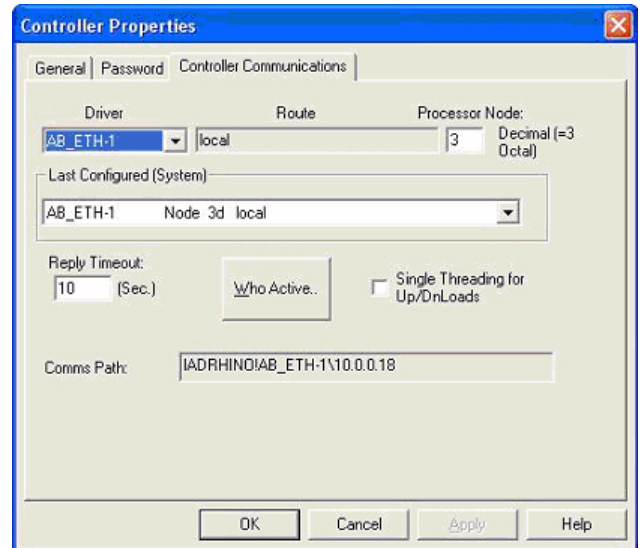
4.5.10.2. Setting up Processor and EtherNet Channel

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.

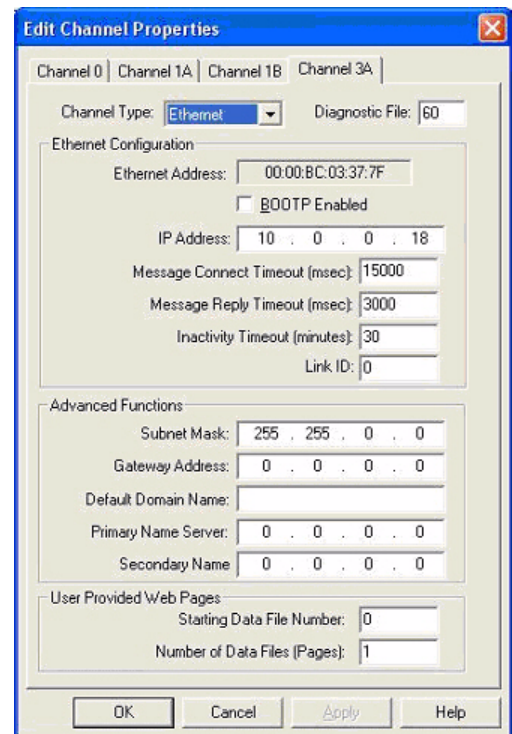


3. Select the **Controller Communications** tab and select the proper driver for RSLogix 5.
4. Click **OK** to apply your changes and close the **Controller Properties** dialog.



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.

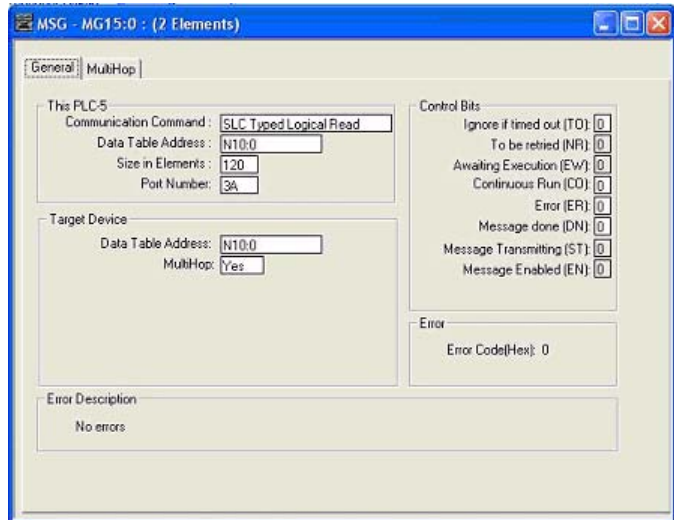
Note: The EtherNet channel may be different on your PLC-5.
6. Click **OK** to apply your changes and close the **Edit Channel Properties** dialog.



4.5.10.3. SLC Typed Read - Receive Data Message

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

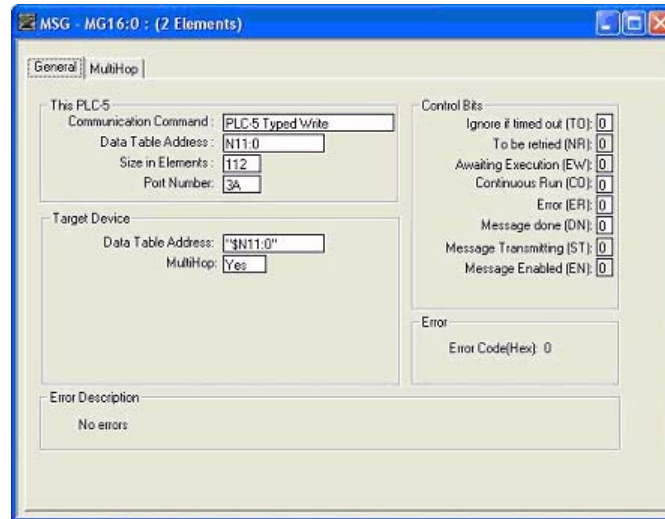
1. Assign a dedicated message file to the **Control** block in the ladder logic.
2. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
3. Make the following changes:
 - a. Specify the file address to receive data information in the **Data Table Address** box on **This PLC-5** panel. For more information on file addresses, see [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - 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.
 - d. The port number for the EtherNet channel may be different on your PLC-5.
 - e. Specify the port-specific read file address for the DeviceMaster UP in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - f. Set the **MultiHop** option to **Yes**.



4.5.10.4. 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.



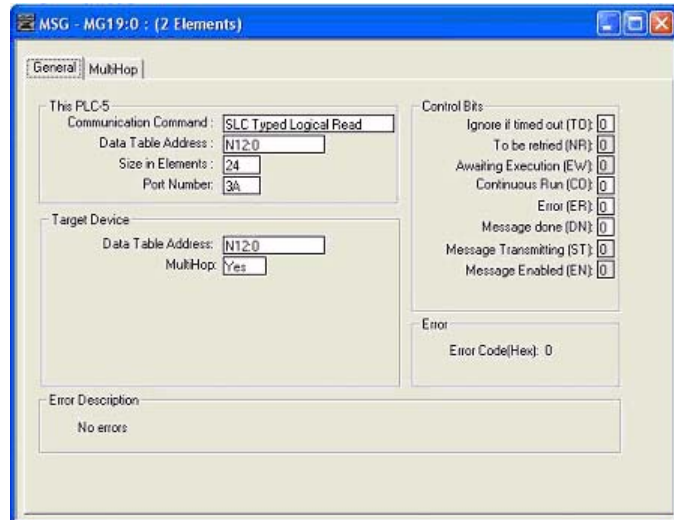
3. Make the following changes:
 - a. Specify the file address to transmit data information in the **Data Table Address** box on **This PLC-5** panel. For more information on file addresses, see [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - b. Specify a size in the **Size of Elements** 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.
 - d. The port number for the EtherNet channel may be different on your PLC-5.
 - e. Specify the port-specific transmit file address for the DeviceMaster UP in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - f. Set the **MultiHop** option to **Yes**.

4.5.10.5. 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - 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.
 - d. The port number for the EtherNet channel may be different on your PLC-5.
 - e. Specify the port-specific statistics file address for the DeviceMaster UP in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - f. Set the **MultiHop** option to **Yes**.



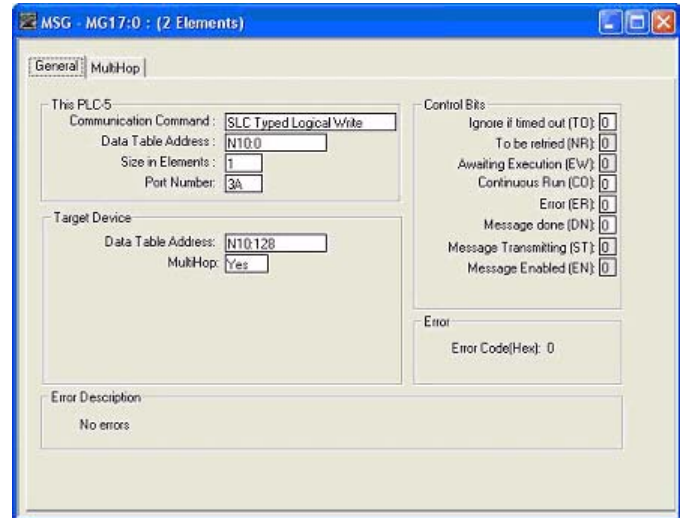
4.5.10.6. SLC Typed Write - Set Receive Produced Sequence Number Message

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.
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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - 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 UP in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
 - e. Set the **MultiHop** option to **Yes**.



4.5.10.7. 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.

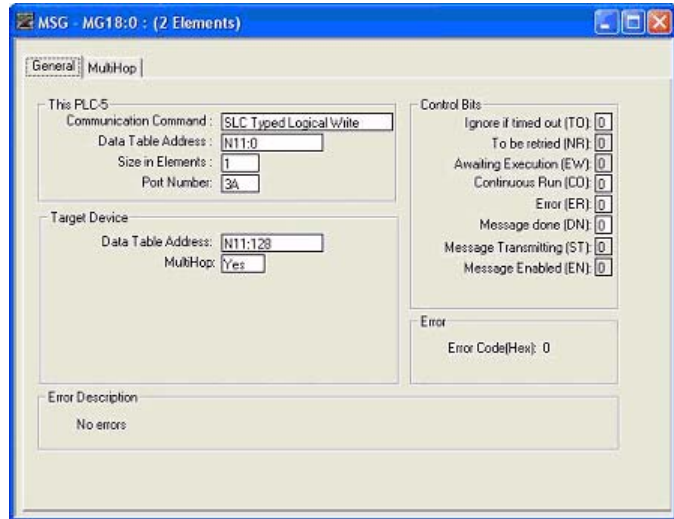
1. Assign a dedicated message file to the **Control** block in the ladder logic (as shown above).
2. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
3. Make the following changes:



- 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
- b. Type **1** in the **Size of Elements** box.
- c. Set the **Port Number** parameter to **3A** to use the EtherNet port.

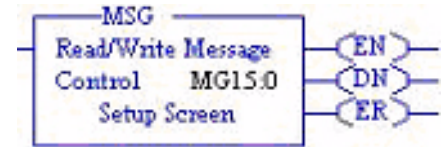
Note: *The port number for the EtherNet channel may be different on your PLC-5.*

- d. Specify the port-specific transmit sequence number file address for the DeviceMaster UP in the **Data Table Address** box on the **Target Device** panel. For more information on file addresses, see [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
- e. Set the **MultiHop** option to **Yes**.



4.5.10.8. PLC-5 Typed Read - Receive Data Message

The following screen depicts a *PLC-5 Typed Read - Receive Data* message in ladder logic.

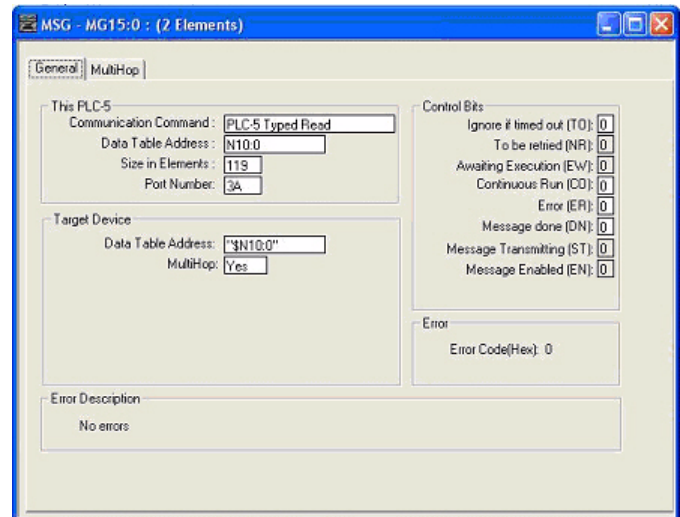


1. Assign a dedicated message file to the **Control** block in the ladder logic (as shown above).
2. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
3. Make the following changes:

- 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
- 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 UP 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
- e. Set the **MultiHop** option to **Yes**.



4.5.10.9. PLC-5 Typed Write - Transmit Data Message

The following screen depicts a *PLC-5 Typed Write - Transmit Data* message in ladder logic.

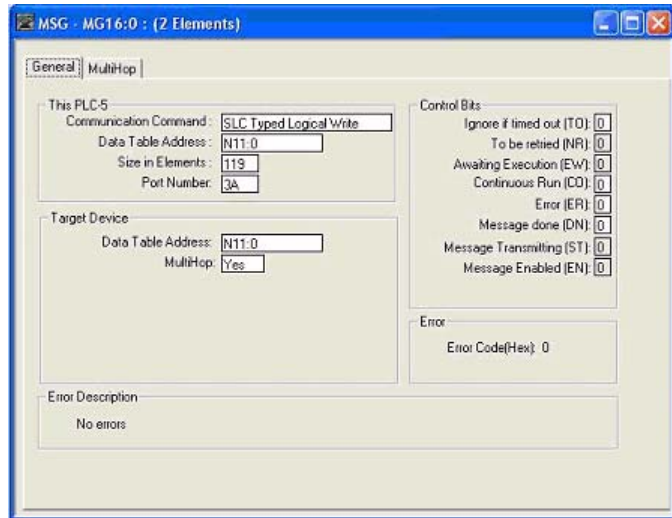
1. Assign a dedicated message file to the **Control** block in the ladder logic (as shown above).
2. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
3. Make the following changes:



- 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
- 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.*

- d. Specify the port-specific transmit file address for the DeviceMaster UP 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
- e. Set the **MultiHop** option to **Yes**.



4.5.10.10. PLC-5 Typed Read - Retrieve Statistics Message

The following screen depicts an PLC-5 Typed Read - Retrieve Statistics message in ladder logic.

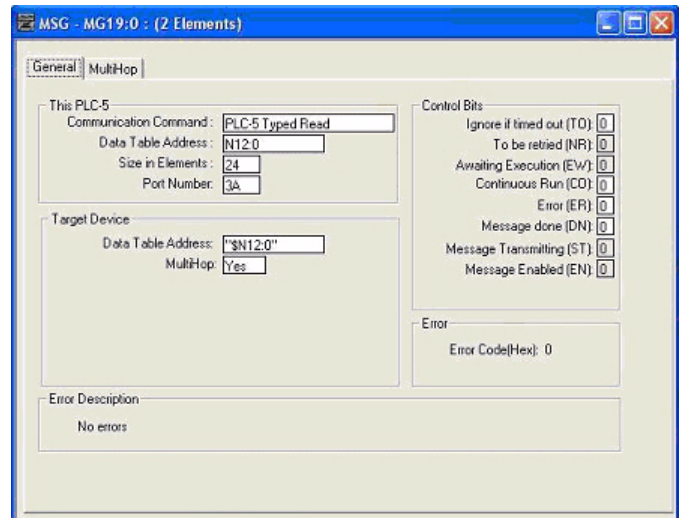


1. Assign a dedicated message file to the **Control** block in the ladder logic (as shown above).
2. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
3. Make the following changes:

- 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
- 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 UP 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
- e. Set the **MultiHop** option to **Yes**.



4.5.10.11. 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.

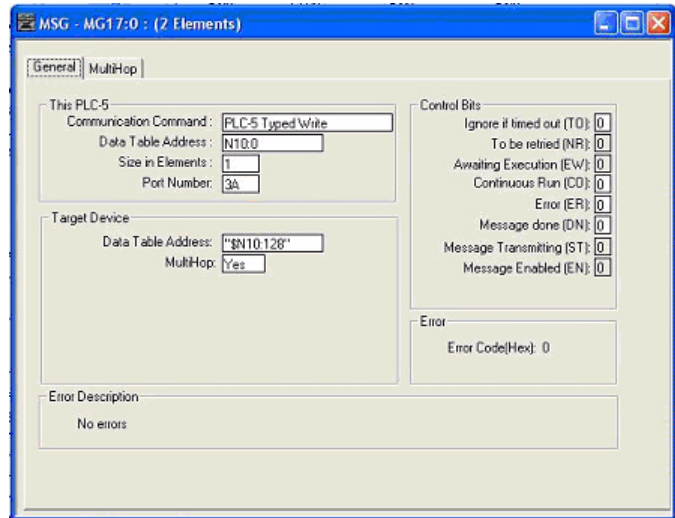


1. Assign a dedicated message file to the **Control** block in the ladder logic.
2. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
3. Make the following changes:

- 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
- 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 UP 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
- e. Set the **MultiHop** option to **Yes**.



4.5.10.12. 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.

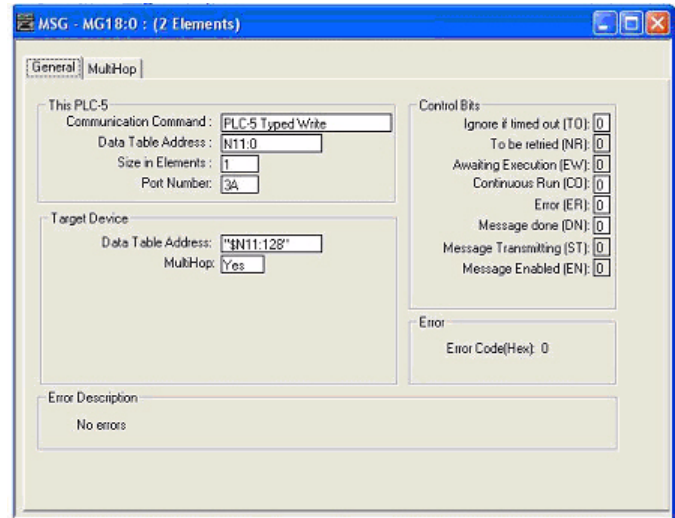


1. Assign a dedicated message file to the **Control** block in the ladder logic.
2. In the ladder logic, double-click the **Setup Screen** in the **MSG** instruction.
3. Make the following changes:

- 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
- b. Type **1** in the **Size of Elements** box.
- c. Set the **Port Number** parameter to **3A** to use the EtherNet port.

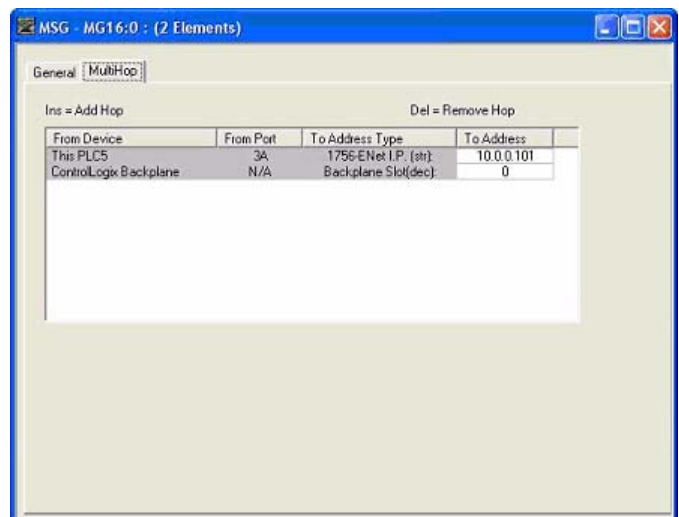
Note: *The port number for the EtherNet channel may be different on your PLC-5.*

- d. Specify the port-specific transmit sequence number file address for the DeviceMaster UP 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 [2.4.2.1. DeviceMaster UP File Addressing](#) on Page 47.
- e. Set the **MultiHop** option to **Yes**.



4.5.10.13. MultiHop Screen

1. Select the **MultiHop** tab on the **MSG** dialog.
2. Make the following changes.
 - a. Type the IP address for the DeviceMaster UP (the designated EtherNet/IP Device) in the **To Address** box.
 - b. Type **0** (zero) in the **To Address** box for the ControlLogix Backplane.



4.6. EDS Files

You do not need to add DeviceMaster UP to RSLinx for normal DeviceMaster UP-to-PLC communications. However, you can easily add the DeviceMaster UP and its associated Electronic Data Sheet (EDS) files to RSLinx.

4.6.0.1. Requirements

The following EDS files and associated icons are included in the self-installing file (.MSI) and are copied to the **Control/EtherNetIP** directory on your computer when you open the .MSI file and follow the prompts.

| File Name | Description |
|---|---------------------------------|
| DeviceMaster UP_1P_9012-x.xx.eds ^a | DeviceMaster UP 1-port |
| DeviceMaster UP_1P_9013-x.xx.eds ^a | DeviceMaster UP embedded 1-port |
| DeviceMaster UP_4P_8763-x.xx.eds ^a | DeviceMaster UP 4-port |
| DeviceMaster UP_1P-x.xx.ico ^a | DeviceMaster UP 1-port icon |
| DeviceMaster UP_4P-x.xx.ico ^a | DeviceMaster UP 4-port icon |

a. Where 9012, 9013 and 8763 indicates the model type for DeviceMaster UP and x.xx indicates the major version associated with the file.

4.6.0.2. Adding DeviceMaster UP 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 UP.

Note: A yellow question mark appears by the DeviceMaster UP(s) in the RSWho window when the associated EDS file(s) are not installed.

4.6.0.3. 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 **Control/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 UP.
 - b. Select **Change icon**.
 - c. Browse to the **Control/EtherNetIP** directory and select the icon associated with your DeviceMaster UP.

Note: You may also select your own icon stored elsewhere.
7. Click **Next** to continue.
8. Click **Finish** to exit.

4.6.0.4. Troubleshooting RSLinx

If RSLinx does not display the device after adding DeviceMaster UP 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 UP unit or units should now appear with the associated icon or icons.

Index

A

Adding

DeviceMaster UP to RSLinx [132](#)

EDS files to RSLinx [132](#)

application connection statistics (serial or socket) [72](#)

Application Specific Settings ETX Tx

Append [60](#)

Application Specific Settings Strip Rx STX/ETX [60](#)

Application Specific Settings STX Tx

Append [60](#)

Application TCP Connection

Configuration [68](#)

audience [7](#)

B

Bad Responses To Msgs Sent To PLC [73](#)

Barcode Formats [67](#)

Broadcasts Received From PLC [73](#)

BYTE [12](#)

C

changing

IP network information [75](#)

serial port parameters [57](#)

socket port parameters [61](#)

Communication window [92](#)

configuration

ports [55](#)

Connect IP Address

Application TCP Connection Configuration [68](#)

Device TCP Connection Configuration [62](#)

Connect Port

Application TCP Connection Configuration [68](#)

Device TCP Connection Configuration [62](#)

Connect To Mode

Application TCP Connection Configuration [68](#)

Device TCP Connection Configuration [62](#)

ControlLogix

EtherNet/IP interface profile [13](#)

Family, described [11](#)

D

data type definitions [12](#)

Device TCP Connection Configuration [61](#)

DeviceMaster UP

Connect to PLC

Requirement [81](#), [94](#), [114](#)

DHCP

IP configuration [75](#)

DINT [12](#)

Disable Non-Filtered TO PLC Rx Queue [64](#)

Discard Rx Packets with Errors [60](#)

Discard Unrecognized Data Mode (RFID/Barcode) [67](#)

Disconnect Mode

Application TCP Connection Configuration [69](#)

Device TCP Connection Configuration [62](#)

DWORD [12](#)

E

Edit Network Configuration page [75](#)

Edit Serial Port Configuration page [57](#)

Edit Socket Port Configuration page [61](#)

EDS files, described [132](#)

embedded web page [55](#)

Ethernet Device Configuration page [61](#)

EtherNet link object [43](#)

EtherNet/IP

application setup [10](#)

described [12](#)

EtherNet/IP Diagnostics page [74](#)

EtherNet/IP Settings Page [63](#)

ETX Rx Detect [60](#)

example instructions for

ControlLogix PLC [81](#)

PLC-5 PLC [114](#)

SLC PLC [94](#)

Example program considerations

RSLogix 5 [115](#)

RSLogix 500 [95](#)

F

Filter Age Time (Time filtered after last read) [67](#)

filtering statistics (serial or socket) [71](#)

Filtering/Data Extraction Configuration page [65](#)

First Error Description [74](#)

Flow [58](#)

Framing Error Count [70](#)

G

getting started [7](#)

Guidelines

Polling Receive Method [77](#)

Unsolicited - Write-to-Tag Receive Method [77](#)

Unsolicited - Write-to-Tag-Synced Method [77](#)

I

identity object [39](#)
Idle Timer
 Application TCP Connection Configuration [69](#)
 Device TCP Connection Configuration [62](#)
Improper Configuration Errors [74](#)
INT [12](#)
Interface profile
 ControlLogix for EtherNet/IP [13](#)
Invalid Data Items
 Discarded From Application [71](#)
 Discarded From PLC [71](#)
Invalid Message Data Errors [74](#)
Invalid Network Path Errors [73](#)
IP address
 default value [75](#)
IP Address field [56](#)
IP Config field [56](#)
IP gateway address
 default address [75](#)
IP Gateway field [56](#)
IP Netmask field [56](#)

L

Last Error Description [74](#)
Listen
 Application TCP Connection Configuration [68](#)
 Device TCP Connection Configuration [62](#)
Listen Port
 Application TCP Connection Configuration [68](#)
 Device TCP Connection Configuration [62](#)
loopbackExamplePolling.L5K [84](#)
loopbackExampleTagWrite.L5K [82](#)
loopbackExampleTagWriteSynced.L5K [83](#)
lpbkExamplePlc5MsgPollRS.SY6 [115](#)
lpbkExamplePlc5MsgPollRS5.PLC [115](#)
lpbkExamplePlc5MsgPollRS5.SY5 [115](#)
lpbkExamplePlc5MsgPollRS500.SLC [95](#)
lpbkExamplePlc5MsgPollRS500.SY5 [95](#)
lpbkExamplePlc5MsgPollRS500.SY6 [95](#)
lpbkExampleSlcMsgPollRS.SY6 [115](#)
lpbkExampleSlcMsgPollRS5.SY5 [115](#)
lpbkExampleSlcMsgPollRS500.SLC [95](#)
lpbkExampleSlcMsgPollRS500.SY5 [95](#)
lpbkExampleSlcMsgPollRS500.SY6 [95](#)
lpbkExampleSPlcMsgPollRS5.PLC [115](#)

M

Maximum PLC Update Rate (Write-To-Tag/ File) [63](#)
message router [40](#)
Messages/Responses Received From PLC [73](#)
Messages/Responses Sent To PLC [73](#)
MicroLogix, described [11](#)
MultiHop screen
 RSLogix 5 [131](#)
 RSLogix 500 [113](#)

N

No Response From PLC Errors [73](#)

O

Object
 PCCC [44](#)
object
 Ethernet link [43](#)
 identity [39](#)
 message router [40](#)
 port [41](#)
 TCP [42](#)
object definition
 serial port configuration [13](#)
 serial port data transfer object [25](#)
 serial port statistics [27](#)
 socket port data transfer [37](#)
Overrun Error Count [70](#)

P

Parity Error Count [70](#)
PCCC object [44](#)
Pending Request Limit Errors [73](#)
PLC

Algorithm
 Described [78](#)
 Polling [80](#)
 Unsolicited - Write-to-Tag [78](#)
 Unsolicited - Write-to-Tag-Synced [79](#)
 Connect to DeviceMaster UP
 Requirement [81](#), [94](#), [114](#)
 Controller Slot Number [63](#)
 Program examples
 Requirements for
 ControlLogix PLC [81](#)
 PLC-5 PLC [114](#)
 SLC or MicroLogix PLC [94](#)
 RSLogix 5000
 loopbackExamplePolling.L5K [84](#)
 loopbackExampleTagWrite.L5K [82](#)
 loopbackExampleTagWriteSynced.L5K [83](#)
 Programming
 example instructions for
 ControlLogix PLC [81](#)
 PLC-5 PLC [114](#)
 SLC PLC [94](#)
 programming overview [77](#)
 Requirements for
 ControlLogix PLC [81](#)
 PLC-5 PLC [114](#)
 SLC or MicroLogix PLC [94](#)
 RSLogix 5000 program examples
 Modifying for older versions of RSLogix 5000 [93](#)
PLC Interface Diagnostics page [73](#)
PLC IP address [63](#)
PLC Specific Settings ETX Tx Append [60](#)

PLC Specific Settings Strip Rx STX/ETX [60](#)

PLC Specific Settings STX Tx Append [60](#)

PLC-5

described [11](#)

PCCC object [44](#)

receive data message [48](#)

Requirements [46](#)

retrieve statistics message [50](#)

sequence number messages [49](#)

transmit data message [49](#)

Typed Read

Receive Data Message

RSLogix 5 [127](#)

RSLogix 500 [109](#)

Retrieve Statistics Message

RSLogix 5 [129](#)

RSLogix 500 [111](#)

Typed Write

Set Receive Produced Sequence Number Message

RSLogix 5 [130](#)

RSLogix 500 [112](#)

Set Transmit Produced Sequence Number Message

RSLogix 5 [131](#)

RSLogix 500 [113](#)

Transmit Data Message

RSLogix 5 [128](#)

RSLogix 500 [110](#)

PLC-5/SLC Tx MS Byte First [64](#)

Polling Receive Method

Guidelines [77](#)

loopbackExamplePolling.L5K [84](#)

PLC algorithm [80](#)

polling receive method [20](#)

port

changing

serial parameters [57](#)

socket parameters [61](#)

saving options [69](#)

port object [41](#)

Programming

example instructions for

ControlLogix PLC [81](#)

PLC-5 PLC [114](#)

SLC PLC [94](#)

PLC

Requirements for

ControlLogix PLCs [81](#)

SLC or MicroLogix PLC [94](#)

programming

overview [77](#)

programming interface [11](#)

R

Request data from DeviceMaster UP [89](#)

Request Messages From PLC [73](#)

Request statistics from DeviceMaster UP [91](#)

Requirements for

Connecting

DeviceMaster UP to PLC

RSLogix 5 [114](#)

RSLogix 500 [94](#)

RSLogix 5000 [81](#)

PLC-5/SLC [46](#)

RSLink [132](#)

RSLogix 5 PLC program examples [114](#)

RSLogix 500 PLC program examples [94](#)

RSLogix 5000 PLC program examples [81](#)

Setting up processor and Ethernet channel [120](#)

Reset Port [69](#)

Reset Statistics [70](#)

Reset statistics [69](#)

RFID Antenna Grouping [66](#)

RFID Reader Interface Type [67](#)

RFID Tags With Unknown Formats [71](#)

RSLink

Adding DeviceMaster UP [132](#)

Adding EDS files [132](#)

Requirements [132](#)

Troubleshooting [133](#)

RSLogix 5

Configure and run example program [118](#)

Described [114](#)

Example program considerations [115](#)

Requirements [114](#)

Screen examples [120](#)

RSLogix 500

Configure and run example program [98](#)

Described [94](#)

Example program considerations [95](#)

Requirements [94](#)

Screen examples [101](#)

RSLogix 5000

Described [81](#)

Modifying PLC program for older version of

RSLogix 5000 [93](#)

Screen examples [86](#)

RX (To PLC) EtherNet/IP Transfer Method [63](#)

RX (To PLC) Produced Data Tag Name [64](#)

RX Byte Count [70](#)

Application Connection Statistics [72](#)

Socket Interface Statistics [72](#)

RX Con Sequence Error Count [71](#)

RX Pkt Count [70](#)

Application Connection Statistics [72](#)

Socket Interface Statistics [72](#)

RX Timeout Between Packets [58](#)

S**Save in Flash** [69](#)**saving port options** [69](#)**Screen examples**

MultiHop screen

 RSLogix 5 [131](#) RSLogix 500 [113](#)

PLC-5

Typed Read

Receive Data Message

 RSLogix 5 [127](#) RSLogix 500 [109](#)

Retrieve Statistics Message

 RSLogix 5 [129](#) RSLogix 500 [111](#)

Typed Write

 Set Receive Produced Sequence Number Mes-
 sage RSLogix 5 [130](#) RSLogix 500 [112](#) Set Transmit Produced Sequence Number
 Message RSLogix 5 [131](#) RSLogix 500 [113](#)

Transmit Data Message

 RSLogix 5 [128](#) RSLogix 500 [110](#)RSLogix 5 [120](#)RSLogix 500 [101](#)RSLogix 5000 [86](#) Communication window [92](#) Request data from DeviceMaster UP [89](#) Request statistics from DeviceMaster UP [91](#) Send configuration to DeviceMaster UP [87](#) Send consumed sequence number to DeviceMas-
 ter UP [90](#) Transmit data to DeviceMaster UP [86](#)

SLC

Typed Read

Receive Data Message

 RSLogix 5 [122](#) RSLogix 500 [104](#)

Retrieve Statistics Message

 RSLogix 5 [124](#) RSLogix 500 [106](#)

Typed Write

 Set Receive Produced Sequence Number Mes-
 sage RSLogix 5 [125](#) RSLogix 500 [107](#) Set Transmit Produced Sequence Number
 Message RSLogix 5 [126](#) RSLogix 500 [108](#)

Typed Write (cont.)

Transmit Data Message

 RSLogix 5 [123](#) RSLogix 500 [105](#)**Send configuration to DeviceMaster UP** [87](#)**Send consumed sequence number to Device-
Master UP** [90](#)**Serial Device Configuration Page** [57](#)**Serial number field** [56](#)**Serial Packet Identification Settings** [59](#)**serial port** application TCP connection configuration [68](#) configuration [55](#) configuration object definition (70 hex) [13](#) data transfer object definition (71 hex) [25](#) EtherNet/IP settings [63](#) filtering/data extraction configuration [65](#) settings [57](#) statistics object definition (72 hex) [27](#)**Server Configuration** [56](#)**Setting up**

Processor and Ethernet channel

 RSLogix 5 screen example [120](#) RSLogix 500 screen examples [101](#)**settings** serial port [57](#) socket port [61](#)**SLC** described [11](#) PCCC object [44](#) receive data message [48](#) Requirements [46](#) retrieve statistics message [50](#) sequence number messages [49](#) transmit data message [49](#)

Typed Read

Receive Data Message

 RSLogix 5 [122](#) RSLogix 500 [104](#)

Retrieve Statistics Message

 RSLogix 5 [124](#) RSLogix 500 [106](#)

Typed Write

 Set Receive Produced Sequence Number Mes-
 sage RSLogix 5 [125](#) RSLogix 500 [107](#) Set Transmit Produced Sequence Number Mes-
 sage RSLogix 5 [126](#) RSLogix 500 [108](#)

Transmit Data Message

 RSLogix 5 [123](#) RSLogix 500 [105](#)**Socket Interface Statistics** [72](#)

socket port
application TCP connection configuration [68](#)
configuration [55](#)
data transfer object definition (74 hex) [37](#)
EtherNet/IP settings [63](#)
filtering/data extraction configuration [65](#)
settings [61](#)

software
version [56](#)

Static
IP configuration [75](#)

STRING [12](#)

STX Rx Detect [59](#)

Submit [69](#)

subnet mask
default value [75](#)

System Resource Errors [74](#)

T

TCP object [42](#)

To APP Dropped Packet Count [72](#)

To Application Dropped Packet Count

Socket Interface Statistics [72](#)

To Application Filter Mode [66](#)

To Application Filtering Options (RFID Only) [66](#)

To Application Filtering Options (RFID/Barcode) [66](#)

To Device Dropped Packet Count
Application Connection Statistics [72](#)

To PLC Dropped Packet Count [70](#)

Socket Interface Statistics [72](#)

To PLC Filter Mode [65](#)

To PLC Filtering Options (RFID Only) [65](#)

To PLC Filtering Options (RFID/Barcode) [65](#)

Transmit

Data to DeviceMaster UP
RSLogix 5000 [86](#)

Troubleshooting

RSLinx [133](#)

TX Byte Count [70](#)

Application Connection Statistics [72](#)

Socket Interface Statistics [72](#)

Tx Duplicate Seq Errors [71](#)

TX Pkt Count [70](#)

Application Connection Statistics [72](#)

Socket Interface Statistics [72](#)

Tx Sequence Number Checking [64](#)

Tx Unexpected Seq Errors [71](#)

U

UDINT [12](#)

UINT [12](#)

Undo Changes [69](#)

Unexpected Event Errors [74](#)

Unsolicited - Write-to-Tag Receive Method

Guidelines [77](#)

loopbackExampleTagWrite.L5K [82](#)

PLC algorithm [78](#)

Unsolicited - Write-to-Tag Receive method [19](#)

Unsolicited - Write-to-Tag-Synced Receive Method

Guidelines [77](#)

loopbackExampleTagWriteSynced [83](#)

PLC algorithm [79](#)

Unsolicited - Write-to-Tag-Synced Receive method [19](#)

Unsupported CIP Request Attribute Errors [74](#)

Unsupported CIP Request Class Errors [74](#)

Unsupported CIP Request Instance Errors [74](#)

Unsupported CIP Request Service Errors [74](#)

UPC/EAN Standard 12-14 Digit [67](#)

UPD/EAN Eight Digit [67](#)

USINT [12](#)

V

Valid Data Items

Filtered From Application [71](#)

Filtered From PLC [71](#)

Sent To Application [71](#)

Sent To PLC [71](#)

W

web page

configuration [55](#)

Edit Network Configuration [75](#)

serial port settings [57](#)

socket port settings [61](#)

WORD [12](#)

