IO-Link

IO-Link Master EIP-4

EtherNet/IP Reference Manual

COMTROL®

Trademark Notices

Microsoft and Windows are registered trademarks of Microsoft Corporation.

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

First Edition, March 12, 2014

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

Document Number: 2000589 Rev. A

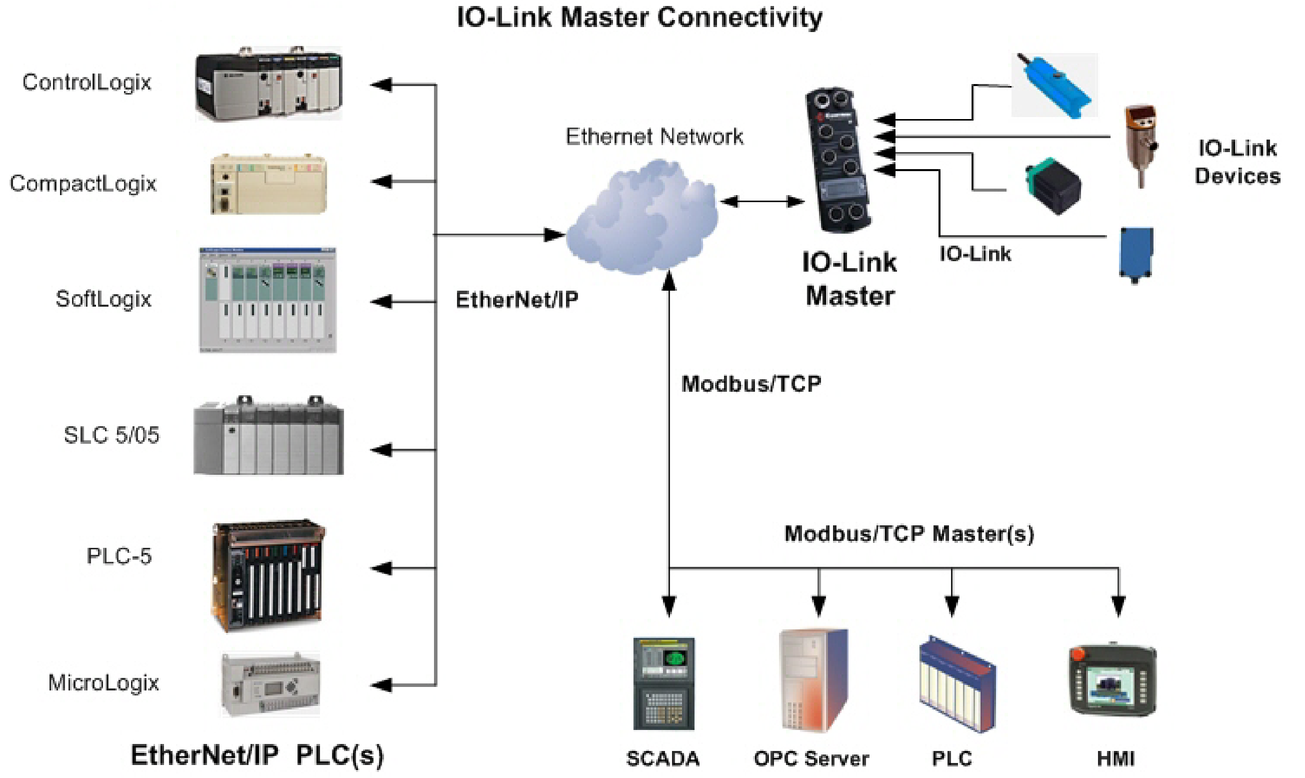
Table of Contents

Chapter 1. EtherNet/IP Interface

1.1. Introduction

This section is intended to describe the EtherNet/IP and Modbus/TCP interfaces provided by the IO-Link Master.

These interfaces provide the ability to retrieve port and device status information, input and output process data and access to IO-Link device ISDU (SPDU) data blocks.



1.1.1. Functionality Summary

The EtherNet/IP interface consists of:

- Input Process Data blocks that include:
  - Port communication status
  - PDI valid status
  - Auxiliary Input status (pin 2 of IO-Link connector)
  - The active event code (zero if no active event)
  - The input process data received from the port. This may be
    - IO-Link mode: IO-Link device input process data
    - I/O Input mode: Input bit status
    - I/O Output mode: Output bit status (configurable option)

- Output Process Data blocks that include:
  - The active event code to clear ((configurable option)
  - The output process data to be sent to the port. This may be
    - IO-Link mode: IO-Link device output process data
    - I/O Output mode: Output bit status
- ISDU (ISDU) interface:
  - Provides single and nested batch read/write capabilities
  - Requires use of MSG instructions
  - Provides both blocking and non-blocking message capabilities
    - Blocking message responses are not returned until all the ISDU command(s) have completed.
    - Non-blocking messages return immediately. The PLC must then request the ISDU command(s) response status until a valid response is returned.
- Web based configuration and diagnostic pages:
  - IO-Link interface configuration and diagnostics
  - EtherNet/IP interface configuration and diagnostics
- EtherNet/IP interface support for ControlLogix, SLC, MicroLogix, and PLC-5 PLC families.
- Modbus/TCP slave interface.
- Example PLC programs to aid the PLC programmer.

1.1.2. Data Type Definitions

The following data type definitions apply.

Data Type Definitions	
BOOL	Boolean; TRUE if = 1; False if = 0
USINT	Unsigned Short Integer (8 bit)
CHAR	Character (8 bit)
SINT	Short Integer (8 bit)
UINT	Unsigned Integer (16 bit)
INT	Signed Integer (16 bit)
UDINT	Unsigned Double Integer (32 bit)
DINT	Signed Double Integer (32 bit)
STRING	Character String (1 byte per character)
BYTE	Bit String (8 bit)
WORD	Bit String (16 bits)
DWORD	Bit String (32 bits)

1.1.3. Terms and Definitions

This section uses the following terms and definitions.

Term	Definition
Class 1	Otherwise called **implicit** messaging, is a method of communication between EtherNet/IP controllers and devices that: • Uses Ethernet UDP messages. • Is cyclic in nature. Input and/or output data is exchanged between the controllers and devices at regular time intervals.
Class 3	Otherwise called **explicit** messaging, is a method of communication between EtherNet/IP controllers and devices that: • Uses Ethernet TCP/IP messages. • By itself is not cyclic in nature. The controller and devices must send individual messages to each other.
EtherNet/IP	An Ethernet based industrial communication protocol utilized to communicate between controllers, often times PLCS, and devices.
Ethernet TCP/IP	Standard Ethernet communications protocol utilizing socket communication interfaces that **guarantees delivery** to the intended device.
Ethernet UDP/IP	Standard Ethernet communications protocol utilizing socket communication interfaces that **does not guarantee delivery**. The data may or may get to the intended device.
IO-Link Master	IO-Link gateway that provides communication between IO-Link devices and Ethernet protocols such as EtherNet/IP and Modbus/TCP.
Multicast	Multicast addressing involves Ethernet devices sending messages to each other using a multicast address. Multicast addressing: • Uses a specified IP address range designated for multicast communication. • Allows either one or multiple devices to receive the same messages.
Point-to-Point	Point-to-Point, otherwise called **unicast**, addressing involves Ethernet devices sending messages directly to each other using their own IP addresses. Messages are sent to only one device.
PDI data (Process Data Input)	Process data received from an IO-Link device or I/O interface that can be provided to external controllers such as PLCs, HMIs, SCADA, and OPC Servers.
PDO data (Process Data Output)	Process data received from external controllers such as PLCs, HMIs, SCADA, and OPC Servers and sent to an IO-Link device or I/O interface. ***Note:*** *IO-Link devices may or may not support PDO data.*
ISDU	Service Process Data Unit. Otherwise called ISDU, refers to the Service Data units on IO-Link devices that are used for information, status and configuration settings.
ISDU	Indexed Service Data Unit. Otherwise called ISDU, refers to the Service Data units on IO-Link devices that are used for information, status and configuration settings.
Class 1	Otherwise called **implicit** messaging, is a method of communication between EtherNet/IP controllers and devices that: • Uses Ethernet UDP messages. • Is cyclic in nature. Input and/or output data is exchanged between the controllers and devices at regular time intervals.

1.2. Data Transfer Methods

The IO-Link Master provides a selection of process data transfer methods and a number of options to customize the process data handling.

- *1.2.1. Receive Process Data Methods*
- *1.2.2. Transmit Process Data Methods* on Page 11

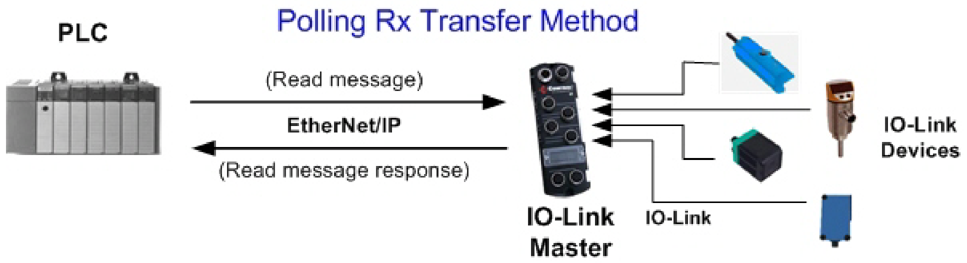
1.2.1. Receive Process Data Methods

The IO-Link Master supports the following receive process data methods:

- *1.2.1.1. Polling-PLC Requests Data*
- *1.2.1.2. Write-to-Tag/File-IO-Link Master Writes Data Directly Into PLC Memory*
- *1.2.1.3. Class 1 Connection (Input Only)-PLC and IO-Link Master Utilize an I/O Connection*

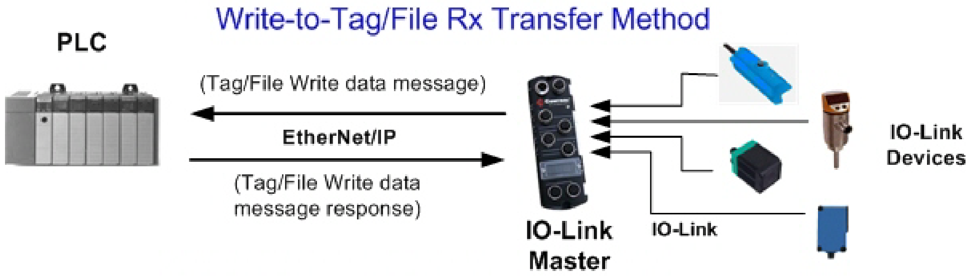
1.2.1.1. Polling-PLC Requests Data

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



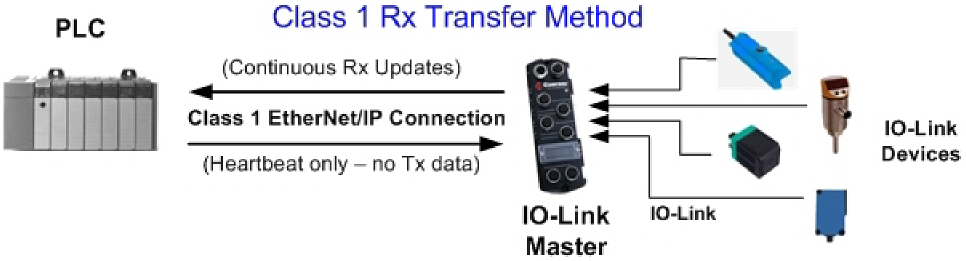
1.2.1.2. Write-to-Tag/File-IO-Link Master Writes Data Directly Into PLC Memory

Also called *Master-Mode* for some industrial protocols, the Write-to-Tag/File method requires the IO-Link Master to send messages that write data directly into a tag or file on the PLC. The IO-Link Master sends changed data to the PLC immediately and, optionally, can be configured to also send "heartbeat" update messages at a regular time interval.


Write-to-Tag/File Rx Transfer Method
PLC
(Tag/File Write data message)
EtherNet/IP
(Tag/File Write data message response)
IO-Link Master
IO-Link
IO-Link Devices

1.2.1.3. Class 1 Connection (Input Only)-PLC and IO-Link Master Utilize an I/O Connection

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


Class 1 Rx Transfer Method
PLC
(Continuous Rx Updates)
Class 1 EtherNet/IP Connection
(Heartbeat only – no Tx data)
IO-Link Master
IO-Link
IO-Link Devices

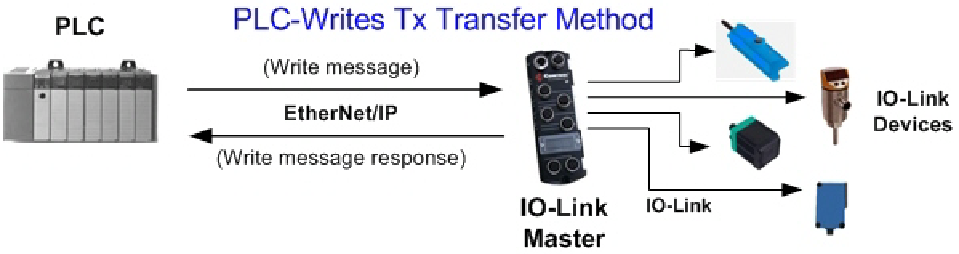
1.2.2. Transmit Process Data Methods

The IO-Link Master supports the following transmit process data methods:

- *1.2.2.1. PLC-Writes*
- *1.2.2.2. Read-from-Tag/File-IO-Link Master Reads Data from PLC Memory*
- *1.2.2.3. Class 1 Connection (Input and Output)-PLC and IO-Link Master Utilize an I/O Connection*

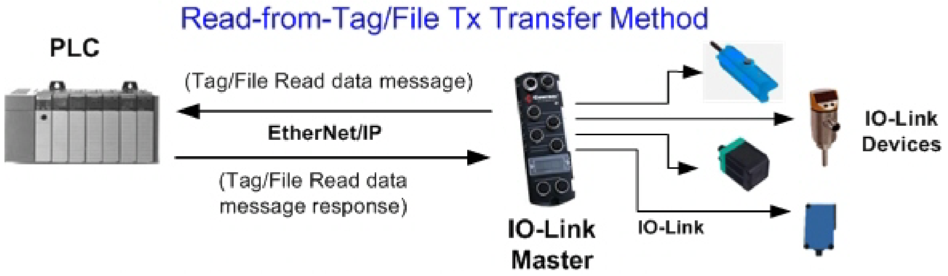
1.2.2.1. PLC-Writes

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



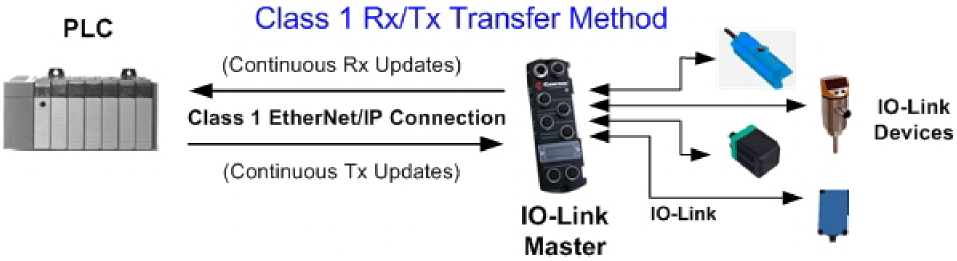
1.2.2.2. Read-from-Tag/File-IO-Link Master Reads Data from PLC Memory

Also called *Master-Mode* for some industrial protocols, the Read-from-Tag/File method requires the IO-Link Master to read data from a tag or file on the PLC. In this method, the IO-Link Master requests data from the PLC at configurable time intervals.


Read-from-Tag/File Tx Transfer Method
PLC
(Tag/File Read data message)
EtherNet/IP
(Tag/File Read data message response)
IO-Link Master
IO-Link
IO-Link Devices

1.2.2.3. Class 1 Connection (Input and Output)-PLC and IO-Link Master Utilize an I/O Connection

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



1.3. Process Data Block Descriptions

This subsection discusses the following:

- *1.3.1. Input Process Data Block Description*
- *1.3.2. Output Process Data Block Description* on Page 15

1.3.1. Input Process Data Block Description

The Input Process Data Block format is dependent on the configured PDI Data Format. The following tables describe the Input Process Data Block in the possible formats.

Parameter Name	Data Type	Description
Port Status	BYTE	The status of the IO-Link device. **Bit 0 (0x01):** 0 = IO-Link port communication initialization process is inactive 1 = IO-Link port communication initialization process is active **Bit 1 (0x02):** 0 = IO-Link port communication is not operational 1 = IO-Link port communication is operational **Bit 2 (0x04):** 0 = IO-Link input process data is not valid. 1 = IO-Link input process data is valid. **Bit 3 (0x08):** 0= No fault detected 1= Fault detected • A minor communication fault is indicated by the Operational status bit being set to 1. A minor communication fault results from: - A temporary loss of communication to the IO-Link device. - A recoverable IO-Link Master software or hardware fault. • A major communication fault is indicated by the Operational bit being set to 0. - An unrecoverable loss of communication to the IO-Link device. - An unrecoverable IO-Link Master software or hardware fault. **Bits 4-7:** Reserved (0)
Auxiliary I/O	BYTE	Auxiliary I/O: **Bit 0**: The status of the Pin 2 auxiliary bit. 0 = off 1 = on **Bits 1-7**: Reserved (0)
Event Code	INT	16-bit event code received from the IO-Link device.
PDI Data *Default Length* = 32 bytes	Array of up to 32 BYTEs	The PDI data as received from the IO-Link device. May contain from 0 to 32 bytes of PDI data. The definition of the PDI data is device dependent. ***Note:*** *Length is configurable using the web page interface.*

1.3.1.1. Input Process Data Block-8 Bit Data Format

The following table provides detailed information about the Input Process Data Block-8 Bit data format.

Byte	Bit 7 Bit 0
0	Port Status
1	Auxiliary I/O
2	Event Code LSB
3	Event Code MSB
4	PDI Data Byte 0
5	PDI Data Byte 1
..	..
..	..
N+3	PDI Data Byte (N-1)

1.3.1.2. Input Process Data Block-16 Bit Data Format

The following table provides detailed information about the Input Process Data Block-16 data format.

Word	Bit 15 Bit 8	Bit 7 Bit 0
0	Port Status	Auxiliary I/O
1	Event Code	
2	PDI Data Word 0	
3	PDI Data Word 1	
..	..	
..	..	
N+1	PDI Data Word (N-1)	

1.3.1.3. Input Process Data Block-32 Bit Data Format

The following table provides detailed information about the Input Process Data Block-32 Bit data format.

Long Word	Bit 31 Bit 24	Bit 23 Bit 16	Bit 15 Bit 0
0	Port Status	Auxiliary I/O	Event Code
2	PDI Data Long Word 0		
3	PDI Data Long Word 1		
..	..		
N	PDI Data Long Word (N-1)		

1.3.2. Output Process Data Block Description

The contents of the Output Process Data Block are configurable.

Parameter Name	Data	Description
Event to Clear (Configurable option) *Default*: Not included	INT	If included, allows clearing of 16-bit event code received in the PDI data block via the PDU data block.
PDO Data *Default Length* = 32 bytes	Array of up to 32 BYTEs	The PDO data written to the IO-Link device. May contain from 0 to 32 bytes of PDO data. The definition and length of the PDO data is device dependent. ***Note:*** *Length is configurable via web page interface.*

1.3.2.1. Output Process Data Block-8 Bit (SINT) Data Format

With the Include **Event to Clear** option selected:

Byte	Bit 7 Bit 0
0	Event Code LSB
1	Event Code MSB
2	PDO Data Byte 0
3	PDO Data Byte 1
..	..
..	..
N+1	PDO Data Byte (N-1)

Without the **Event to Clear** option selected:

Byte	Bit 7 Bit 0
0	PDO Data Byte 0
1	PDO Data Byte 1
..	..
..	..
N-1	PDO Data Byte (N-1)

1.3.2.2. Output Process Data Block-16 Bit (INT) Data Format

With the **Include Event to Clear** option selected:

Word	Bit 15 Bit 0
0	Event Code
1	PDO Data Word 0
2	PDO Data Word 1
..	..
..	..
N	PDO Data Word (N-1)

Without the **Event to Clear** option selected:

Word	Bit 15 Bit 0
0	PDO Data Word 0
1	PDO Data Word 1
..	..
..	..
N-1	PDO Data Word (N-1)

1.3.2.3. Output Process Data Block-32 Bit (DINT) Data Format

With the **Include Event to Clear** option selected:

Long Word	Bit 31 Bit 16	Bit 15 Bit 0
0	0	Event Code
1	PDO Data Long Word 0	
2	PDO Data Long Word 1	
..	..	
..	..	
N - 1	PDO Data Long Word (N-1)	

Without the **Event to Clear** option selected:

Long Word	Bit 31 Bit 0
0	PDO Data Long Word 0
1	PDO Data Long Word 1
..	..
..	..
N - 1	PDO Data Long Word (N-1)

Chapter 2. Functionality Descriptions

This appendix discusses the following:

- *2.1. Event Handling*
- *2.2. ISDU Handling* on Page 21

2.1. Event Handling

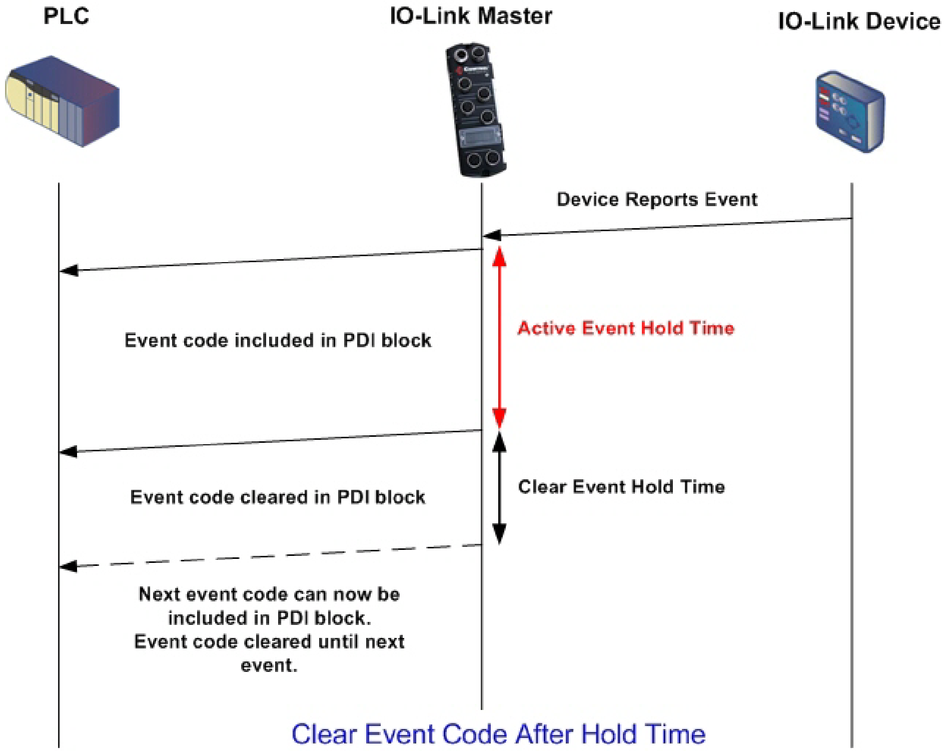
The IO-Link Master event handling is designed to provide real-time updates of event codes received directly from the IO-Link device. The IO-Link event code:

- Is included in the second 16-bit word of the Input Process Data (PDI) block.
    - An active event is indicated by a non-zero value.
    - Inactive or no event is indicated by a zero value.
- Two methods are provided to clear an event:
    - Enable the *Clear Event After Hold Time* option.
    - The IO-Link Master keeps, or holds, the active event code in the PDI block until the configured *Active Event Hold Time* has passed.
    - The IO-Link Master then clears the event code in the PDI block and waits until the *Clear Event Hold Time* has passed before including another event code in the PDI block.
    - Enable the *Clear Event In PDO Block* option.
        - The IO-Link Master monitors the PDO block received from the PLC.
        - The IO-Link Master expects the first entry of the PDO block to indicate an event code to be cleared.
        - If there is an active event code in the PDI block and the PDO block both contain the same event code, the event code is cleared in the PDI block.
        - The IO-Link Master then clears event code in the PDI block and waits until the *Clear Event Hold Time* has passed before including another event code in the PDI block.
- The two methods can be used separately or together to control clearing of events.

The next subsections illustrate the event clearing process for the various event configurations.

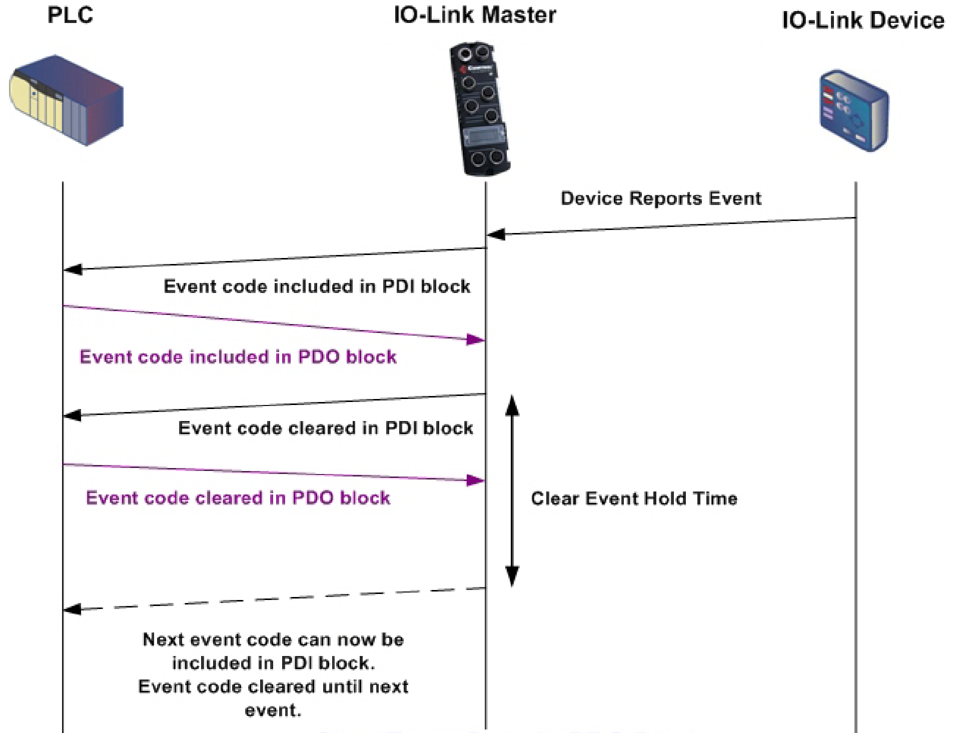
2.1.1. Clear Event After Hold Time Process

This illustrates clearing the event after the hold time process.


PLC
IO-Link Master
IO-Link Device
Device Reports Event
Active Event Hold Time
Event code included in PDI block
Clear Event Hold Time
Event code cleared in PDI block
Next event code can now be included in PDI block.
Event code cleared until next event.
Clear Event Code After Hold Time

2.1.2. Clear Event in PDO Block Process

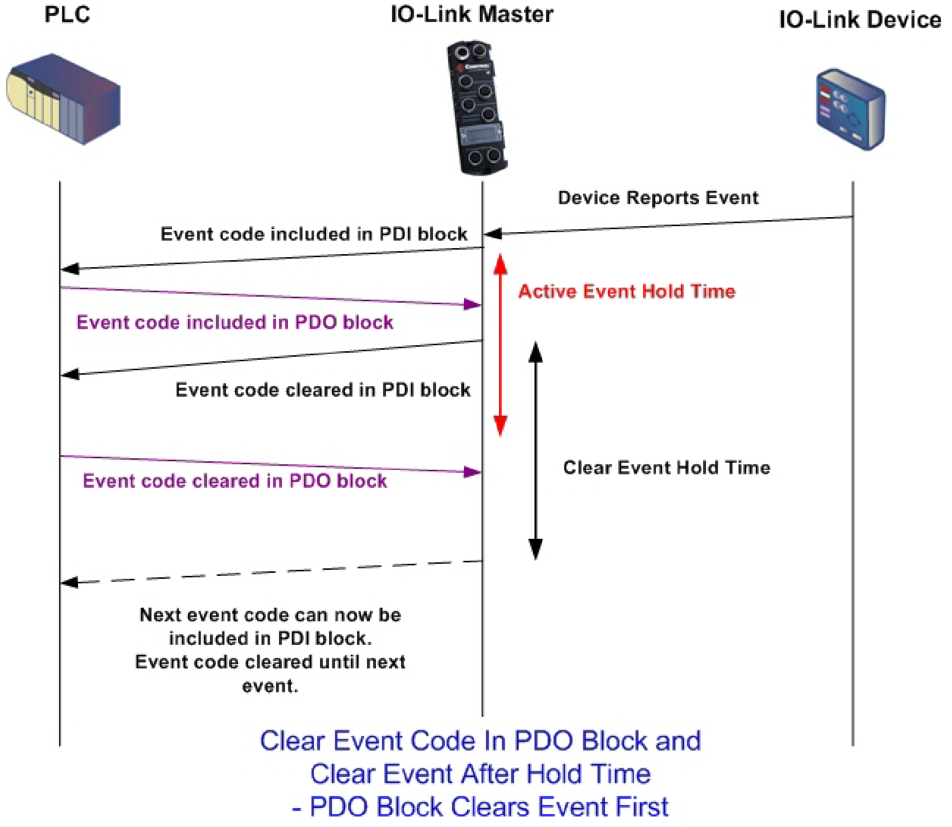
This illustrates clearing the event in the PDO block process.


PLC
IO-Link Master
IO-Link Device
Device Reports Event
Event code included in PDI block
Event code included in PDO block
Event code cleared in PDI block
Event code cleared in PDO block
Clear Event Hold Time
Next event code can now be included in PDI block. Event code cleared until next event.

Clear Event Code In PDO Block

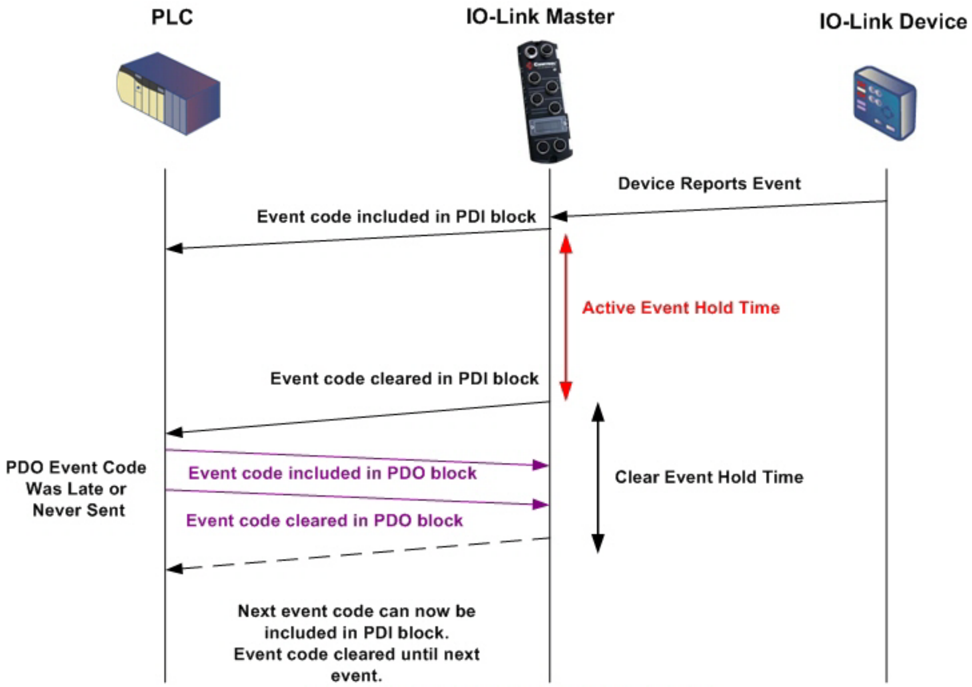
2.1.3. Clear Event Code in PDO Block and Clear Event After Hold Time Process-PDO Block First

This illustrates clearing the event code in the PDO block and clearing the event after the hold time process with the PDO block first.


PLC
IO-Link Master
IO-Link Device
Device Reports Event
Event code included in PDI block
Active Event Hold Time
Event code included in PDO block
Event code cleared in PDI block
Clear Event Hold Time
Event code cleared in PDO block
Next event code can now be included in PDI block. Event code cleared until next event.
Clear Event Code In PDO Block and Clear Event After Hold Time - PDO Block Clears Event First

2.1.4. Clear Event Code in PDO Block and Clear Event After Hold Time Process-Hold Time Expires

This illustrates clearing the event code in the PDO block and clearing the event after the hold time process with the hold time expired.


PLC
IO-Link Master
IO-Link Device
Device Reports Event
Event code included in PDI block
Active Event Hold Time
Event code cleared in PDI block
PDO Event Code Was Late or Never Sent
Event code included in PDO block
Event code cleared in PDO block
Clear Event Hold Time
Next event code can now be included in PDI block. Event code cleared until next event.

Clear Event Code In PDO Block and Clear Event After Hold Time - Hold Time Expires and Clears Event

2.2. ISDU Handling

The IO-Link Master provides a very flexible ISDU interface that is used by all supported industrial protocols. The ISDU interface contains the following:

- An ISDU *request* may contain one or multiple individual ISDU read and/or write *commands*.
- Individual ISDU command based byte swapping capabilities.
- Variable sized command structures to allow access to wide range of ISDU block sizes.
- A single ISDU request may contain as many ISDU read and/or write commands as allowed by the industrial protocol payload. For example, if an industrial protocol provides up to 500 byte read/write payloads, then an ISDU request may contain multiple commands of various lengths that can total up to 500 bytes in length.
- For the ControlLogix family of EtherNet/IP PLCs, both blocking and non-blocking ISDU request methods are provided.
  - The IO-Link Master implements blocking ISDU requests by not responding to an ISDU request message until all commands have been processed.
  - The IO-Link Master implements non-blocking ISDU requests by:
    - Responding to an ISDU request message immediately after receiving and verifying the ISDU request.
    - Requiring the PLC to monitor the ISDU request status with read messages. The IO-Link Master will not return a completed status until all of the ISDU commands have been processed.

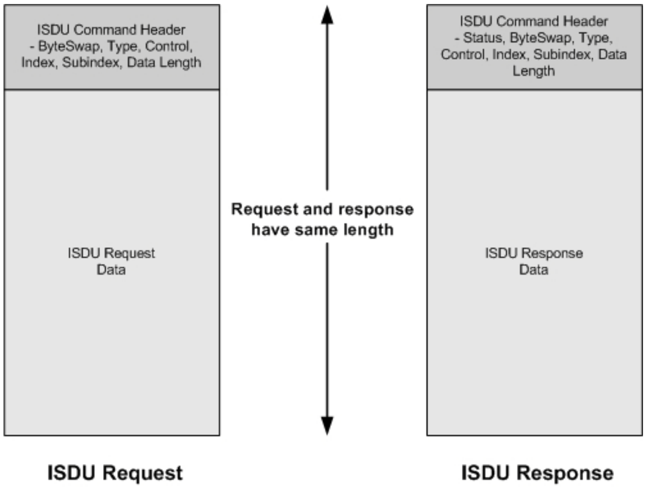
2.2.1. ISDU Request/Response Structure

ISDU requests may contain a single command or multiple, nested commands. This subsection discusses the following:

- *2.2.1.1. Single ISDU Command Request* on Page 22
- *2.2.1.2. Multiple ISDU Command Structure* on Page 23

2.2.1.1. Single ISDU Command Request

This illustrates a single ISDU command request.


ISDU Command Header - ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Request Data
Request and response have same length
ISDU Command Header - Status, ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Response Data
ISDU Request
ISDU Response

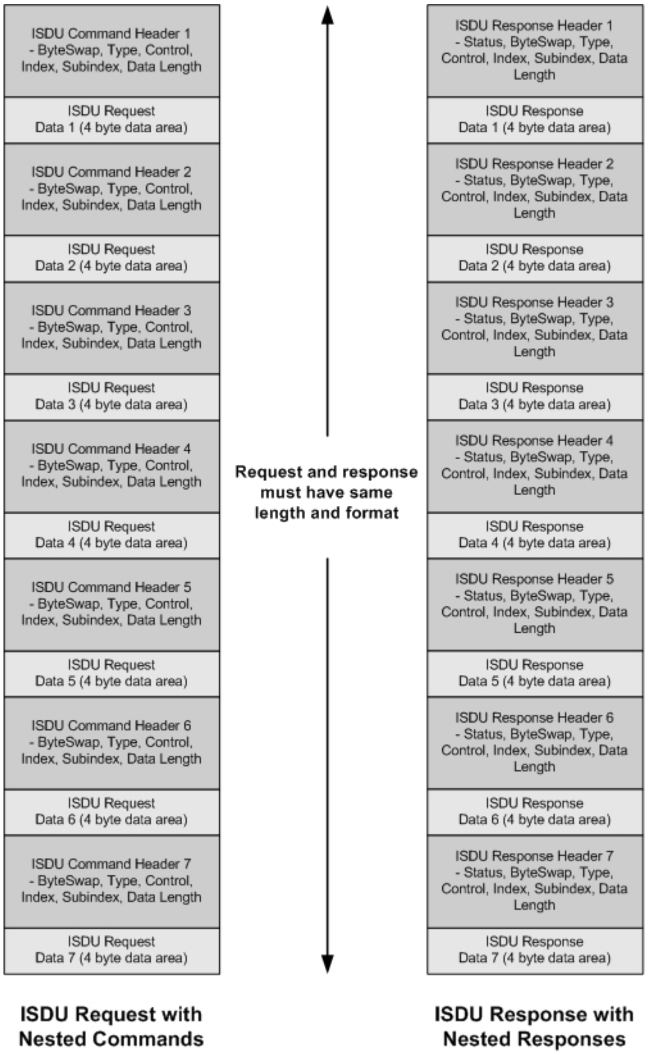
Single Command ISDU Request/Response

2.2.1.2. Multiple ISDU Command Structure

ISDU requests with multiple commands may consist of commands of the same data size or commands with different data sizes. The following are two examples of multiple ISDU commands.

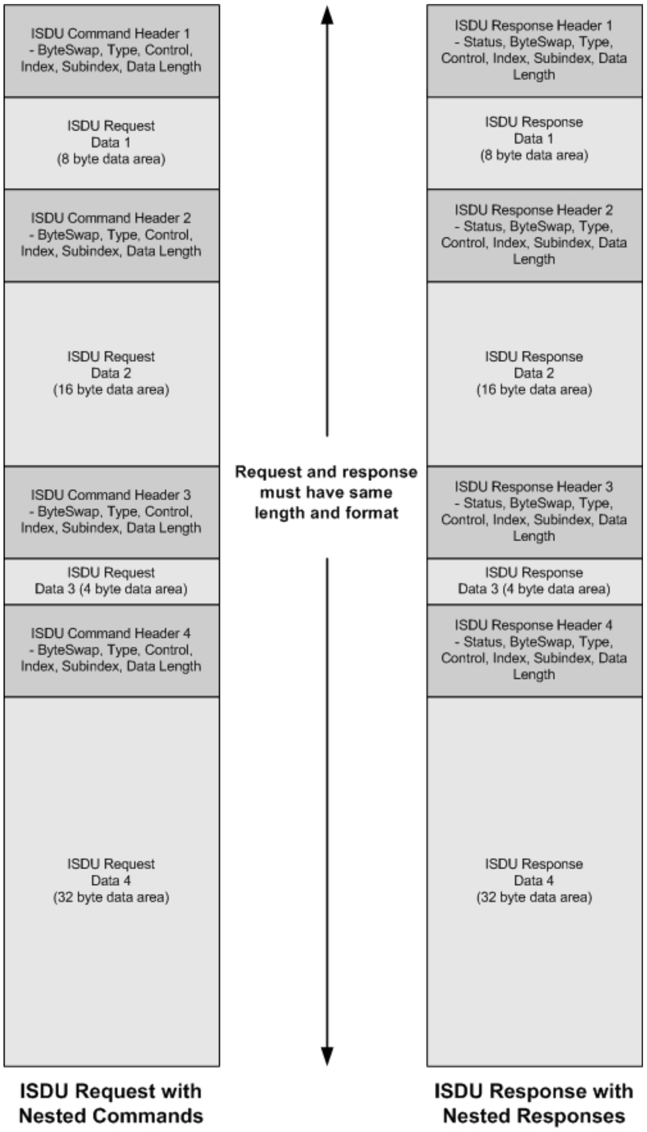
- ISDU commands of same data size (Page 23)
- ISDU commands of different data sizes (Page 24)

Multiple Command ISDU Request/Response of Same Data Area Length


ISDU Command Header 1 - ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Request Data 1 (4 byte data area)
ISDU Command Header 2 - ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Request Data 2 (4 byte data area)
ISDU Command Header 3 - ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Request Data 3 (4 byte data area)
ISDU Command Header 4 - ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Request Data 4 (4 byte data area)
ISDU Command Header 5 - ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Request Data 5 (4 byte data area)
ISDU Command Header 6 - ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Request Data 6 (4 byte data area)
ISDU Command Header 7 - ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Request Data 7 (4 byte data area)
Request and response must have same length and format
ISDU Response Header 1 - Status, ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Response Data 1 (4 byte data area)
ISDU Response Header 2 - Status, ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Response Data 2 (4 byte data area)
ISDU Response Header 3 - Status, ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Response Data 3 (4 byte data area)
ISDU Response Header 4 - Status, ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Response Data 4 (4 byte data area)
ISDU Response Header 5 - Status, ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Response Data 5 (4 byte data area)
ISDU Response Header 6 - Status, ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Response Data 6 (4 byte data area)
ISDU Response Header 7 - Status, ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Response Data 7 (4 byte data area)
ISDU Request with Nested Commands
ISDU Response with Nested Responses

Example - Multiple Command ISDU Request/Response of Same Data Area Length

Multiple Command ISDU Request/Response of Different Data Lengths


ISDU Command Header 1 - ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Request Data 1 (8 byte data area)
ISDU Command Header 2 - ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Request Data 2 (16 byte data area)
ISDU Command Header 3 - ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Request Data 3 (4 byte data area)
ISDU Command Header 4 - ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Request Data 4 (32 byte data area)
ISDU Request with Nested Commands
Request and response must have same length and format
ISDU Response Header 1 - Status, ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Response Data 1 (8 byte data area)
ISDU Response Header 2 - Status, ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Response Data 2 (16 byte data area)
ISDU Response Header 3 - Status, ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Response Data 3 (4 byte data area)
ISDU Response Header 4 - Status, ByteSwap, Type, Control, Index, Subindex, Data Length
ISDU Response Data 4 (32 byte data area)
ISDU Response with Nested Responses

Example - Multiple Command ISDU Request/Response of Different Data Area Lengths

2.2.2. ISDU Request Message Format-From PLC to IO-Link Master

Write and read ISDU commands have the same message data format. Each ISDU request message is comprised of one or more commands. The command(s) can consist of either a series of nested commands or a single read command.

Note: *A list of nested ISDU commands is terminated with either a control field of 0, (single/last operation), or the end of the message data.*

2.2.2.1. Standard ISDU Request Command Format

This table displays a standard ISDU request command format with ControlLogix PLCs.

Name	Data Type	Parameter Descriptions
Byte Swapping	USINT	**Bits 0-3**: 0= No byte swapping. 1= 16-bit (INT) byte swapping of ISDU data. 2= 32-bit (DINT) byte swapping of ISDU data. **Bits 4-7**: Set to zero. Unused.
RdWrControlType	USINT	Provides the control and type of ISDU command. **Bits 0-3, Type Field:** 0 = NOP (No operation) 1 = Read operation 2 = Write operation 3 = Read/Write “OR” 4 = Read/Write “AND” **Bits 4-7, Control Field:** 0 = Single/Last Operation (length can vary from to 1 to 232) 1 = Nested batch command – fixed 4 byte data area 2 = Nested batch command – fixed 8 byte data area 3 = Nested batch command – fixed 16 byte data area 4 = Nested batch command – fixed 32 byte data area 5 = Nested batch command – fixed 64 byte data area 6 = Nested batch command – fixed 128 byte data area 7 = Nested batch command – fixed 232 byte data area
Index	UINT	The parameter address of the data object in the IO-Link device.
Subindex	UINT	The data element address of a structured parameter of the data object in the IO-Link device.
Datalength	UINT	Length of data to read or write. For nested batch commands, the data length can vary from 1 to the fixed data area size.
Data	Array of USINTs, UINTs, or UDINTs.	Size of array is determined by the Control field in RdWrControlType. ***Note:*** *Data is valid only for write commands.*

2.2.2.2. Integer (16-Bit Word) ISDU Request Command Format

This table shows an integer (16 bit word) ISDU request command format with a SLC, MicroLogix, PLC-5, or Modbus/TCP.

Name	Data Type	Parameter Description
Byte Swapping / RdWrControlType	UINT	Provides the control, type and byte swapping of ISDU command **Bits 0-3, Type Field:** 0 = NOP (No operation) 1 = Read operation 2 = Write operation 3 = Read/Write "OR" 4 = Read/Write "AND" **Bits 4-7, Control Field:** 0 = Single/Last Operation (length can vary from to 1 to 232) 1 = Nested batch command – fixed 4 byte data area 2 = Nested batch command – fixed 8 byte data area 3 = Nested batch command – fixed 16 byte data area 4 = Nested batch command – fixed 32 byte data area 5 = Nested batch command – fixed 64 byte data area 6 = Nested batch command – fixed 128 byte data area 7 = Nested batch command – fixed 232 byte data area **Bits 8-11:** 0= No byte swapping. 1= 16-bit (INT) byte swapping of ISDU data. 2= 32-bit (DINT) byte swapping of ISDU data. **Bits 12-15:** Set to zero. Unused.
Index	UINT	The parameter address of the data object in the IO-Link device.
Subindex	UINT	The data element address of a structured parameter of the data object in the IO-Link device.
Datalength	UINT	Length of data to read or write. For nested batch commands, the data length can vary from 1 to the fixed data area size.
Data	Array of USINTs, UINTs, or UDINTs.	Size of array is determined by the Control field in RdWrControlType. ***Note:*** *Data is valid only for write commands.*

2.2.3. ISDU Response Message Format

The ISDU responses have the same data format as requests with the only exception being the returned command status. Each ISDU response message is comprised of one or more responses to the single and/or nested command(s) received in the request.

2.2.3.1. Standard ISDU Response Command Format

The following table show the standard ISDU response command format with ControlLogix PLCs.

Name	Data Type	Parameter Description
Status	USINT	Indicates the byte alignment and status of the command response. **Byte swapping, bits 0-3:** 0= No byte swapping. 1= 16-bit (INT) byte swapping of TX/RX ISDU data. 2= 32-bit (DINT) byte swapping of TX/RX ISDU data. **Status, bits 4-7:** 0 = NOP (No operation) 1 = In process (Only valid for non-blocking requests) 2 = Success 3 = Failure: IO-Link device rejected the request. 4 = Timed out: IO-Link device did not respond
RdWrControlType	USINT	Provides the control and type of ISDU request **Bits 0-3, Type Field:** 0 = NOP (No operation) 1 = Read operation 2 = Write operation 3 = Read/Write “OR” 4 = Read/Write “AND” **Bits 4-7, Control Field:** 0 = Single/Last Operation (length can vary from to 1 to 232) 1 = Nested batch command – fixed 4 byte data area 2 = Nested batch command – fixed 8 byte data area 3 = Nested batch command – fixed 16 byte data area 4 = Nested batch command – fixed 32 byte data area 5 = Nested batch command – fixed 64 byte data area 6 = Nested batch command – fixed 128 byte data area 7 = Nested batch command – fixed 232 byte data area
Index	UINT	The parameter address of the data object in the IO-Link device.
Subindex	UINT	The data element address of a structured parameter of the data object in the IO-Link device.
Datalength	UINT	Length of data that was read or written. For nested batch commands, the data length can vary from 1 to fixed data area size.
Data	Array of USINTs, UINTs, or UDINTs.	Data required for read commands. Optionally can return the data of a write command. The size of the array is determined by the Control field in the RdWrControlType. ***Note:*** *Data field not required for single NOP commands.*

2.2.3.2. Integer (16-Bit Word) ISDU Response Command Format

The following table shows an integer (16-bit word) ISDU response command format with SLC, MicroLogix, PLC-5, or Modbus/TCP.

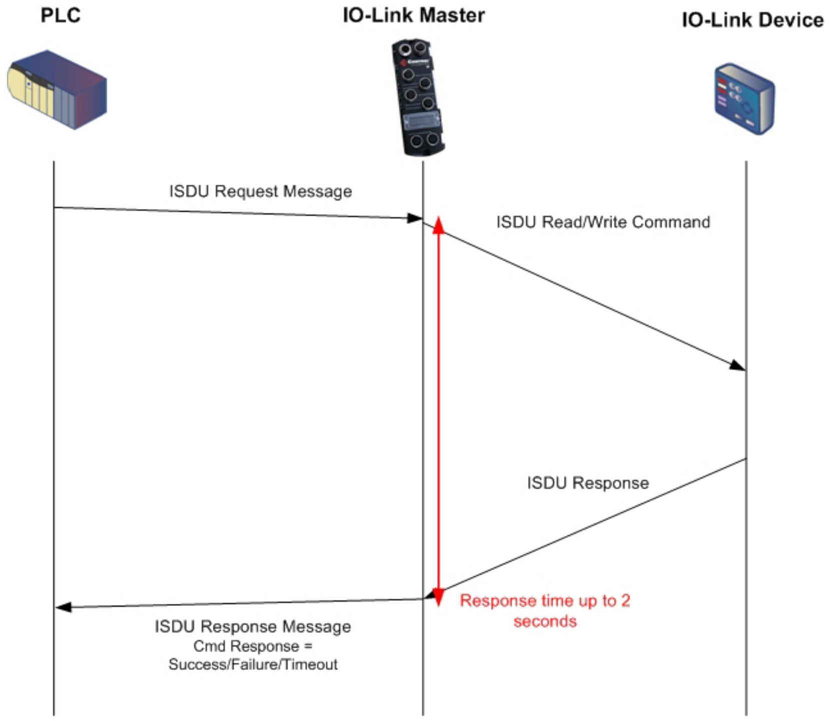
Name	Data Type	Parameter Descriptions
Status, Byte-Swapping, RdWrControlType	UINT	Indicates the control, type, byte swapping and status of the ISDU command. **Bits 0-3, Type Field:** 0 = NOP (No operation) 1 = Read operation 2 = Write operation 3 = Read/Write “OR” 4 = Read/Write “AND” **Bits 4-7, Control Field:** 0 = Single/Last Operation (length can vary from to 1 to 232) 1 = Nested batch command – fixed 4 byte data area 2 = Nested batch command – fixed 8 byte data area 3 = Nested batch command – fixed 16 byte data area 4 = Nested batch command – fixed 32 byte data area 5 = Nested batch command – fixed 64 byte data area 6 = Nested batch command – fixed 128 byte data area 7 = Nested batch command – fixed 232 byte data area **Byte swapping, bits 8-11:** 0= No byte swapping. 1= 16-bit (INT) byte swapping of TX/RX ISDU data. 2= 32-bit (DINT) byte swapping of TX/RX ISDU data. **Status, bits 12-15:** 0 = NOP (No operation) 1 = In process (Only valid for non-blocking requests) 2 = Success 3 = Failure: IO-Link device rejected the request. 4 = Timed out: IO-Link device did not respond
Index	UINT	The parameter address of the data object in the IO-Link device
Subindex	UINT	The data element address of a structured parameter of the data object in the IO-Link device.
Datalength	UINT	Length of data that was read or written. For nested batch commands, the data length can vary from 1 to fixed data area size.
Data	Array of USINTs, UINTs, or UDINTs	Data returned for read commands. Contains the data of a write command. The size of the array is determined by the Control field in **RdWrControlType**. ***Note:*** *Data field not required for single NOP commands.*

2.2.4. ISDU Blocking and Non-Blocking Methods

The IO-Link Master supports both blocking and non-blocking ISDU requests. The following diagrams demonstrate how each mode works.

2.2.4.1. Single Command Blocking

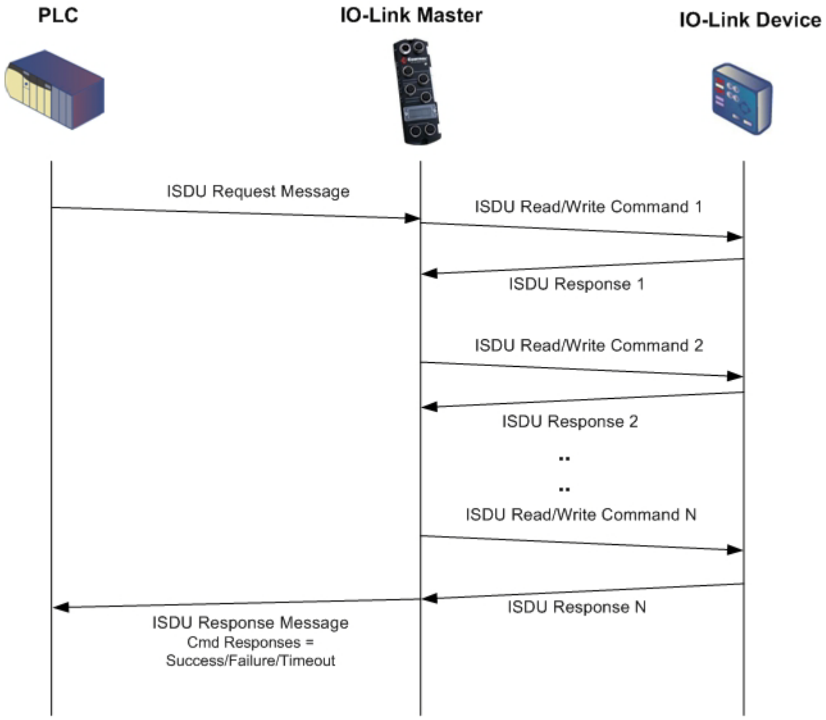
The following illustrates the single command blocking method.



Single Command ISDU Blocking Process

2.2.4.2. Multiple Command Blocking

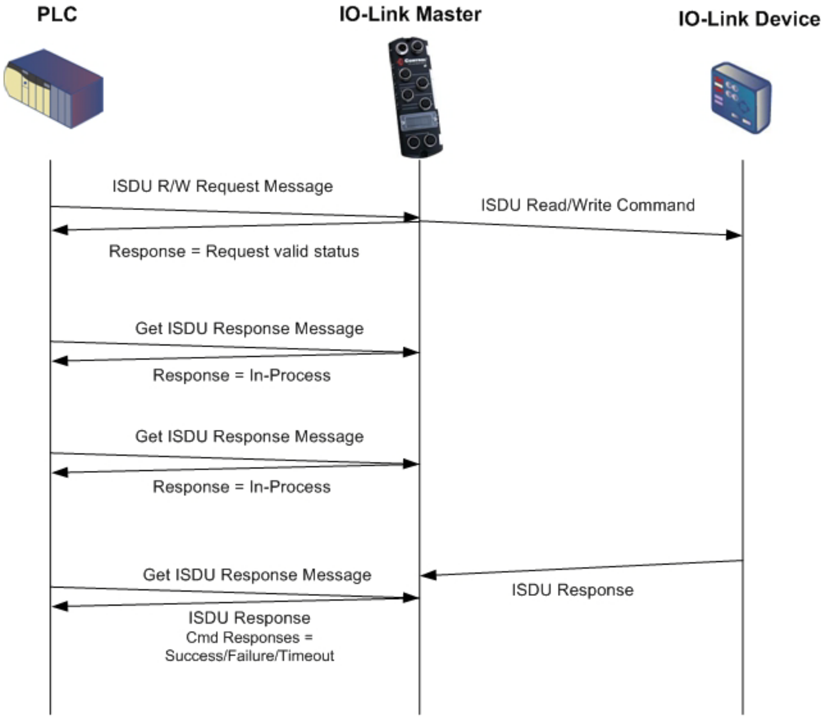
This illustrates the multiple command blocking method.



Multiple Command ISDU Blocking Process

2.2.4.3. Single Command Non-Blocking

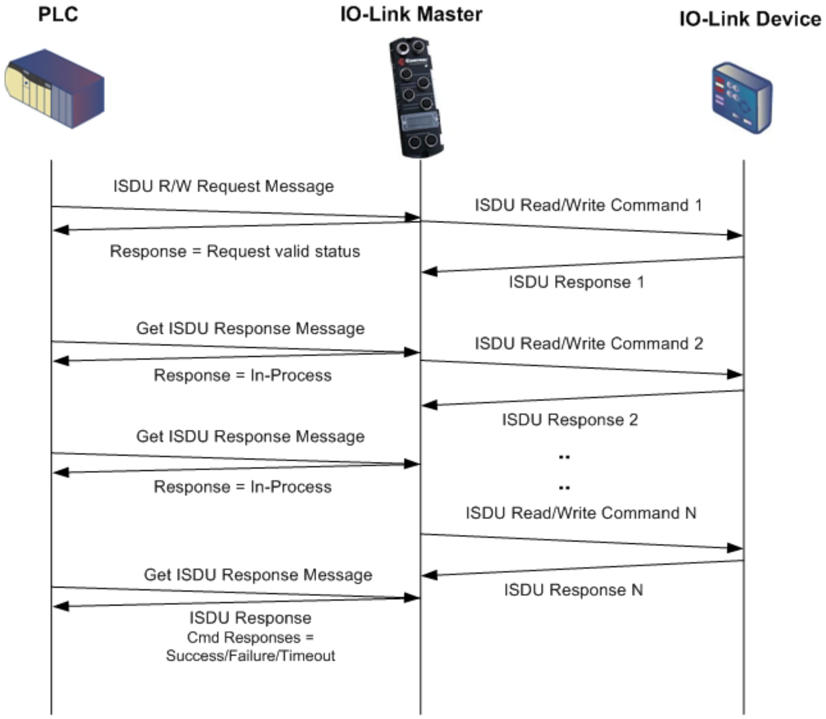
This illustrates the single command non-blocking method.



Single Command ISDU Non-Blocking Process

2.2.4.4. Multiple Command Non-Blocking

This illustrates the multiple command non-blocking method.



Multiple Command ISDU Non-Blocking Process

Chapter 3. EtherNet/IP CIP Object Definitions

The following are the vendor specific CIP Object definitions as supported in the IO-Link Master:

- *3.1. IO-Link Port Information Object Definition (71 hex)*
- *3.2. PDI (Process Data Input) Transfer Object Definition (72 hex)* on Page 41
- *3.3. PDO (Process Data Output) Transfer Object Definition (73 hex)* on Page 42
- *3.4. ISDU Read/Write Object Definition (74 hex)* on Page 43

The following are standard CIP Object Definitions that are supported in the IO-Link Master.

- *3.5. Identity Object (01hex, 1 instance)* on Page 45
- *3.6. Message Router Object (02 hex)* on Page 48
- *3.7. Connection Manager Object (06 hex)* on Page 49
- *3.8. Port Object (F4 hex-1 instance)* on Page 50
- *3.9. TCP Object (F5 hex-1 instance)* on Page 52
- *3.10. Ethernet Link Object (F6 hex-1 instance)* on Page 54
- *3.11. PCCC Object (67 hex-1 instance)* on Page 56

3.1. IO-Link Port Information Object Definition (71 hex)

The IO-Link Device Information object defines the attributes by which the PLC can request standard device information stored in the IO-Link device’s ISDU blocks.

3.1.1. Class Attributes

The following table shows the class attributes for IO-Link port information object definition (71 hex).

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Revision	UINT	1	Get
2	Max Instance	UINT	4	Get
3	Num Instances	UINT	4 ***Note:*** *Instance number determines the IO-Link port.*	Get

3.1.2. Instance Attributes

The following table shows the instance attributes for IO-Link port information object definition (71 hex).

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Vendor Name	Array of 64 SINTs	0-255	Get
2	Vendor Text	Array of 64 SINTs	0-255	Get
3	Product Name	Array of 64 SINTs	0-255	Get
4	Product Id	Array of 64 SINTs	0-255	Get
5	Product Text	Array of 64 SINTs	0-255	Get
6	Serial Number	Array of 16 SINTs	0-255	Get
7	Hardware Revision	Array of 64 SINTs	0-255	Get
8	Firmware Revision	Array of 64 SINTs	0-255	Get
9	Device PDI Length	INT	0-32	Get
10	Device PDO Length	INT	0-32	Get
11	PDI Block Length	INT	4-36	Get
12	PDO Block Length	INT	0-36	Get
13	Input Assembly PDI Offset	INT	0-108 (8-bit format) 0-54(16-bit format) 0-27 (32-bit format)	Get
14	Input Assembly PDO Offset	INT	16-246 (8-bit format) 8-123(16-bit format) 4-62 (32-bit format)	Get
15	Output Assembly PDO Offset	INT	0-102 (8-bit format) 0-51 (16-bit format) 0-26 (32-bit format)	Get
16	Control Flags	INT	Bit settings	Get

3.1.3. Common Services

The following table shows the common services for IO-Link port information object definition (71 hex).

Service Code	Implemented in Class	Implemented in Instance	Service Name
01 hex	Yes	Yes	Get_Attributes_All
0E hex	Yes	Yes	Get_Attribute_Single

3.1.4. Instance Attribute Definitions

These attributes provide access to the standard ISDU information blocks on the IO-Link devices. These ISDUs are read at IO-Link device initialization time and then provided once the IO-Link device is operational.

3.1.4.1. Attribute 1-Vendor Name

Data	Attribute 1 - Vendor Name Description
64 ASCII characters	Requested from ISDU block index 16, contains the Vendor Name description of the IO-Link device.

3.1.4.2. Attribute 2-Vendor Text

Data	Attribute 2 - Vendor Text Description
64 ASCII characters	Requested from ISDU block index 17, contains the Vendor Text description of the IO-Link device.

3.1.4.3. Attribute 3-Product Name

Data	Attribute 3 - Product Name Description
64 ASCII characters	Requested from ISDU block index 18, contains the Product Name description of the IO-Link device.

3.1.4.4. Attribute 4-Product ID

Data	Attribute 4 - Product ID Description
64 ASCII characters	Requested from ISDU block index 19, contains the Product ID description of the IO-Link device.

3.1.4.5. Attribute 5-Product Text

Data	Attribute 5 - Product Text Description
64 ASCII characters	Requested from ISDU block index 20, contains the Product Text description of the IO-Link device.

3.1.4.6. Attribute 6-Serial Number

Data	Attribute 6 - Serial Number Description
16 ASCII characters	Requested from ISDU block index 21, contains the Vendor Specific Serial Number of the IO-Link device.

3.1.4.7. Attribute 7-Hardware Revision

Data	Attribute 7 - Hardware Revision Description
64 ASCII characters	Requested from ISDU block index 22, contains the Hardware Revision of the IO-Link device.

3.1.4.8. Attribute 8-Firmware Revision

Data	Attribute 8 - Firmware Revision Description
64 ASCII characters	Requested from ISDU block index 23, contains the Firmware Revision of the IO-Link device.

3.1.4.9. Attribute 9-Device PDI Length

Data	Attribute 9 - Device PDI Length Description
INT (0-32)	Requested from ISDU block index 0, sub-index 5. Contains the number of PDI data bytes provided by the IO-Link device.

3.1.4.10. Attribute 10-Device PDO Length

Data	Attribute 10 - Device PDO Length Description
INT	Requested from ISDU block index 0, sub-index 6. Contains the number of PDO data bytes required by the IO-Link device.

3.1.4.11. Attribute 11-PDI Data Block Length

Data	Attribute 11 - PDI Data Block Length Description
INT	The configured PDI block length in units based on the configurable PDI data format (8-bit, 16-bit, 32-bit). This contains the PDI block header, (port status, auxiliary bit, event code) status and the PDI data.

3.1.4.12. Attribute 12-PDO Data Block Length

Data	Attribute 12 - PDO Data Block Length Description
INT	The configured PDO data block length in units based on the configurable PDO data format (8-bit, 16-bit, 32-bit). Depending on the configuration, this may include both the returned event code and the PDO data.

3.1.4.13. Attribute 13-Input Assembly PDI Offset

Data	Attribute 13 - Input Assembly PDI Offset Description
INT	Based from the start of the first Input Assembly instance, the PDI data block’s offset for the corresponding port’s PDI data block. This index is based on the configurable PDI data format (8-bit, 16-bit, 32-bit). To use this offset effectively, it is recommended to set IO-Link Master PDI and PDO data as well as the Class 1 I/O connection all to the same data format.

3.1.4.14. Attribute 14-Input Assembly PDO Offset

Data	Attribute 14 - Input Assembly PDO Offset Description
INT	Based from the start of the first Input Assembly instance, the PDO data block’s offset for the corresponding port’s PDO data block. This index is based on the configurable PDO data format (8-bit, 16-bit, 32-bit). To use this offset effectively, it is recommended to set IO-Link Master PDI and PDO data as well as the Class 1 I/O connection all to the same data format.

3.1.4.15. Attribute 15-Output Assembly PDO Offset

Data	Attribute 15 - Output Assembly PDO Offset Description
INT	Based from the start of the first Output Assembly instance, the PDO data block's offset for the corresponding port's PDO data block. This index is based on the configurable PDO data format (8-bit, 16-bit, 32-bit). To use this offset effectively, it is recommended to set IO-Link Master PDI and PDO data as well as the Class 1 I/O connection all to the same data format.

3.1.4.16. Attribute 16-Control Flags

Data	Attribute 16 - Control Flags Description
INT (bit-mapped word)	**Bit 0 (01h):** 1 = Indicates that the event code to clear is expected in the PDO block 0 = Indicates that the event code to clear is not expected in the PDO block. The PDO data block only contains PDO data. **Bit 1 (02h):** 1 = Indicates that the IO-Link device is SIO mode capable 0 = Indicates that the IO-Link device is not SIO mode capable **Bits 2 (04h)** 1 = Indicates that Class 1 Rx (receive PDI block) is enabled 0 = Indicates that Class 1 Rx (receive PDI block) is disabled **Bit 3 (08h):** 1 = Indicates that Class 1 Tx (transmit PDO) is enabled 0 = Indicates that Class 1 Tx (transmit PDO) is disabled **Bits 4-15: Reserved**

3.2. PDI (Process Data Input) Transfer Object Definition (72 hex)

The PDI Transfer object defines the attributes by which the PLC can request the PDI data block from the IO-Link Master.

3.2.1. Class Attributes

The following table displays Class Attributes for the PDI Transfer Object Definition (72 hex).

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

3.2.2. Instance Attributes

The following table displays Instance Attributes for the PDI Transfer Object Definition (72 hex).

Attribute ID	Name	Data Type	Length	Data Values	Access Rule
1	Port 1 PDI data block	Array of BYTEs	4-36 bytes	0-255	Get
2	Port 2 PDI data block	Array of BYTEs	4-36 bytes	0-255	Get
3	Port 3 PDI data block	Array of BYTEs	4-36 bytes	0-255	Get
4	Port 4 PDI data block	Array of BYTEs	4-36 bytes	0-255	Get

3.2.3. Common Services

The following table shows Common Services for the PDI Transfer Object Definition (72 hex).

Service Code	Implemented in Class	Implemented in Instance	Service Name
01 hex	Yes	Yes	Get_Attributes_All
0E hex	Yes	Yes	Get_Attribute_Single

3.2.4. Instance Attribute Definitions - Attribute 1 to 4-PDI Data Blocks

These attributes provide access to the PDI data blocks.

- Get Attribute Single requests return the PDI data block for a specific port.
- Get Attribute All requests return all PDI data blocks from the IO-Link Master.

All PDI data is returned in the configured PDI format (8-bit, 16-bit or 32-bit). Refer to *3.2. PDI (Process Data Input) Transfer Object Definition (72 hex)* on Page 41 for a detailed explanation of the PDI data block.

3.3. PDO (Process Data Output) Transfer Object Definition (73 hex)

The PDO Transfer object defines the attributes by which the PLC can:

- Request the PDO data block from the IO-Link Master.
- Write PDO data block to the IO-Link Master.

3.3.1. Class Attributes

The following table displays the Class Attributes for the PDO Transfer Object Definition (73 hex).

Attribute ID	Name	Data Type	Data Value	Access Rule
1	Revision	UINT	1	Get
2	Max Instance	UINT	1	Get
3	Num Instances	UINT	1	Get

3.3.2. Instance Attributes

The following table displays the Instance Attributes for the PDO Transfer Object Definition (73 hex).

Attribute ID	Name	Data Type	Length	Data Value	Access Rule
1	Port 1 PDO data block	Array of BYTEs	0-36 bytes	0-255	Get/Set
2	Port 2 PDO data block	Array of BYTEs	0-36 bytes	0-255	Get/Set
3	Port 3 PDO data block	Array of BYTEs	0-36 bytes	0-255	Get/Set
4	Port 4 PDO data block	Array of BYTEs	0-36 bytes	0-255	Get/Set

3.3.3. Common Services

The following table displays the Common Services for the PDO Transfer Object Definition (73 hex).

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

3.3.4. Instance Attribute Definitions - Attribute 1 to 4-PDO Data Blocks

These attributes provide write access to the PDO data blocks.

- Get Attribute Single requests return the current PDO data block for a specific port.
- Get Attribute All requests return all current PDO data blocks from the IO-Link Master.
- Set Attribute Single allows writing the PDO data to one IO-Link port on the IO-Link Master.
- Set Attribute All messages allow writing of PDO data to all IO-Link ports on the IO-Link Master.

All PDO data is received and returned in the configured PDO format (8-bit, 16-bit or 32-bit). Refer to 3.3. PDO (Process Data Output) Transfer Object Definition (73 hex) on Page 42 for a detailed explanation of the PDO data block.

3.4. ISDU Read/Write Object Definition (74 hex)

The ISDU Read/Write object defines the attributes by which the PLC can:

- Send an ISDU request containing one or more read and/or write ISDU commands to an IO-Link device via the IO-Link Master.
- Request the ISDU response(s) from the IO-Link Master.
- Send both blocking and non-blocking ISDU requests.

Refer to the ISDU Handling chapter for a detailed description of the ISDU functionality.

3.4.1. Class Attributes

The following table shows the Class Attributes for the ISDU Read/Write Object Definition (74 hex).

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Revision	UINT	1	Get
2	Max Instance	UINT	4	Get
3	Num Instances	UINT	4 ***Note:*** *Instance number determines IO-Link port on the IO-Link Master.*	Get

3.4.2. Instance Attributes

The following table shows the Instance Attributes for the ISDU Read/Write Object Definition (74 hex).

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	ISDU Response	ISDU response data block	0-255	Get
2	ISDU Read/Write Request	ISDU request data block	0-255	Set

3.4.3. Common Services

The following table shows the Common Services for the ISDU Read/Write Object Definition (74 hex).

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

3.4.4. Object Specific Services

The following table shows the Object Specific Services for the ISDU Read/Write Object Definition (74 hex).

Service Code	Implemented in Class	Implemented in Instance	Service Name
4B hex	No	Yes	Blocking ISDU Request

The Blocking ISDU Request service allows one message instruction to both send an ISDU request and receive the response. Using this service causes the message to be active for several seconds.

3.4.5. Instance Attribute Definitions

The following attributes provide access to the ISDU blocks on the IO-Link devices.

3.4.5.1. Attribute 1-ISDU Read/Write Response (Non-Blocking only)

Get Attribute Single messages returns the ISDU response for a specific port through the IO-Link Master. The response may need to be read multiple times until a response of Success, Failure, or Timed Out has been received.

3.4.5.2. Attribute 2-ISDU Read/Write Request (Non-blocking only)

Set Attribute Single messages can send read/write type ISDU requests to the IO-Link devices via the IO-Link Master. The ISDU request message need be sent only once for each ISDU read/write request.

3.5. Identity Object (01hex, 1 instance)

The Identity Object provides identification of and general information about the IO-Link Master.

3.5.1. Class Attributes

This table shows the Class Attributes for the Identity Object (01 hex, 1 Instance).

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

3.5.2. Instance Attributes

This table shows the Instance Attributes for the Identity Object (01 hex, 1 Instance).

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Vendor ID	UINT	909 (Comtrol)	Get
2	Device Type	UINT	2B hex (Generic Device)	Get
3	Product Code	UINT	As defined by Comtrol	Get
4	Revision (Product or Software release) *Structure of:* Major Revision Minor Revision	 USINT USINT	 1 to 127 1 to 255	Get
5	Status	WORD	See Below	Get
6	Serial Number	UDINT	1-FFFFFFFF hex	Get
7	Product Name Structure of: Name Length Name String	 USINT STRING	 Length of string See below	 Get Get

3.5.3. Status Word

Refer to Page 52 of Volume 3.5 of the CIP Common Specification.

The following applies to the Identity Object status word for the IO-Link Master.

Status Word Bit	Setting	Description
0	0	Ownership Flag. Does not apply to the IO-Link Master.
1	0	Reserved.
2	0	IO-Link Master is operating on the default configuration.
	1	The IO-Link Master has a configuration other than the default configuration.
3	0	Reserved.
4-7	0101 (0x50)	Indicates that there is a major fault (either Bit 10 or Bit 11 is set).
	0100 (0x40)	Indicates the stored configuration is invalid.
	0011 (0x30)	Indicates the system is operational and there are no I/O (Class 1) connections.
	0110 (0x60)	Indicates the system is operational and there is at least one active I/O (Class 1) connection.
	0000	Indicates the system is not operational. It may be in any of the following states: • System startup. • Configuration in process. • Idle. • Critical (major) fault.
8	0	No recoverable minor fault. No error history entry reported within the last ten seconds.
	1	Recoverable minor fault. The IO-Link Master has reported an error within the last ten seconds and a major fault has not been detected.
9	1	Unrecoverable minor fault. Does not apply to the IO-Link Master.
10	0	No recoverable major fault.
	1	A major recoverable fault exists. This is a fault that the IO-Link Master may be able to recover from by a system reset. If the system does not recover automatically, a system reset message or a power cycle of the IO-Link Master may be required.
11	0	No major unrecoverable fault.
	1	A major unrecoverable fault has occurred in the IO-Link Master. If the major fault is not corrected with a system reset or a power cycle, refer to the User Manual or call Comtrol support.
12-15	0	Reserved.

3.5.4. Common Services

Service Code	Implemented in Class	Implemented in Instance	Service Name
01 hex	Yes	Yes	Get_Attribute_All

05 hex	No	Yes	Reset
0E hex	Yes	Yes	Get_Attribute_Single

3.6. Message Router Object (02 hex)

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

3.6.1. Class Attributes

This table displays the Class Attributes for the Message Router Object (02 hex).

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

3.6.2. Instance Attributes

This table displays the Instance Attributes for the Message Router Object (02 hex)

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

3.6.3. Common Services

This table displays the Common Services for the Message Router Object (02 hex)

Service Code	Implemented in Class	Implemented in Instance	Service Name
01 hex	Yes	No	Get_Attribute_All
0E hex	Yes	Yes	Get_Attribute_Single
0A hex	No	Yes	Multiple_Service_Req

3.7. Connection Manager Object (06 hex)

This object provides services for connection and connection-less communications.

This object has no supported attributes.

3.7.1. Class Attributes Object (06 hex)

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

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

3.7.2. Instance Attributes (02 hex)

This table displays the Instance Attributes for the Message Router Object (02 hex).

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

3.7.3. Common Services Object (06 hex)

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

Service Code	Implemented in Class	Implemented in Instance	Service Name
01 hex	Yes	Yes	Get_Attribute_All
02 hex	No	Yes	Set_Attribute_ALL
0E hex	Yes	Yes	Get_Attribute_Single
10 hex	No	Yes	Set_Attribute_Single
4E hex	N/A	N/A	Forward_Close

Service Code	Implemented in Class	Implemented in Instance	Service Name
52 hex	N/A	N/A	Unconnected_Send
54 hex	N/A	N/A	Forward_Open
5A hex	N/A	N/A	Get_Connection_Owner
5B hex	N/A	N/A	Large_Forward_Open

3.8. Port Object (F4 hex-1 instance)

The Port Object enumerates the CIP ports present on the IO-Link Master.

3.8.1. Class Attributes

This table illustrates the Class Attributes for the Port Object (F4 hex - 1 Instance)

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
6	Maximum Number Class Attributes	UINT	9	Get
7	Maximum Number Instance Attributes	UINT	7	Get
8	Entry Port	UINT	1	Get
9	All Ports	Array of UINT	[0]=0 [1]=0 [2] = 1 (Vendor Specific) [3] = 1 (Backplane) [4]=TCP_IP_PORT_TYPE (4) [5]=TCP_IP_PORT_NUMBER(2)	Get

3.8.2. Instance Attributes

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

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Port Type	UINT	1	Get
2	Port Number	UINT	1	Get
3	Port Object *Structure of:* 16 bit word count in path Path	UINT Array of UINT	2 [0]=6420 hex [1]=0124 hex	Get Get
4	Port Name *Structure of:* String Length Port Name	USINT Array of USINT	10 “Backplane”	Get Get
7	Node Address	USINT[2]	0x10, 0x00	Get

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

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Port Type	UINT	4 (TCP/IP)	Get
2	Port Number	UINT	2 (TCP/IP)	Get
3	Port Object *Structure of:* 16 bit word count in path Path	UINT Array of UINT	2 [0]=F520 hex [1]=0124 hex	Get Get
4	Port Name *Structure of:* String Length Port Name	USINT Array of USINT	17 “Ethernet/IP Port”	Get Get
7	Node Address	USINT[2]	0x10, 0x00	Get

3.8.3. Common Services

This table illustrates the Common Services for the Port Object (F4 hex - 1 Instance).

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

3.9. TCP Object (F5 hex-1 instance)

The TCP/IP Interface Object provides the mechanism to retrieve the TCP/IP attributes for the IO-Link Master.

3.9.1. Class Attributes

This table shows the Class Attributes for the TCP Object (F5 hex - I Instance).

Attribute ID	Name	Data Type	Data Value	Access Rule
1	Revision	UINT	1	Get
2	Max Instance	UINT	1	Get
3	Num Instances	UINT	1	Get
4	Optional Attribute List	UINT	4	Get
6	Maximum Number Class Attribute	UINT	7	Get
7	Maximum Number Instance Attribute	UINT	9	Get

3.9.2. Instance Attributes

This table shows the Instance Attributes for the TCP Object (F5 hex - I Instance).

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Status	DWORD	0 = The Interface Configuration attribute has not been configured. 1 = The Interface Configuration attribute contains configuration obtained from DHCP or nonvolatile storage. 2 = The IP address member of the Interface Configuration attribute contains configuration obtained, in part, from the hardware rotary switch settings. • Upper 3 bytes from nonvolatile storage. • Least significant byte from rotary switches.	Get
2	Configuration Capability	DWORD	34 hex (DHCP, Settable and Hardware) 04 hex = DHCP 10 hex = Settable 20 hex = Hardware configurable	Get

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
3	Configuration Control	DWORD	Interface control Flags: 0 = The device shall use statically-assigned IP configuration values. 2 = The device shall obtain its interface configuration values via DHCP.	Set/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 Address> <Name server> <Name server2> <Length of name> <Domain name>	Set/Get
6	Host Name Structure of: Host Name Length Host Name String	UINT STRING	0 to 15 <Default =IP NULL (0)>	Set/Get
8	TTL (Time-to-Live) value for IP multicast packets.	USINT	1 to 255 <Default = 1>	Set/Get
9	IP Multicast Address Configuration	*Struct of:* USINT - Alloc Control USINT - Reserved UINT - Num Mcast UDINT - Start Mcast Address	Alloc Control: 0 = Default Algorithm 1 = Configuration Num Mcast: 1 to 32 Start Mcast Address: 239.192.1.0 to 239.255.255.255	Set/Get

3.9.3. Common Services

This table shows the Common Services for the TCP Object (F5 hex - I Instance).

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

3.10. Ethernet Link Object (F6 hex-1 instance)

The Ethernet Link Object maintains link-specific counters and status information for the Ethernet communications interface on the IO-Link Master.

3.10.1. Class Attributes

This table displays the Class Attributes for the Ethernet Link Object (F6 hex - 1 Instance).

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

3.10.2. Instance Attributes

This table displays the Instance Attributes for the Ethernet Link Object (F6 hex - 1 Instance).

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Interface speed (Current operational speed)	UDINT	10=10 Mbit 100=100 Mbit	Get
2	Interface Flags (Current operational status)	DWORD	Bit 0 =link status (0=inactive) (1=active) Bit 1=Half/Full Duplex (0=half duplex) (2=full duplex) Bits 2-4: 00 = negotiation in progress 01 = negotiation failed 02 = negotiation failed speed OK 03 = negotiation success	Get
3	Physical Address	Array of 6 USINT	MAC Address	Get
7	Interface Type	USINT	2 = Twisted Pair	Get
8	Interface State	USINT	1 = Interface is enabled and operational	Get
9	Admin State	USINT	1 = Interface enabled	Get
10	Interface Label	USINT16 Array of USINT	Length = 1 to 64 ASCII characters <Default = IP address in "xxx.xxx.xxx.xxx" format>	Get

3.10.3. Common Services

This table displays the Common Services for the Ethernet Link Object (F6 hex - 1 Instance)

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

3.11. PCCC Object (67 hex-1 instance)

The PCCC Object provides the ability to encapsulate and then transmit and receive PCCC messages between devices on an Ethernet/IP network. This object is used to communicate to MicroLogix, SLC 5/05 and PLC-5 PLCs over EtherNet/IP.

The PCCC Object does not support the following:

- Class Attributes
- Instance Attributes

3.11.1. Instances

The PCCC Object supports Instance 1.

3.11.2. Common Services

The following table displays the Common Services for the PCCC Object.

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

3.11.3. Message Structure Execute_PCCC: Request Message

This table displays the message structure for the Execute_PCCC Request Message for the PCCC Object.

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

3.11.4. Message Structure Execute_PCCC: Response Message

This table displays the message structure for the Execute_PCCC Response Message for the PCCC Object.

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

3.11.5. Supported PCCC Command Types

The following table displays the Supported PCCC Command Types for the PCCC Object.

CMD	FNC	Description
0F hex	A2 hex	SLC 500 protected typed read with 3 address fields
0F hex	AA hex	SLC 500 protected typed write with 3 address fields

3.12. Assembly Object (For Class 1 Interface)

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

For the IO-Link Master, the Assembly Object corresponds to the PDI and PDO Transfer objects. Each instance of the Assembly Object corresponds to one or more of the PDI and/or PDO Transfer Object attributes.

The Assembly Object is linked to the Process IO vendor specific object, which provides access to the PDI and PDO data. The Assembly object defines the interface by which a Class 1 PLC or controller can:

- Request the PDI data block from the IO-Link Master.
- Write the PDO data block to the IO-Link Master.

3.12.1. Class Attributes

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

Attribute ID	Name	Data Type	Data Value(s)	Access Rule
1	Revision	UINT	1	Get
2	Max Instance	UINT	12	Get
3	Num Instances	UINT	12	Get

3.12.2. Instance Definitions

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

Assembly Instance Number	Description	Data Type	Data Values	Access Rule
101	PDI data blocks from Ports 1 to 4. PDO data blocks from ports 1-4	BYTE Array Valid read lengths: 1-288	0-255	Get
102	PDI data blocks from Ports 2 to 4. PDO data blocks from Ports 1-4	BYTE Array Valid read lengths: 1-252	0-255	Get
103	PDI data blocks from Ports 3 to 4. PDO data blocks from Ports 1-4	BYTE Array Valid read lengths: 1-216	0-255	Get
104	PDI data blocks from Port 4. PDO data blocks from Ports 1-4	BYTE Array Valid read lengths: 1-180		
105	PDO data blocks from Ports 1-4	BYTE Array Valid read lengths: 0-144	0-255	Get

Assembly Instance Number	Description	Data Type	Data Values	Access Rule
106	PDO data blocks from Ports 2-4	BYTE Array Valid read lengths: 0-108	0-255	Get
107	PDO data blocks from Ports 3-4	BYTE Array Valid read lengths: 0-72	0-255	Get
108	PDO data blocks from Port 4	BYTE Array Valid read lengths: 0-36	0-255	Get
109	PDO data blocks to Ports 1-4	BYTE Array Valid read lengths: 0-144	0-255	Set
111	PDO data blocks to Ports 3-4	BYTE Array Valid read lengths: 0-72	0-255	Set
112	PDO data blocks to Port 4	BYTE Array Valid read lengths: 0-36	0-255	Set

3.12.3. Instance Attributes

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

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

3.12.4. Common Services

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

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

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

Dependent on the instance number, this is either the PDI data block and/or the PDO data block.

3.12.6. Instance Attribute Definitions: Attribute 4-Data Length

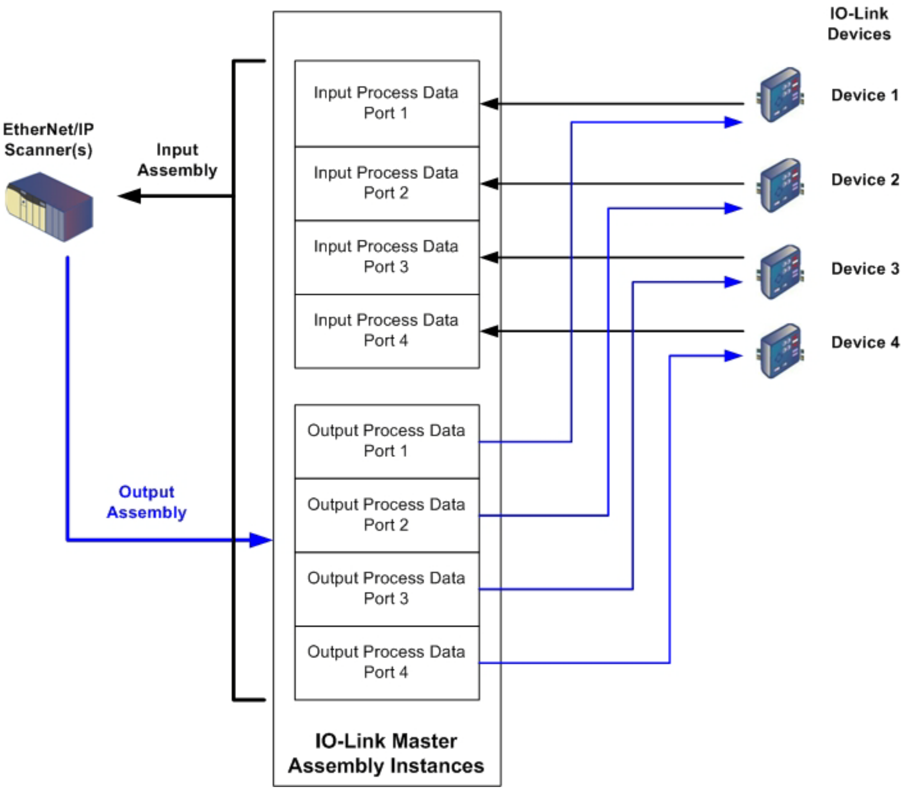
This is the maximum data length for each Assembly instance.

3.12.7. Overview of Assembly Interface

The Assembly interface is designed to:

- Provide access to all Input and Output assemblies.
- Maximize flexibility for the PLC programmer.
- Minimize required PLC and IO-Link communication bandwidth.
- Be as easy to use as possible.

The following diagram illustrates the Assembly instances for a four port IO-Link Master. There is one Assembly input and output instance assigned to each IO-Link port.


IO-Link Devices
EtherNet/IP Scanner(s)
Input Assembly
Input Process Data Port 1
Input Process Data Port 2
Input Process Data Port 3
Input Process Data Port 4
Output Process Data Port 1
Output Process Data Port 2
Output Process Data Port 3
Output Process Data Port 4
Output Assembly
Device 1
Device 2
Device 3
Device 4
IO-Link Master Assembly Instances

3.12.8. Grouping of Assembly Instances

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

Assembly Controller Access									
	Assembly Instance Number	Controller Port 1 Access		Controller Port 2 Access		Controller Port 3 Access		Controller Port 4 Access	
		Read (Input)	Write (Output)	Read (Input)	Write (Output)	Read (Input)	Write (Output)	Read (Input)	Write (Output)
Read (Input) Process Data Input	**101 (Port 1)**	▨							
	102 (Port 2)	▨		▨					
	103 (Port 3)	▨		▨		▨			
	104 (Port 4)	▨		▨		▨		▨	
Read (Input) Process Data Output	**105 (Port 1)**	▨		▨		▨		▨	
	106 (Port 2)	▨		▨		▨		▨	
	107 (Port 3)	▨		▨		▨		▨	
	108 (Port 4)	▨		▨		▨		▨	
Write (Output) Process Data Output	**109 (Port 1)**		▨						
	110 (Port 2)		▨		▨				
	111 (Port 3)		▨		▨		▨		
	112 (Port 4)		▨		▨		▨		▨

Where:

- All accessible data can be read (input) and written (output) from one I/O connection.
- Controller Read (Input) access:
  - One or more input instances may be read with one I/O connection. (i.e. If addressing the instance 101, all input instances for both PDI and PDO data, 101 to 108, may be read in one connection.)
  - The length of the Read (Input) connection can range from 1 to the total length for all input instances.
  - Multiple controllers can read access to the Input Assembly instances at one time.
- Controller Write (Output) access:
  - Only output instances may be written.

- One or more output instances may be written to with one connection.
- The length of the Write (Output) connection must be equal to the total length of the output instance(s).
- Only one controller may have write access to an output instance.

Chapter 4. IO-Link Port Configuration

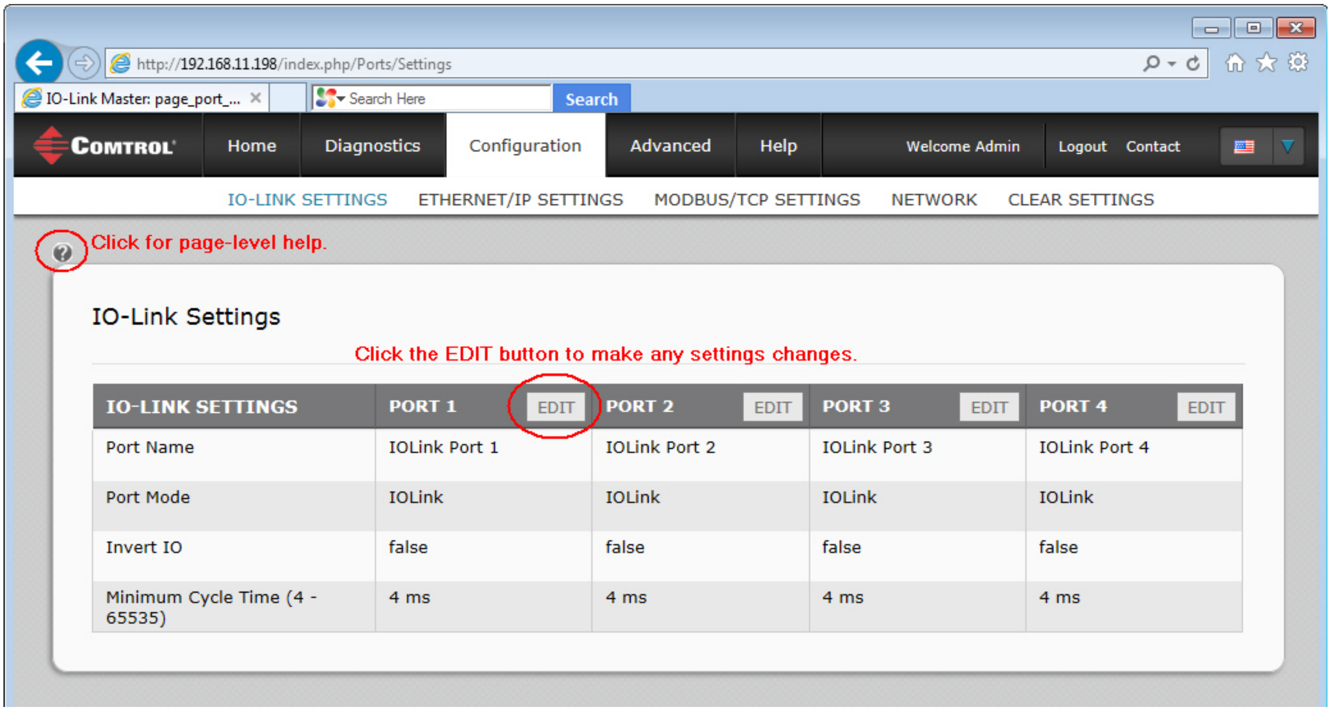
This section discusses port configuration, which includes these topics:

- 4.1. IO-Link Settings Configuration Page
- 4.2. EtherNet/IP Settings Configuration Page on Page 65

Note: The IO-Link Master may work out of the box for ControlLogix PLCs.

4.1. IO-Link Settings Configuration Page

Use the *IO-Link Settings* page to configure IO-Link port characteristics for the IO-Link Master.



IO-Link Settings

IO-LINK SETTINGS	PORT 1 EDIT	PORT 2 EDIT	PORT 3 EDIT	PORT 4 EDIT
Port Name	IOLink Port 1	IOLink Port 2	IOLink Port 3	IOLink Port 4
Port Mode	IOLink	IOLink	IOLink	IOLink
Invert IO	false	false	false	false
Minimum Cycle Time (4 - 65535)	4 ms	4 ms	4 ms	4 ms

4.1.1. Editing IO-Link Settings

You can use this procedure to configure IO-Link characteristics for each port. The following table or help system provides information about each option.

1. If necessary, open the IO-Link Master web interface with your web browser using the IP address or through PortVision DX.
2. Click **Configuration** in the menu bar, which by default loads the *IO-Link Settings* page.
3. Click the **EDIT** button for the port that you want to configure.
4. Make appropriate selections for the IO-Link device that you will connect to that port. You can use the help system if you require definitions or values for the options or *4.1.2. IO-Link Settings Parameters* on Page 64.
5. Click the **SAVE** button.
6. Repeat for each port that requires configuration changes.

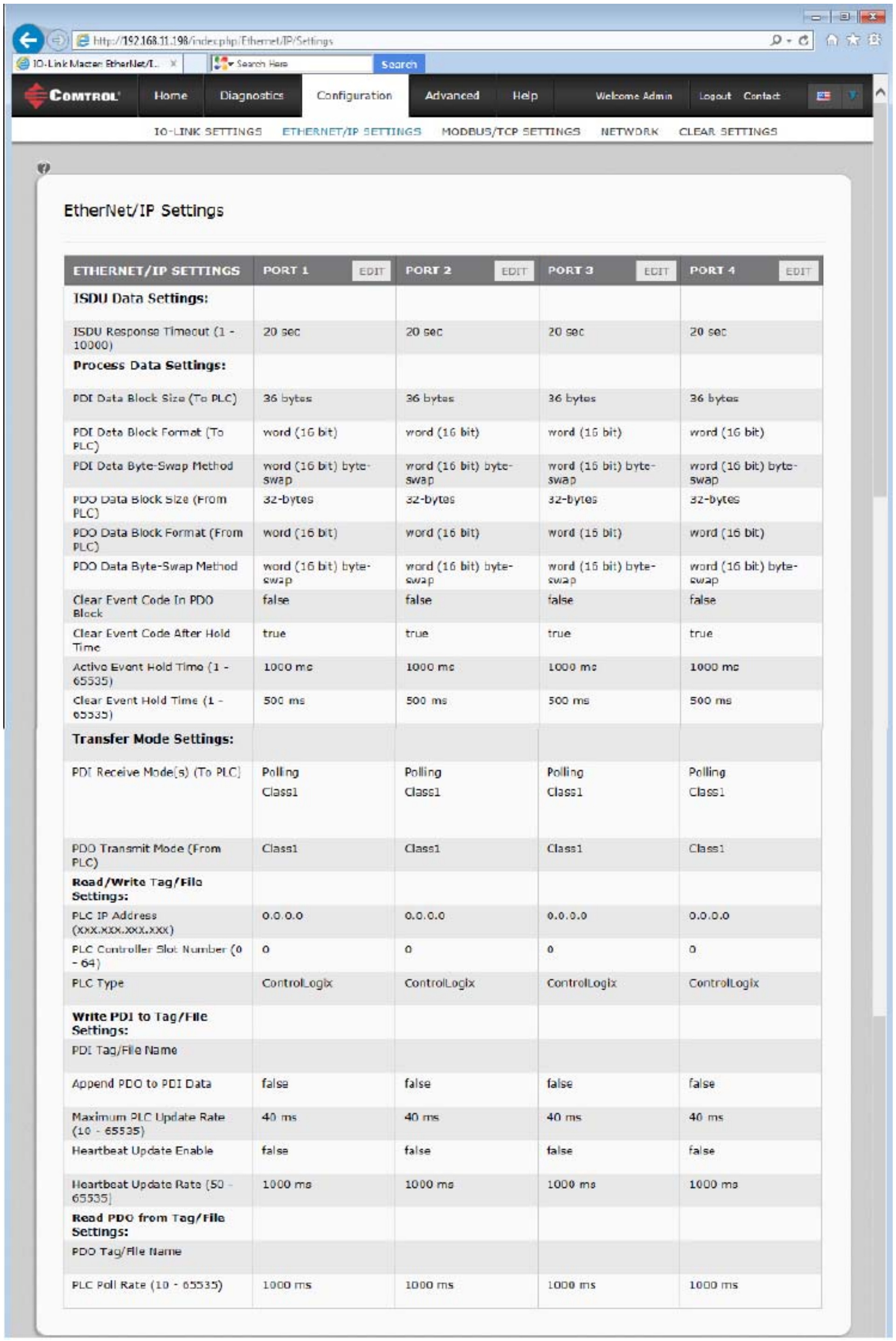
4.1.2. IO-Link Settings Parameters

The *IO-Link Settings* configuration page supports the following options.

IO-LINK SETTINGS Page	
Port Name	User defined port or device description. • Standard ASCII characters • Max length = 80 characters
Port Mode *Default*: IO-Link	Selected IO-Link Port Mode. Valid settings are: • Reset • IO-Link • Digital In • Digital Out
Invert IO *Default*: False	If enabled and the *Port Mode* is Digital In or Digital Out, inverts the I/O value. 0= False (Disabled - Do not invert IO) 1= True (Enabled - Invert IO) *Note: Does not affect the Auxiliary Input.*
Minimum Cycle Time *Default*: 4	The minimum, or fastest, cycle time that the IO-Link device may operate at. The valid range is 4-65535 ms.

4.2. EtherNet/IP Settings Configuration Page

Use the *EtherNet/IP Settings* page to configure EtherNet/IP port options.



EtherNet/IP Settings

ETHERNET/IP SETTINGS	PORT 1 EDIT	PORT 2 EDIT	PORT 3 EDIT	PORT 4 EDIT
ISDU Data Settings:				
ISDU Response Timeout (1 - 10000)	20 sec	20 sec	20 sec	20 sec
Process Data Settings:				
PDI Data Block Size (To PLC)	36 bytes	36 bytes	36 bytes	36 bytes
PDI Data Block Format (To PLC)	word (16 bit)	word (16 bit)	word (16 bit)	word (16 bit)
PDI Data Byte-Swap Method	word (16 bit) byte-swap	word (16 bit) byte-swap	word (16 bit) byte-swap	word (16 bit) byte-swap
PDO Data Block Size (From PLC)	32-bytes	32-bytes	32-bytes	32-bytes
PDO Data Block Format (From PLC)	word (16 bit)	word (16 bit)	word (16 bit)	word (16 bit)
PDO Data Byte-Swap Method	word (16 bit) byte-swap	word (16 bit) byte-swap	word (16 bit) byte-swap	word (16 bit) byte-swap
Clear Event Code In PDO Block	false	false	false	false
Clear Event Code After Hold Time	true	true	true	true
Active Event Hold Time (1 - 65535)	1000 ms	1000 ms	1000 ms	1000 ms
Clear Event Hold Time (1 - 65535)	500 ms	500 ms	500 ms	500 ms
Transfer Mode Settings:				
PDI Receive Mode(s) (To PLC)	Polling Class1	Polling Class1	Polling Class1	Polling Class1
PDO Transmit Mode (From PLC)	Class1	Class1	Class1	Class1
Read/Write Tag/File Settings:				
PLC IP Address (xxx.xxx.xxx.xxx)	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0
PLC Controller Slot Number (0 - 64)	0	0	0	0
PLC Type	ControlLogix	ControlLogix	ControlLogix	ControlLogix
Write PDI to Tag/File Settings:				
PDI Tag/File Name				
Append PDO to PDI Data	false	false	false	false
Maximum PLC Update Rate (10 - 65535)	40 ms	40 ms	40 ms	40 ms
Heartbeat Update Enable	false	false	false	false
Heartbeat Update Rate (50 - 65535)	1000 ms	1000 ms	1000 ms	1000 ms
Read PDO from Tag/File Settings:				
PDO Tag/File Name				
PLC Poll Rate (10 - 65535)	1000 ms	1000 ms	1000 ms	1000 ms

4.2.1. Editing EtherNet/IP Settings

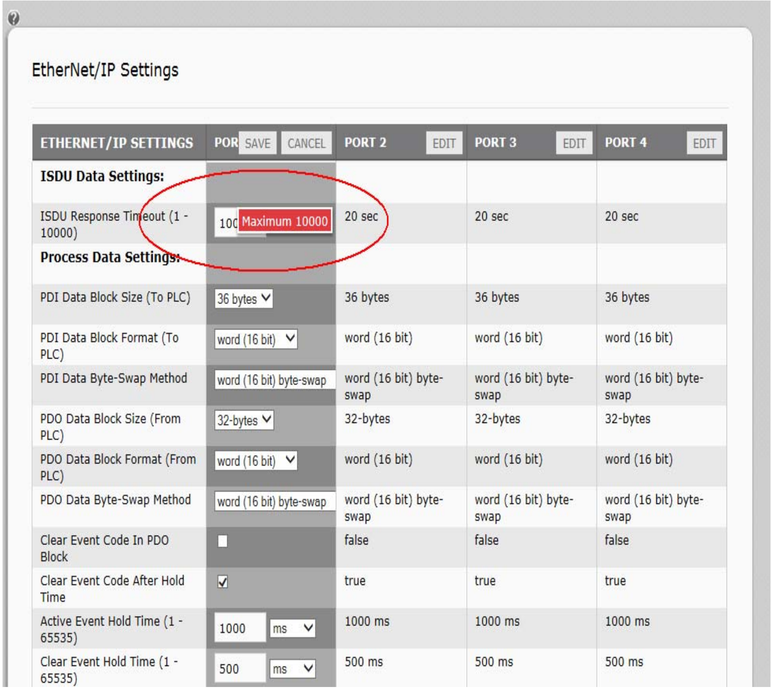
You can use this procedure to configure EtherNet/IP characteristics for each port.

1. If necessary, open the IO-Link Master web interface with your web browser using the IP address.
2. Click **Configuration** in the menu bar.
3. Click the **ETHERNET/IP SETTINGS** submenu.
4. Click the **EDIT** button for the port that you want to configure.
5. Make appropriate selections for the IO-Link device that you will connect to that port.

   You can use the help system if you require definitions or values for the options or *4.2.2. EtherNet/IP Settings Parameters* on Page 67.
6. Scroll to the top of the page and click the **SAVE** button.

   Make sure that the port now displays the **EDIT** button.

   If it displays the **SAVE** and **CANCEL** buttons, that means that one of the parameters contains an incorrect value. If necessary, scroll down the page, make the needed corrections, and click **SAVE**.



EtherNet/IP Settings

ETHERNET/IP SETTINGS	POR SAVE CANCEL	PORT 2 EDIT	PORT 3 EDIT	PORT 4 EDIT
ISDU Data Settings:				
ISDU Response Timeout (1 - 10000)	100 Maximum 10000	20 sec	20 sec	20 sec
Process Data Settings:				
PDI Data Block Size (To PLC)	36 bytes	36 bytes	36 bytes	36 bytes
PDI Data Block Format (To PLC)	word (16 bit)	word (16 bit)	word (16 bit)	word (16 bit)
PDI Data Byte-Swap Method	word (16 bit) byte-swap	word (16 bit) byte-swap	word (16 bit) byte-swap	word (16 bit) byte-swap
PDO Data Block Size (From PLC)	32-bytes	32-bytes	32-bytes	32-bytes
PDO Data Block Format (From PLC)	word (16 bit)	word (16 bit)	word (16 bit)	word (16 bit)
PDO Data Byte-Swap Method	word (16 bit) byte-swap	word (16 bit) byte-swap	word (16 bit) byte-swap	word (16 bit) byte-swap
Clear Event Code In PDO Block	☐	false	false	false
Clear Event Code After Hold Time	☑	true	true	true
Active Event Hold Time (1 - 65535)	1000 ms	1000 ms	1000 ms	1000 ms
Clear Event Hold Time (1 - 65535)	500 ms	500 ms	500 ms	500 ms

7. Repeat for each port that requires configuration changes.

4.2.2. EtherNet/IP Settings Parameters

The *EtherNET/IP Settings* configuration page supports the following options.

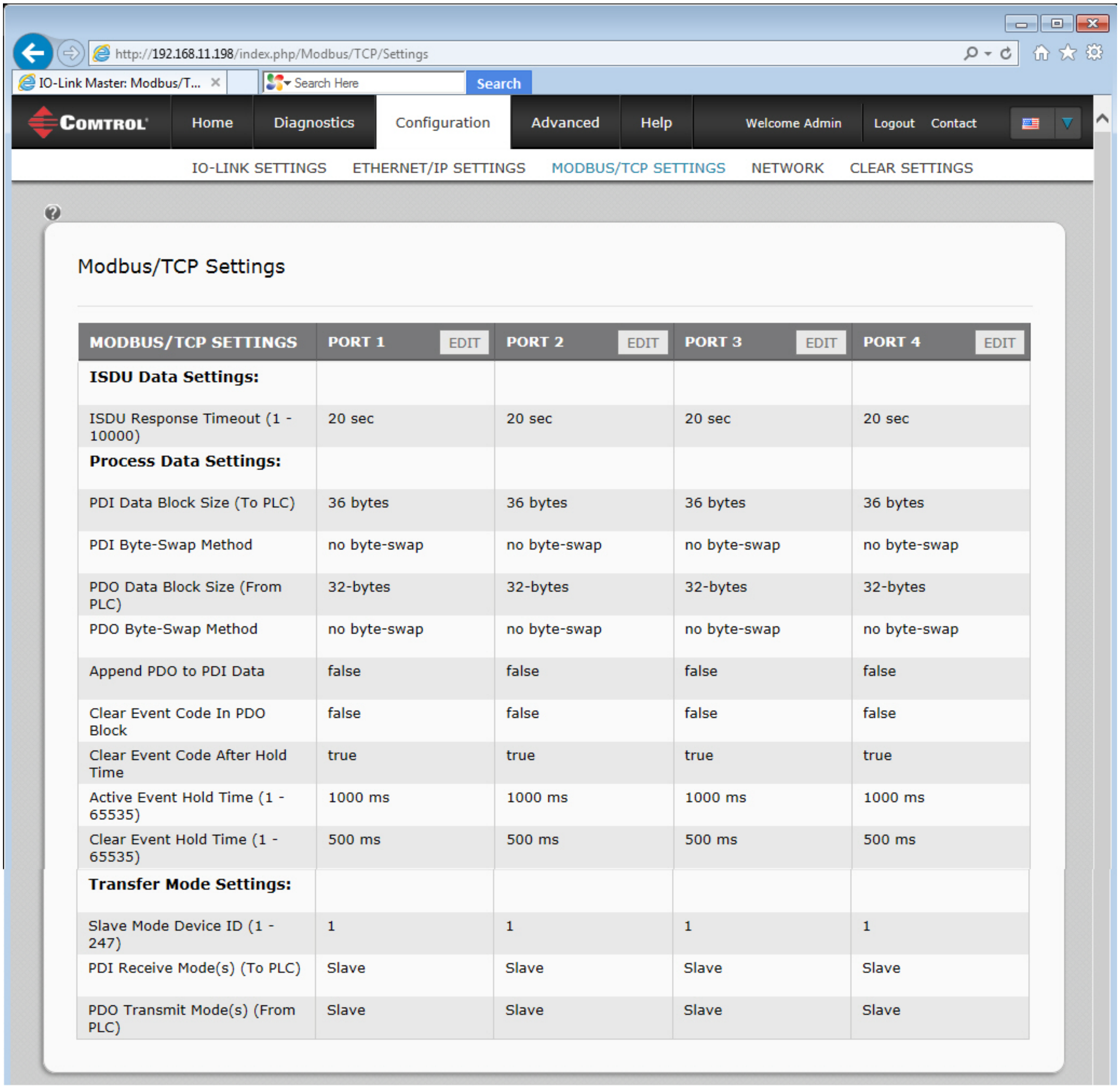
EtherNet/IP Settings Page	
ISDU Data Settings	
ISDU Response Timeout *Default*: 20 seconds	The time that the IO-Link Master's EtherNet/IP interface waits for a response to an ISDU request. The timeout needs to set long enough to allow all commands within the ISDU request to be processed. Valid range: 1-10,000 seconds
Process Data Settings	
PDI Data Block Size (To PLC) *Default*: 36-bytes	The configurable PDI data block length. Supported optional lengths are: • 4-bytes (header only) • 8-bytes (4 bytes data) • 16-bytes (12 bytes data) • 24-bytes (20 bytes data) • 36-bytes (32 bytes data)
PDI Data Block Format (To PLC) *Default*: Word-16	Data format of PDI data block to be transferred to the PLC(s) in Class 1 and/or Write-to-Tag/File PDI Transfer Modes. Supported formats are: • Byte-8 (8-bit or SINT) • Word-16 (16-bit or INT) • Dword-32 (32-bit or DINT) *Note: The Data Block Format is independent of the PDI Data Byte-Swap Method.* *This setting is not used for the SLC, PLC-5 and MicroLogix PLCs which are always Word-16.*
PDI Data Byte-Swap Method *Default*: Work (16-bit) byte swap	If enabled, the IO-Link Master swaps the data bytes in word (2 byte) format or dword (4 byte) format. Supported values are: • No byte-swap – data passed through as received • Word (16-bit) byte-swap – data is byte-swapped in word format • Dword (32-bit) byte-swap – data is byte-swapped in dword format *Note: The byte swapping must be set correctly in order to convert from IO-Link (big-endian byte order), to EtherNet/IP (little-endian byte order).*

EtherNet/IP Settings Page (Continued)	
PDO Data Block Size (From PLC) *Default*: 32-bytes	The configurable PDO data block length. Supported optional lengths are: • Event code not included: - 4-bytes = all data - 8-bytes = all data - 16-bytes = all data - 24-bytes = all data - 32-bytes = all data - 34-bytes = 32 bytes data, 2 pad bytes - 36-bytes = 32 bytes data, 4 pad bytes • Event code included - PDO Data Format = Byte8: - 4-bytes = 2 byte event code, 2 data bytes - 8-bytes = 2 byte event code, 6 data bytes - 16-bytes = 2 byte event code, 14 data bytes - 24-bytes = 2 byte event code, 22 data bytes - 32-bytes = 2 byte event code, 30 data bytes - 34-bytes = 2 byte event code, 32 data bytes - 36-bytes = 2 byte event code, 32 data bytes, 2 byte pad • Event code included - PDO Data Format = word (16-bit): - 4-bytes = event code word, data word - 8-bytes = event code word, 3 data words - 16-bytes = event code word, 7 data words - 24-bytes = event code word, 11 data words - 32-bytes = event code word, 15 data words - 34-bytes = event code word, 16 data words - 36-bytes = event code word, 16 data words, pad word • Event code included - PDO Data Format = dword (32-bit): - 4-bytes = event code dword - 8-bytes = event code dword, data dword - 16-bytes = event code dword, 3 data dwords - 24-bytes = dword event code, 5 data dwords - 32-bytes = dword event code, 7 data dwords - 34-bytes = dword event code, 7 data dwords, 2 data bytes - 36-bytes = dword event code, 8 data dwords
PDO Data Block Format (From PLC) *Default*: Word-16	Data format of PDO data block received from the PLC(s) in Class 1 or Read from TagOrFile PDO Transfer Modes. Formats include: • Byte-8 (8-bit) • Word-16 (16-bit) • Dword-32 (32-bit) *Note: The Data Block Format is independent of the PDO Data Byte-Swap Method.* *This setting is not used for the SLC, PLC-5 and MicroLogix PLCs which are always Word-16.*

EtherNet/IP Settings Page (Continued)	
PDO Data Byte-Swap Method *Default*: Word (16-bit) byte-swap	If enabled, the IO-Link Master swaps the data bytes in word (2 byte) format or dword (4 byte) format. Supported values are: • No byte-swap – data passed through as received • Word (16-bit) byte-swap – data is byte-swapped in word format • Dword (32-bit) byte-swap – data is byte-swapped in dword format *Note: The byte swapping must be set correctly in order to convert from EtherNet/IP (little-endian byte order), to IO-Link (big-endian byte order).*
Clear Event Code in PDO Block *Default*: False	If enabled, the IO-Link Master expects the first 2 bytes, word, or dword of the PDO block to be used for event code handling. Supported values are: • True = expect event code • False = no event code, expect only PDO data
Clear Event Code After Hold Time *Default*: True	If enabled, the IO-Link Master clears any event code reported in the PDI data block after the **Event Active Hold Time**. Supported values are: • True = clear event code after hold time • False = do not clear event code after hold time
Event Active Hold Time *Default*: 1000 ms	If **Clear Event Code After Hold** time is enabled, the time period an event code is reported in the PDI block before it is cleared. • Valid range: 1-65535 • Valid Units: - ms (milliseconds) - sec (seconds) - min (minutes) - hours - days
Clear Event Hold Time *Default*: 500 ms	Once an event code has been cleared, the time an event code stays cleared in the PDI block before another event code can be reported. • Valid range: 1-65535 • Valid Units: - ms (milliseconds) - sec (seconds) - min (minutes) - hours - days

EtherNet/IP Settings Page (Continued)	
Transfer Mode Settings	
PDI Receive Mode(s) *Default*: Polling, Class1	Determines which PDI Receive (To PLC) Modes are enabled. Supported modes are: • Polling • Class1 • Write-to-TagOrFile
PDO Transmit Mode *Default*: Class 1	Supported modes are: • Off • PLC-Writes • Class1 • Read-from-TagOrFile
Read / Write Tag / File Settings	
PLC IP Address *Default*: 0.0.0.0	The PLC IP Address is required if either **Write-to-TagOrFile** or **Read-from-TagOrFile** mode are enabled. Format: xxx.xxx.xxx.xxx
PLC Controller Slot Number *Default*: 0	The PLC Controller Slot Number is required if either **Write-to-TagOrFile** or **Read-from-TagOrFile** mode are enabled. Valid range: 0-64
PLC Type *Default*: ControlLogix	Indicates the type of PLC that the tag(s) or file(s) are written to and/or read from. Supported PLC Types are: • ControlLogix • SLC • PLC-5 • MicroLogix
Write PDI to Tag / File Settings	
PDI Tag/File Name *Default*: *blank*	The tag or file name to place the PDI data block. • ControlLogix family: - Tags must be same type as **PDI Data Format** (SINT, INT or DINT). - Tags must be an array. - Tags must be at least as long as the **PDI Data Block Length**. • SLC/PLC-5/MicroLogix: - Files must be of INTEGER (16-bit) type. - Files must be named with standard file name conventions (i.e: N10:0, N21:30, etc) - The file must be at least as long as the **PDI Data Block Length**.
Append PDO to PDI Data *Default*: False	If selected, the IO-Link Master appends any PDO data to the end of the PDI data. • False = Do not append PDO data • True = Append PDO data

EtherNet/IP Settings Page (Continued)	
Maximum PLC Update Rate *Default*: 40ms	The maximum rate at which the IO-Link Master updates the PDI tag or file. This parameter is used to ensure that the PLC receives all state changes. Setting the update rate to 10 ms effectively disables this feature. The valid range is 10 to 65535 ms.
Heartbeat Update Enable *Default*: False	If selected, the IO-Link Master updates the PDI data block at the **Heartbeat Update Rate**. • False = Heartbeat update disabled • True = Heartbeat update enabled
Heartbeat Update Rate *Default*: 1000ms	If **Heartbeat Update Enable** is selected, the rate at which the IO-Link Master updates the PDI data block in the **Write-to-Tag/File** mode. The valid range is 50 to 65535 ms.
Read PDO from Tag/File Settings	
PDO Tag/File Name *Default*: *blank*	The tag or file name that the IO-Link Master reads the PDO data block from. • ControlLogix family: - Tags must be same type as **PDO Data Format** (SINT, INT or DINT). - Tags must be an array. - Tags must be at least as long as the **PDO Data Block Length**. • SLC/PLC-5/MicroLogix: - Files must be of INTEGER (16-bit) type. - Files must be named with standard file name conventions (i.e: N10:0, N21:30, etc) The file must be at least as long as the **PDO Data Block Length**.
PLC Poll Rate *Default*: 1000ms	The frequency which the IO-Link Master reads the PDO data block in the Read-from-Tag/File mode. Valid range: 50-65535 ms



Modbus/TCP Settings

MODBUS/TCP SETTINGS	PORT 1 EDIT	PORT 2 EDIT	PORT 3 EDIT	PORT 4 EDIT
ISDU Data Settings:				
ISDU Response Timeout (1 - 10000)	20 sec	20 sec	20 sec	20 sec
Process Data Settings:				
PDI Data Block Size (To PLC)	36 bytes	36 bytes	36 bytes	36 bytes
PDI Byte-Swap Method	no byte-swap	no byte-swap	no byte-swap	no byte-swap
PDO Data Block Size (From PLC)	32-bytes	32-bytes	32-bytes	32-bytes
PDO Byte-Swap Method	no byte-swap	no byte-swap	no byte-swap	no byte-swap
Append PDO to PDI Data	false	false	false	false
Clear Event Code In PDO Block	false	false	false	false
Clear Event Code After Hold Time	true	true	true	true
Active Event Hold Time (1 - 65535)	1000 ms	1000 ms	1000 ms	1000 ms
Clear Event Hold Time (1 - 65535)	500 ms	500 ms	500 ms	500 ms
Transfer Mode Settings:				
Slave Mode Device ID (1 - 247)	1	1	1	1
PDI Receive Mode(s) (To PLC)	Slave	Slave	Slave	Slave
PDO Transmit Mode(s) (From PLC)	Slave	Slave	Slave	Slave

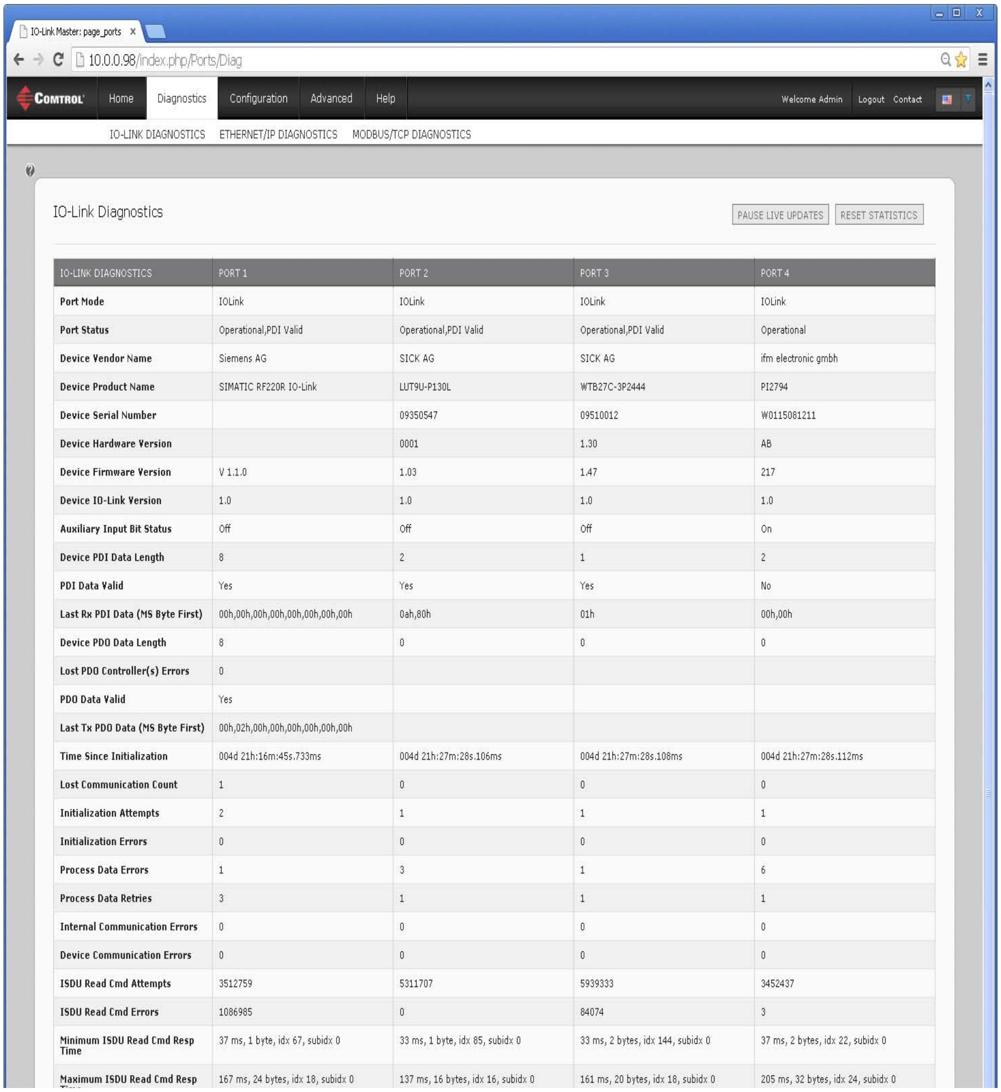
Chapter 5. Using the Diagnostics Pages

This section provides information about the following **Diagnostics** web pages.

- *5.1. IO-Link Port Diagnostics*
- *5.2. EtherNet/IP Diagnostics* on Page 76

5.1. IO-Link Port Diagnostics

The *IO-Link Diagnostics* page may be useful when trying to troubleshoot port issues related to IO-Link configuration.



IO-Link Diagnostics

IO-LINK DIAGNOSTICS	PORT 1	PORT 2	PORT 3	PORT 4
Port Mode	IOLink	IOLink	IOLink	IOLink
Port Status	Operational,PDI Valid	Operational,PDI Valid	Operational,PDI Valid	Operational
Device Vendor Name	Siemens AG	SICK AG	SICK AG	ifm electronic gmbh
Device Product Name	SIMATIC RF220R IO-Link	LUT9U-P130L	WTB27C-3P2444	PI2794
Device Serial Number		09350547	09510012	W0115081211
Device Hardware Version		0001	1.30	AB
Device Firmware Version	V 1.1.0	1.03	1.47	217
Device IO-Link Version	1.0	1.0	1.0	1.0
Auxiliary Input Bit Status	Off	Off	Off	On
Device PDI Data Length	8	2	1	2
PDI Data Valid	Yes	Yes	Yes	No
Last Rx PDI Data (MS Byte First)	00h,00h,00h,00h,00h,00h,00h,00h	0ah,80h	01h	00h,00h
Device PDO Data Length	8	0	0	0
Lost PDO Controller(s) Errors	0			
PDO Data Valid	Yes			
Last Tx PDO Data (MS Byte First)	00h,02h,00h,00h,00h,00h,00h,00h			
Time Since Initialization	004d 21h:16m:45s.733ms	004d 21h:27m:28s.106ms	004d 21h:27m:28s.108ms	004d 21h:27m:28s.112ms
Lost Communication Count	1	0	0	0
Initialization Attempts	2	1	1	1
Initialization Errors	0	0	0	0
Process Data Errors	1	3	1	6
Process Data Retries	3	1	1	1
Internal Communication Errors	0	0	0	0
Device Communication Errors	0	0	0	0
ISDU Read Cmd Attempts	3512759	5311707	5939333	3452437
ISDU Read Cmd Errors	1086985	0	84074	3
Minimum ISDU Read Cmd Resp Time	37 ms, 1 byte, idx 67, subidx 0	33 ms, 1 byte, idx 85, subidx 0	33 ms, 2 bytes, idx 144, subidx 0	37 ms, 2 bytes, idx 22, subidx 0
Maximum ISDU Read Cmd Resp	167 ms, 24 bytes, idx 18, subidx 0	137 ms, 16 bytes, idx 16, subidx 0	161 ms, 20 bytes, idx 18, subidx 0	205 ms, 32 bytes, idx 24, subidx 0

Note: *This image does not illustrate the complete Diagnostics page.*

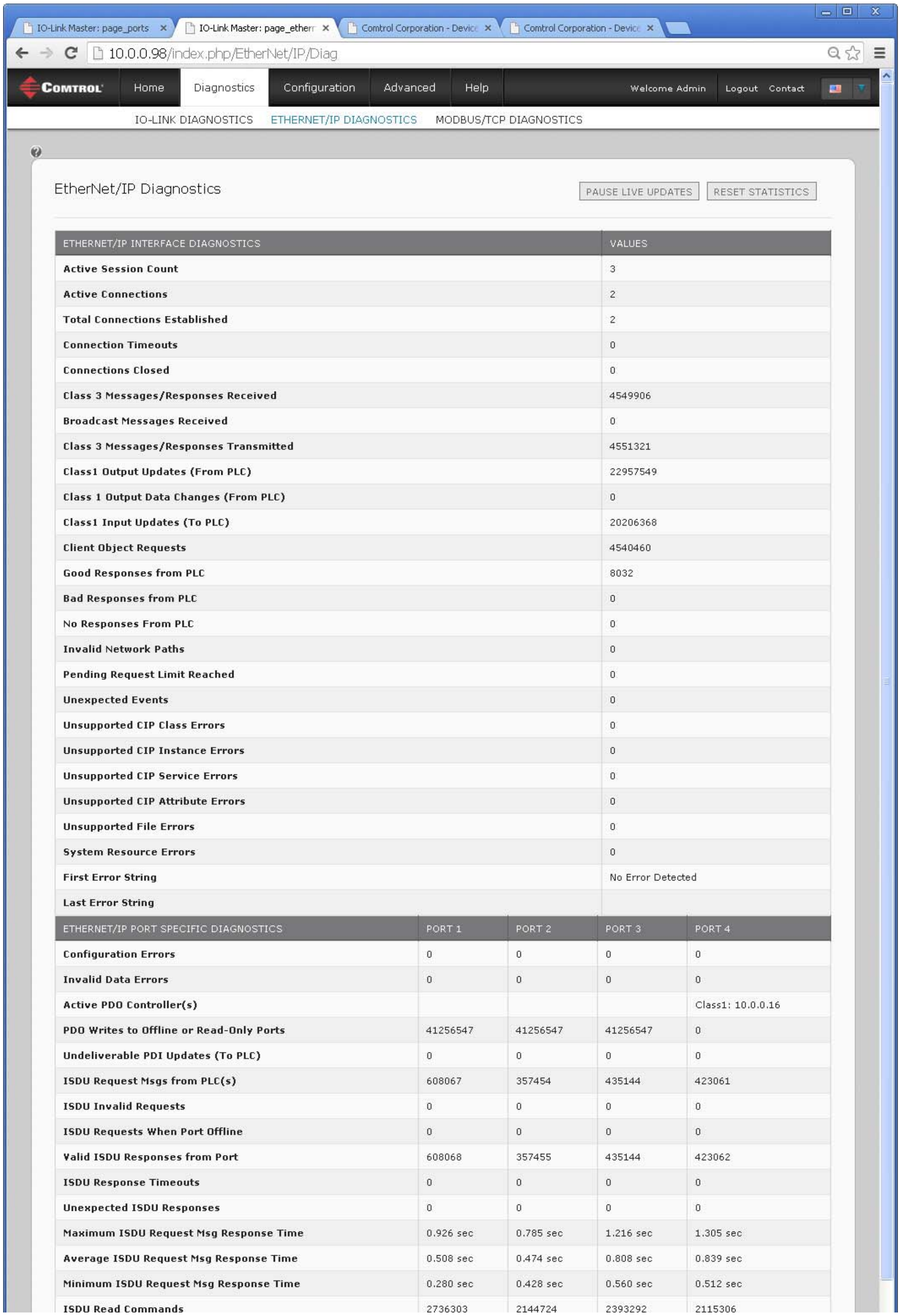
The following table provides information about the *IO-Link Diagnostics* page.

IO-Link Diagnostics	
Port Mode	Displays the active device mode: • Reset = The port is configured to disable all functionality. • IO-Link = The port is configured to IO-Link mode. • Digital In = The port is configured to operate as a digital input. • Digital Out = The port is configured to operate as a digital output.
Port Status	Displays the port status: • Inactive = The port is in active state. Typically, this indicates that the device is either not attached or not detected. • Initializing = The port is in the process of initializing. • Operational = The port is operational and, if in IO-Link mode, communications to the IO-Link device has been established. • PDI Valid = The PDI data is now valid. • Fault = The port has detected a fault and is unable to re-establish communications.
Device Vendor Name	Displays the Device Vendor Name as stored in ISDU Index 16.
Device Product Name	The Device Product Name as stored in ISDU Index 18.
Device Serial Number	The Device Serial Number as stored in ISDU Index 21.
Device Hardware	The Device Hardware Version as stored in ISDU Index 22.
Device Firmware	The Device Firmware Version as stored in ISDU Index 23.
Device IO-Link Version	The supported Device IO-Link Version as stored in ISDU Index 0.
Auxiliary Bit Status	The current status of the auxiliary bit as received on Pin 2 of the IO-Link port.
Last Rx PDI Data (MS Byte First)	The last Rx PDI data as received from the IO-Link device.
Device PDO Data Length	The supported Device PDO Data Length, in bytes, as stored in ISDU Index 0.
Lost PDO Controller(s) Errors	The number of times that the PDO controller(s) were present and then lost connection.
PDO Data Valid	Status of PDO data being received from controller(s).
Device PDI Data Length	The supported Device PDI Data Length, in bytes, as stored in ISDU Index 0.
PDI Data Valid	Current status of PDI data as received from the IO-Link device.
Last Tx PDO Data	The last Tx PDO data.
Time Since Initialization	The time since the last port initialization.
Lost Communication Count	The number of times that communication has been lost to the IO-Link device.
Initialization Attempts	The number of times the IO-Link port was initialized.
Initialization Errors	The number of port initialization errors that occurred.
Process Data Errors	The number of process data errors the port received.
Process Data Retries	The number of process data retries the port performed.
Internal Communication Errors	The number of IO-Link Master internal communication errors that occurred on this port.
Device Communication Errors	The number of device specific communication errors that occurred.

IO-Link Diagnostics (Continued)	
ISDU Read Cmd Attempts	The number of read ISDU command attempts.
ISDU Read Cmd Errors	The number of read ISDU command errors.
Minimum ISDU Read Cmd Resp Time	The minimum, or shortest, read ISDU command response time.
Maximum ISDU Read Cmd Resp Time	The maximum, or longest, read ISDU command response time.
Average ISDU Read Cmd Resp Time	The average ISDU read command response time.
Average ISDU Read Cmd Byte Time	The average per-byte read ISDU command response time.
ISDU Write Cmd Attempts	The number of write ISDU command attempts.
ISDU Write Cmd Errors	The number of write ISDU command errors.
Minimum ISDU Write Cmd Resp Time	The minimum, or shortest, write ISDU command response time.
Maximum ISDU Write Cmd Resp Time	The maximum, or longest, write ISDU command response time.
Average ISDU Write Cmd Resp Time	The average ISDU write command response time.
Average ISDU Write Cmd Byte Time	The average per-byte ISDU write command response time.
Total Events	The total number of events that were received on this port.
First Events	Up to the first, or oldest, three events that were received on this port.
Last Events	Up to the last, or most recent, three events that were received on this port.

5.2. EtherNet/IP Diagnostics

The *EtherNet/IP Diagnostics* page may be useful when trying to troubleshoot EtherNet/IP communications and port issues related to EtherNet/IP configuration.



EtherNet/IP Diagnostics

ETHERNET/IP INTERFACE DIAGNOSTICS	VALUES
Active Session Count	3
Active Connections	2
Total Connections Established	2
Connection Timeouts	0
Connections Closed	0
Class 3 Messages/Responses Received	4549906
Broadcast Messages Received	0
Class 3 Messages/Responses Transmitted	4551321
Class1 Output Updates (From PLC)	22957549
Class 1 Output Data Changes (From PLC)	0
Class1 Input Updates (To PLC)	20206368
Client Object Requests	4540460
Good Responses from PLC	8032
Bad Responses from PLC	0
No Responses From PLC	0
Invalid Network Paths	0
Pending Request Limit Reached	0
Unexpected Events	0
Unsupported CIP Class Errors	0
Unsupported CIP Instance Errors	0
Unsupported CIP Service Errors	0
Unsupported CIP Attribute Errors	0
Unsupported File Errors	0
System Resource Errors	0
First Error String	No Error Detected
Last Error String	

ETHERNET/IP PORT SPECIFIC DIAGNOSTICS	PORT 1	PORT 2	PORT 3	PORT 4
Configuration Errors	0	0	0	0
Invalid Data Errors	0	0	0	0
Active PDO Controller(s)				Class1: 10.0.0.16
PDO Writes to Offline or Read-Only Ports	41256547	41256547	41256547	0
Undeliverable PDI Updates (To PLC)	0	0	0	0
ISDU Request Msgs from PLC(s)	608067	357454	435144	423061
ISDU Invalid Requests	0	0	0	0
ISDU Requests When Port Offline	0	0	0	0
Valid ISDU Responses from Port	608068	357455	435144	423062
ISDU Response Timeouts	0	0	0	0
Unexpected ISDU Responses	0	0	0	0
Maximum ISDU Request Msg Response Time	0.926 sec	0.785 sec	1.216 sec	1.305 sec
Average ISDU Request Msg Response Time	0.508 sec	0.474 sec	0.808 sec	0.839 sec
Minimum ISDU Request Msg Response Time	0.280 sec	0.428 sec	0.560 sec	0.512 sec
ISDU Read Commands	2736303	2144724	2393292	2115306

Note: *This image does not illustrate the complete Diagnostics page.*

The following table provides information about the *EtherNet/IP Diagnostics* page.

EtherNet/IP Diagnostics	
Active Session Count	The number of active Ethernet/IP sessions. A session can: • Support both Class 1 I/O and Class 3 Messages • Can be initiated by either the PLC or the IO-Link Master • Can be terminated by either the PLC or the IO-Link Master
Active Connections	The current number of active connections (both Class 1 and 3).
Total Connections Established	The total number of connections that have been established.
Connection Timeouts	The number of connections that have closed due to timing out.
Connections Closed	The number connections that have closed due to a standard processes.
Class 3 Messages/ Responses Received	The number of Class 3 messages and responses received from the PLC or PLCs.
Broadcast Messages Received	The number of broadcast messages received from PLC or PLCs.
Class 3 Messages/ Responses Transmitted	The number of Class 3 messages and responses sent to the PLC or PLCs.
Class 1 Output Updates (From PLC)	The number of Class 1 output data updates received from the PLC or PLCs.
Class 1 Output Data Changes (From PLC)	The number of changes in Class 1 output data received from the PLC.
Class 1 Input Data Updates (To PLC)	The number of Class 1 input data updates sent to the PLC or PLCs.
Client Object Requests	The number of Class 3 requests to the IO-Link Master vendor specific objects.
Good Responses from PLC	The number of good responses from messages sent to PLC or PLCs.
Bad Responses from PLC	Displays the number of bad responses from messages sent to the PLC or PLCs. Bad responses are typically returned for such errors as: • Incorrect tag or file names • Incorrect tag or file data types • Incorrect tag or file data sizes • PLC is overloaded and cannot handle the amount of Ethernet traffic • PLC malfunction
No Responses from PLC	Displays the number of no responses from messages sent to the PLC or PLCs. No responses are typically returned for such errors as: • Incorrect IP address • Incorrect PLC configuration • PLC malfunction • PLC is overloaded and cannot handle the amount of Ethernet traffic
Invalid Network Paths	Displays the number of network path errors on messages sent to the PLC or PLCs. These are typically caused by incorrect IP address settings.
Pending Request Limit Reached	Displays the number of pending request limit errors. These errors occur when the PLC is sending a continuous stream of messages to the IO-Link Master faster than the IO-Link Master can process them.
Unexpected Events	Displays the number of unexpected event errors. Unexpected event errors occur when the IO-Link Master receives an unexpected message from the PLC such as an unexpected response or unknown message.

EtherNet/IP Diagnostics (Continued)	
Unsupported CIP Class Errors	Displays the number of unsupported CIP class errors. These errors occur when a message that attempts to access an invalid class is received by the IO-Link Master.
Unsupported CIP Instance Errors	Displays the number of unsupported CIP instance errors. These errors occur when a message that attempts to access an invalid instance is received by the IO-Link Master.
Unsupported CIP Service Errors	Displays the number of unsupported CIP service errors. These errors occur when a message that attempts to access an invalid service is sent to the IO-Link Master.
Unsupported CIP Attribute Errors	Displays the number of unsupported CIP request attribute errors. These errors occur when a message that attempts to access an invalid attribute is sent to the IO-Link Master.
Unsupported File Errors	Displays the number of messages from SLC/PLC-5/MicroLogix PLCs that attempt to access an unsupported file address.
System Resource Errors	Displays the number of system resource errors. These errors indicate a system error on the IO-Link Master such as operating system errors or full message queues. These errors typically occur when the PLC or PLCs are sending messages to the IO-Link Master faster than the IO-Link Master can process them.
First Error String	Text description of the first error that occurred.
Last Error String	Text description of the last error that occurred.
EtherNet/IP Port Specific Diagnostics	
Configuration Errors	Displays the number of improper configuration errors. These errors occur when the IO-Link Master receives a message that cannot be performed due to an invalid configuration.
Invalid Data Errors	Displays the number of invalid message data errors. These errors occur when the IO-Link Master receives a message that cannot be performed due to invalid data.
Active PDO Controller(s)	Lists the controller interface(s) type, (Class 1 or Class 3), and IP address that are controlling the PDO data.
PDO Writes to Offline or Read-Only Ports	Displays the number of PDO write messages that were dropped due to any of the following: • The port is configured in IO-Link mode: - There is no device connected to the port. - The IO-Link device is off-line. - The IO-Link device does not support PDO data. • The PDO Transmit Mode (To PLC) is disabled. • The port is configured in Digital Input mode.
Undeliverable PDI Updates (To PLC)	Displays the number of PDI update messages that could not be delivered to the PLC in the Write-to-Tag/File method. Undeliverable updates may result when: The IO-Link Master cannot complete an Ethernet connection to the PLC. The PDI data is changing faster than the **Maximum PLC Update Rate**.
ISDU Request Msgs From PLC(s)	Displays the number of ISDU request messages received from the PLC(s) or other controllers. These request messages may contain one or multiple ISDU commands.
ISDU Invalid Requests	Displays the number of ISDU requests received over EtherNet/IP with one or more invalid commands.

EtherNet/IP Diagnostics (Continued)	
ISDU Requests When Port Offline	Displays the number of ISDU requests received over EtherNet/IP when the IO-Link port was offline. This can occur when: • The IO-Link port is initializing, such as after start-up. • There is no IO-Link device attached to the port. • The IO-Link device is not responding. • Communication to the IO-Link device has been lost.
Valid ISDU Responses From Port	Displays the number of valid ISDU response messages returned from the IO-Link port interface and available to the PLC(s). The response messages contain results to the ISDU command(s) received in the request message.
ISDU Response Timeouts	Displays the number of ISDU requests that did not receive a response within the configured **ISDU Response Timeout**.
Unexpected ISDU Responses	Displays the number of unexpected ISDU responses. Unexpected responses may occur when an ISDU response is received after the ISDU request has timed out. This typically requires setting the **ISDU Response Timeout** to a longer value.
ISDU Read Commands	Displays the number of ISDU read commands received over EtherNet/IP.
Maximum ISDU Request Msg Response Time	Displays the maximum time period required to process all commands within an ISDU request message. The response is not available until all ISDU command(s) contained in the request have been processed.
Average ISDU Request Msg Response Time	Displays the average time period required to process the ISDU request message(s). The response is not available until all ISDU command(s) contained in the request have been processed.
Minimum ISDU Request Msg Response Time	Displays the minimum time period required to process all commands within an ISDU request message. The response is not available until all ISDU command(s) contained in the request have been processed.
ISDU Write Commands	Displays the number of ISDU write commands received over EtherNet/IP.
ISDU NOP Commands	Displays the number of ISDU NOP (no operation) commands received over EtherNet/IP.

Chapter 6. ControlLogix Family - Example PLC Programs

The example RSLogix 5000 PLC program is intended to provide basic working functionality:

- Through a Class 1 connection, provide a PDI data block with the IO-Link port status, auxiliary bit status and the PDI data.
- Through explicit messages, provide the ability to send both read and write ISDU requests to the IO-Link devices and receive the responses.
- Through explicit messages, provide the Device Information block.

Perform the following steps to run the example PLC program on your ControlLogix family PLC.

1. *6.1. Import the PLC program into RSLogix 5000*
2. *6.2. Configure the Controller* on Page 81
3. *6.3. Add the EtherNet/IP Module Interface* on Page 83
4. *6.4. Configure the Ethernet Module* on Page 85
5. *6.5. Example PLC Program Operation* on Page 90
6. *6.6. User Defined Data Structures* on Page 93

6.1. Import the PLC program into RSLogix 5000

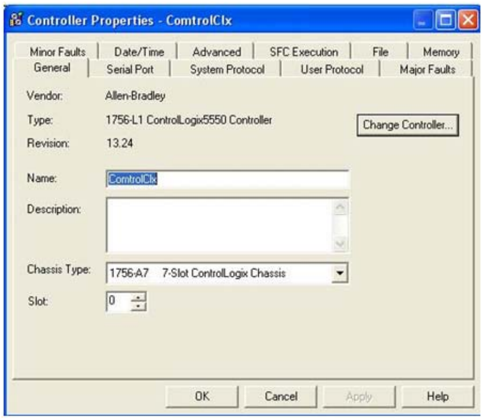
Both the standard **.ACD** file and library file have been provided. If your version of RSLogix 5000 will not open the **.ACD** file, then you will need to import the **.L5K** file.

6.2. Configure the Controller

The following are the controller settings used by Comtrol to create the example PLC program.

Note: *You may need to change the controller settings to match those of your PLC.*

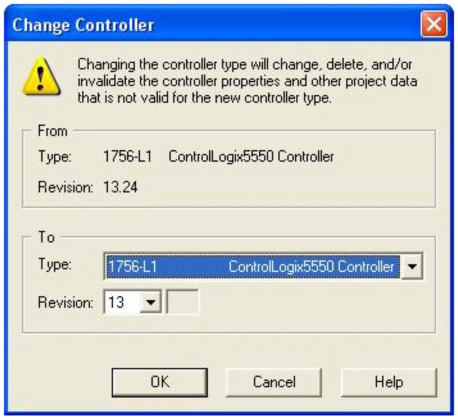
1. Open the RSLogix 5000 *Properties* page, click the *General* tab, enter the name, and click the **Change Controller** button.



Controller Properties - ComtrolClx

Vendor: Allen-Bradley
Type: 1756-L1 ControlLogix5550 Controller
Revision: 13.24
Name: ComtrolClx
Description:
Chassis Type: 1756-A7 7-Slot ControlLogix Chassis
Slot: 0

2. Select the controller type and click **Ok**.



Change Controller

Changing the controller type will change, delete, and/or invalidate the controller properties and other project data that is not valid for the new controller type.

From

Type: 1756-L1 ControlLogix5550 Controller

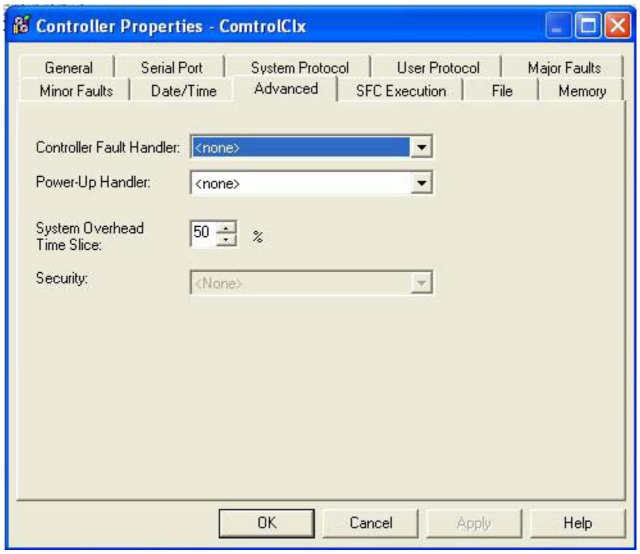
Revision: 13.24

To

Type: 1756-L1 ControlLogix5550 Controller

Revision: 13

3. Set the **System Overhead Time Slice** to 50% and click **Ok**.



Controller Properties - ComtrolClx

Controller Fault Handler: <none>

Power-Up Handler: <none>

System Overhead Time Slice: 50 %

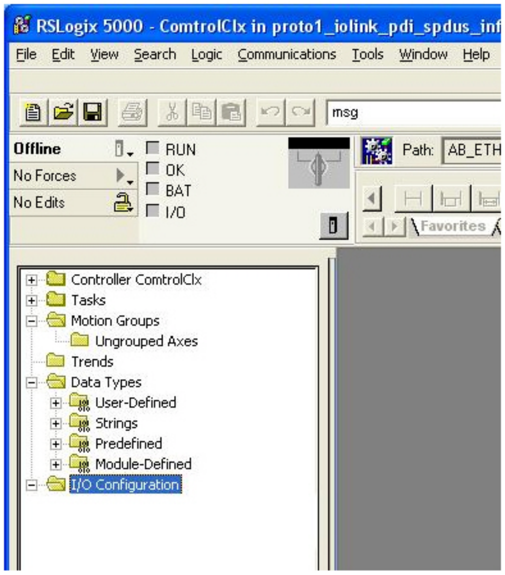
Security: <None>

6.3. Add the EtherNet/IP Module Interface

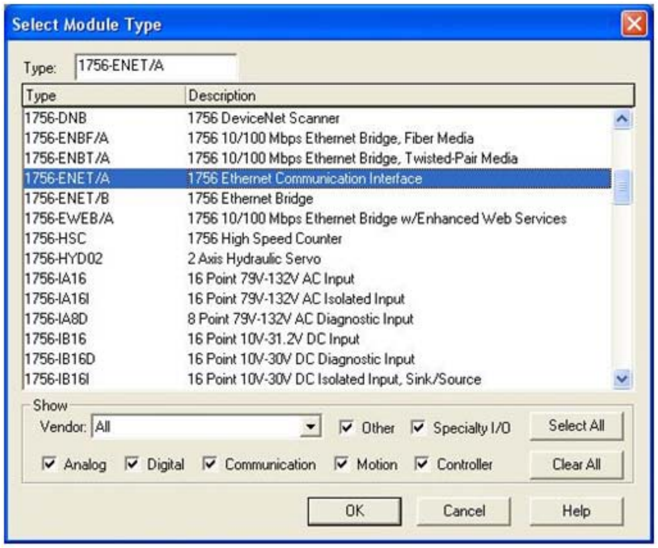
If the controller has been changed or if the Ethernet module is different, you will need to add the EtherNet/IP module to the PLC program.

You can use this procedure to add the Ethernet module for your PLC in the corresponding slot.

1. Click **IO Configuration** and select **New Module**.



2. Select the **Ethernet Module Type** and click **OK**.



Type: 1756-ENET/A

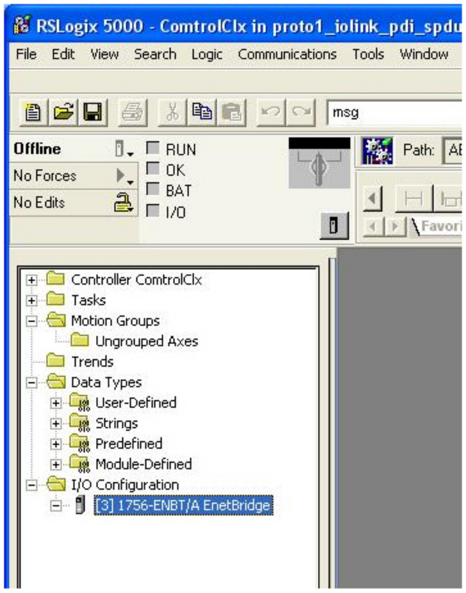
Type	Description
1756-DNB	1756 DeviceNet Scanner
1756-ENBF/A	1756 10/100 Mbps Ethernet Bridge, Fiber Media
1756-ENBT/A	1756 10/100 Mbps Ethernet Bridge, Twisted-Pair Media
1756-ENET/A	1756 Ethernet Communication Interface
1756-ENET/B	1756 Ethernet Bridge
1756-EWEB/A	1756 10/100 Mbps Ethernet Bridge w/Enhanced Web Services
1756-HSC	1756 High Speed Counter
1756-HYD02	2 Axis Hydraulic Servo
1756-IA16	16 Point 79V-132V AC Input
1756-IA16I	16 Point 79V-132V AC Isolated Input
1756-IA8D	8 Point 79V-132V AC Diagnostic Input
1756-IB16	16 Point 10V-31.2V DC Input
1756-IB16D	16 Point 10V-30V DC Diagnostic Input
1756-IB16I	16 Point 10V-30V DC Isolated Input, Sink/Source

Show

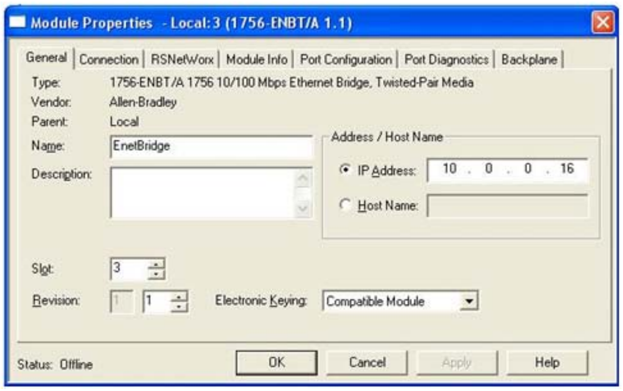
Vendor: All ☑ Other ☑ Specialty I/O

☑ Analog ☑ Digital ☑ Communication ☑ Motion ☑ Controller

3. Right-click the **Ethernet Module** and select **Properties**.



4. Set the **Name**, **IP Address**, **Slot**, and **Revision** for your PLC and then click **OK**.



Type: 1756-ENBT/A 1756 10/100 Mbps Ethernet Bridge, Twisted-Pair Media

Vendor: Allen-Bradley

Parent: Local

Name: EnetBridge

● IP Address: 10 . 0 . 0 . 16

○ Host Name:

Slot: 3

Revision: 1 1

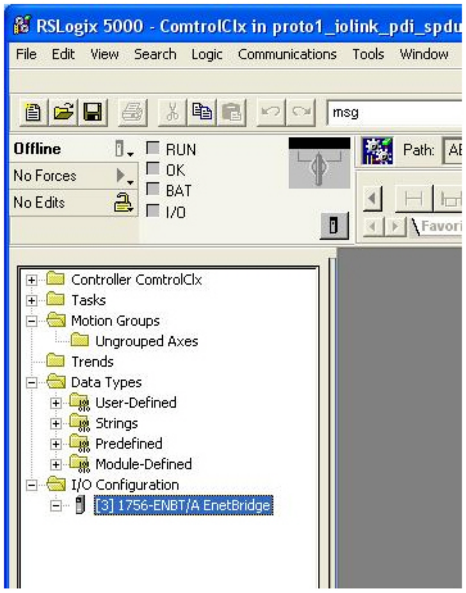
Electronic Keying: Compatible Module

Status: Offline

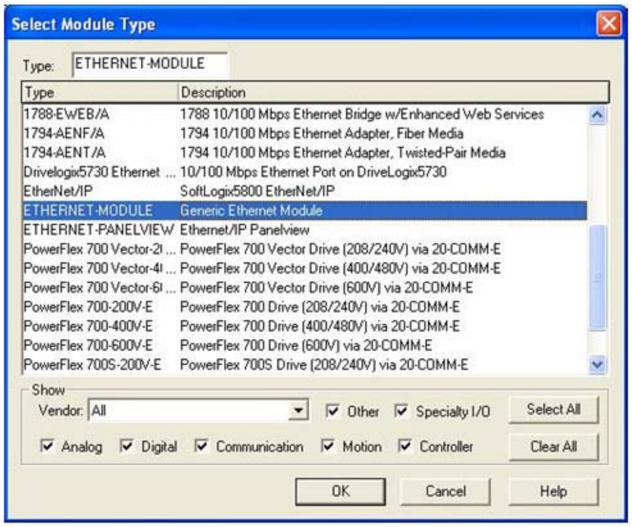
6.4. Configure the Ethernet Module

You can use these procedure as a guideline to configure the Ethernet module.

1. Right-click the **Ethernet interface module** and select **New Module**.



2. Select **ETHERNET-MODULE** Generic Ethernet Module and then click **OK**.



Type: ETHERNET-MODULE

Type	Description
1788-EWEB/A	1788 10/100 Mbps Ethernet Bridge w/Enhanced Web Services
1794-AENF/A	1794 10/100 Mbps Ethernet Adapter, Fiber Media
1794-AENT/A	1794 10/100 Mbps Ethernet Adapter, Twisted-Pair Media
Drivelogix5730 Ethernet ...	10/100 Mbps Ethernet Port on DriveLogix5730
EtherNet/IP	SoftLogix5800 EtherNet/IP
ETHERNET-MODULE	Generic Ethernet Module
ETHERNET-PANELVIEW	Ethernet/IP Panelview
PowerFlex 700 Vector-2I ...	PowerFlex 700 Vector Drive (208/240V) via 20-COMM-E
PowerFlex 700 Vector-4I ...	PowerFlex 700 Vector Drive (400/480V) via 20-COMM-E
PowerFlex 700 Vector-6I ...	PowerFlex 700 Vector Drive (600V) via 20-COMM-E
PowerFlex 700-200V-E	PowerFlex 700 Drive (208/240V) via 20-COMM-E
PowerFlex 700-400V-E	PowerFlex 700 Drive (400/480V) via 20-COMM-E
PowerFlex 700-600V-E	PowerFlex 700 Drive (600V) via 20-COMM-E
PowerFlex 700S-200V-E	PowerFlex 700S Drive (208/240V) via 20-COMM-E

Show

Vendor: All ☑ Other ☑ Specialty I/O

☑ Analog ☑ Digital ☑ Communication ☑ Motion ☑ Controller

3. Enter the following parameters on the *Module Properties* pane.

   a. Enter **IOLinkMstr** for the module **Name**.

   b. If desired, enter a **Description** for the module.

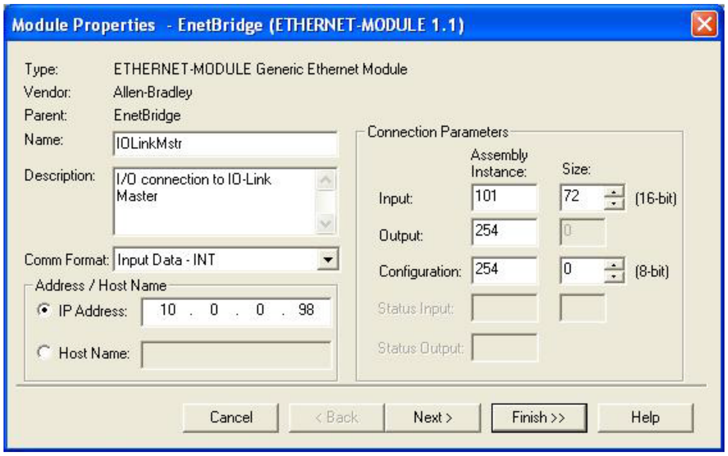
   c. Select **INPUT Data - INT** (16-bit) for the **Comm Format**.

   d. Enter the IP Address of the IO-Link Master module.

   e. Enter the **Connection Parameters**:

   - Enter **101** for the **Input - Assembly Instance**.
   - Enter **72** for the **Input-Size** (input data length in 16-bit words).
   - Enter **254** for the **Output - Assembly Instance**.
   - If not already set to zero, enter **0** for the **Output-Size** (output data length).
   - Set the **Configuration - Assembly Instance** to **254**.
   - Set the **Configuration-Size** to **0**. (There are no configuration parameters).

   f. Click **Next**.



Module Properties - EnetBridge (ETHERNET-MODULE 1.1)

Type: ETHERNET-MODULE Generic Ethernet Module
Vendor: Allen-Bradley
Parent: EnetBridge
Name: IOLinkMstr
Description: I/O connection to IO-Link Master
Comm Format: Input Data - INT

Address / Host Name
● IP Address: 10 . 0 . 0 . 98
○ Host Name:

Connection Parameters

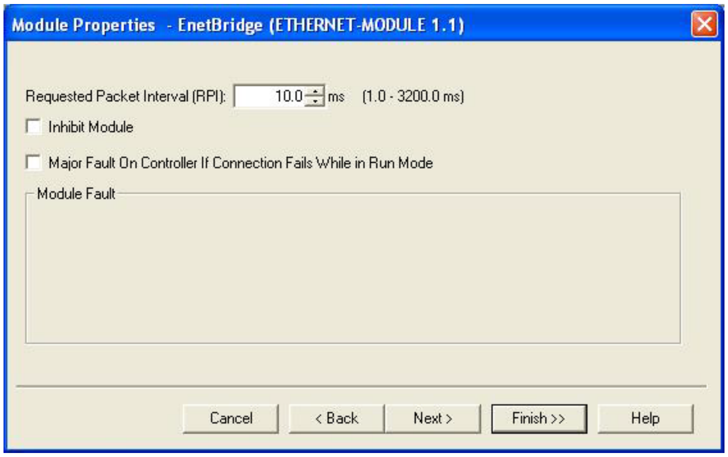
	Assembly Instance:	Size:	
Input:	101	72	(16-bit)
Output:	254	0	
Configuration:	254	0	(8-bit)
Status Input:			
Status Output:			

Note: *Your version of RSLogix 5000 may only allow one Class 1 connection to a specific EtherNet/IP device.*

4. Enter the **Requested Packet Interval**.

   a. Enter the interval value that best suits your system. For the example program, it is recommended to set the interval to **10** ms.

   b. Click **Ok**.



Module Properties - EnetBridge (ETHERNET-MODULE 1.1)

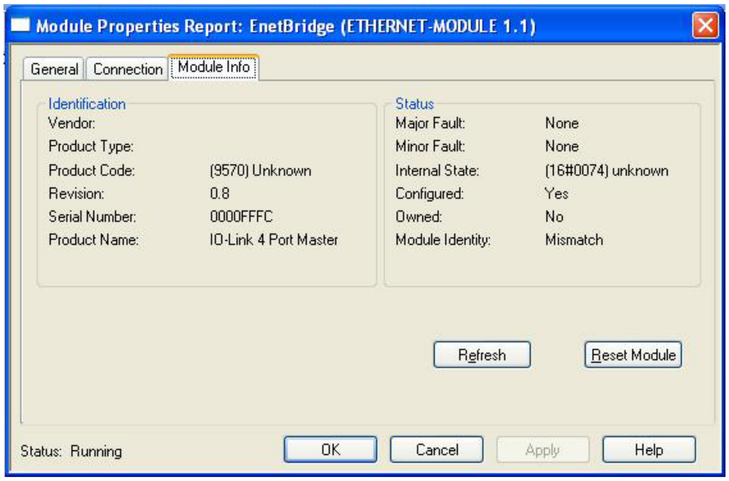
Requested Packet Interval (RPI): 10.0 ms (1.0 - 3200.0 ms)

□ Inhibit Module

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

Module Fault

5. Review the *Module Information* pane.



Module Properties Report: EnetBridge (ETHERNET-MODULE 1.1)

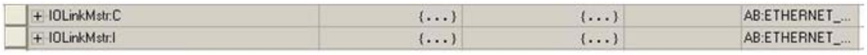
Vendor:	
Product Type:	
Product Code:	(9570) Unknown
Revision:	0.8
Serial Number:	0000FFFC
Product Name:	IO-Link 4 Port Master

Major Fault:	None
Minor Fault:	None
Internal State:	(16#0074) unknown
Configured:	Yes
Owned:	No
Module Identity:	Mismatch

Status: Running

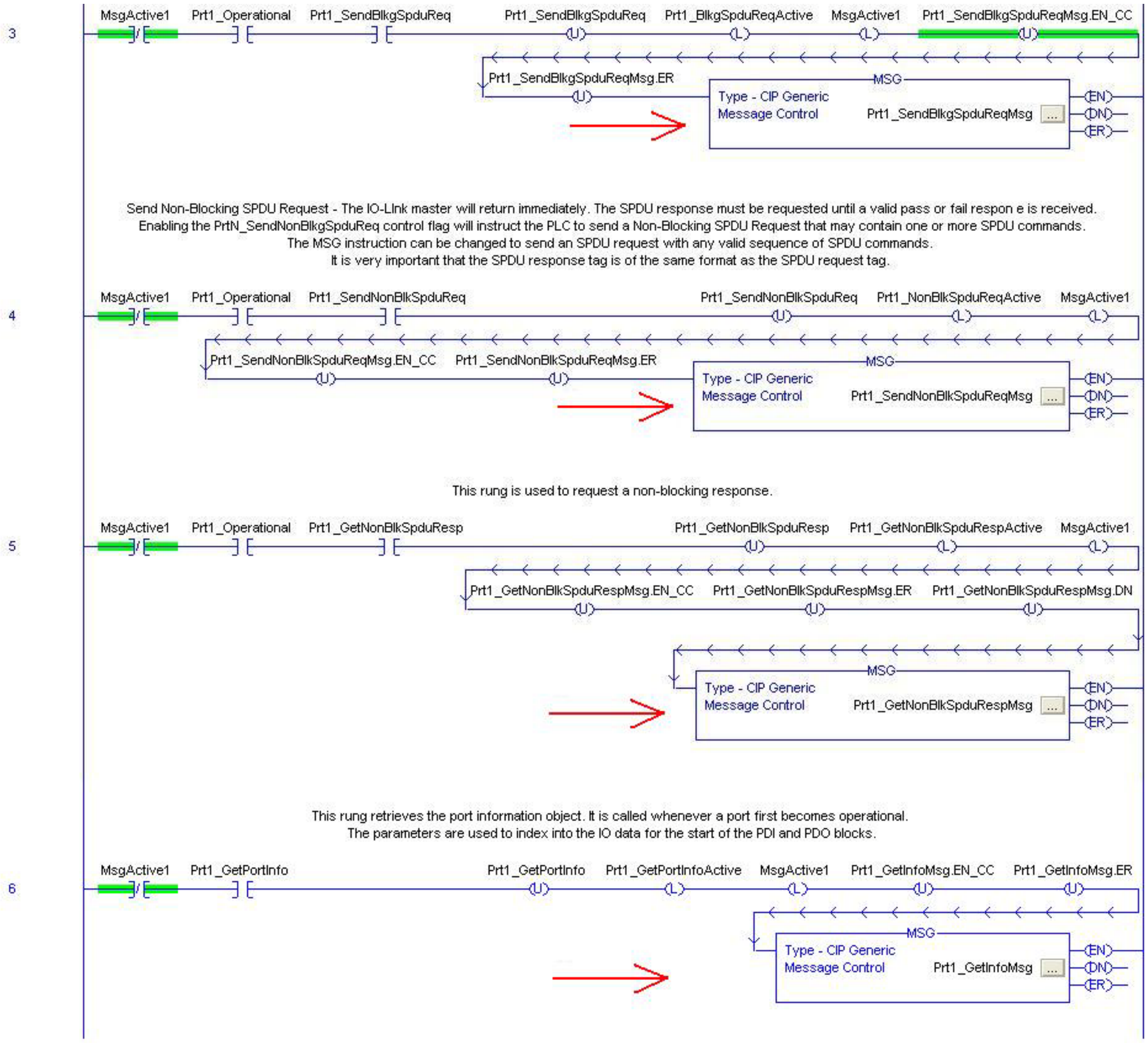
Note: *This pane is not updated until the program is downloaded to the PLC and both PLC and IO-Link Master are running.*

6. Under **Controller Tags**, observe the input tags created for the module. The example PLC program requires the **IOLinkMstr**.I (input data tag). The **IOLinkMstr**.C (configuration tag) is unused and can be ignored.



+ IOLinkMstr:C	{...}	{...}		AB:ETHERNET_...
+ IOLinkMstr:I	{...}	{...}		AB:ETHERNET_...

7. Under **MainProgram**, configure the **Communication Path** for all messages in all four **ProcessIoLinkPortN** subroutines.



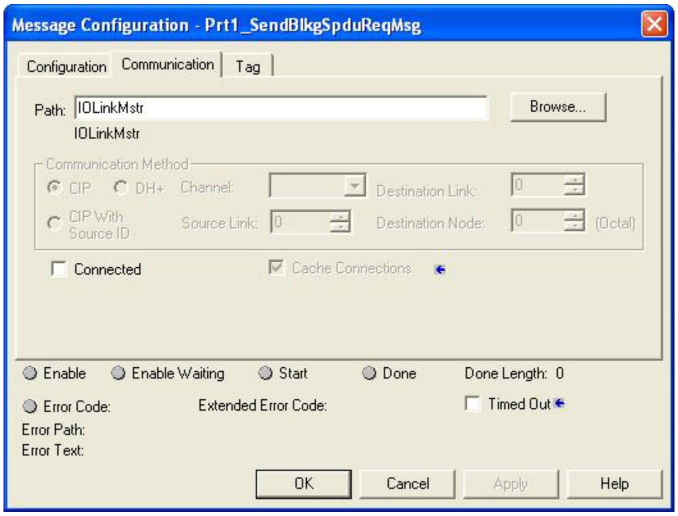
Send Non-Blocking SPDU Request - The IO-LInk master will return immediately. The SPDU response must be requested until a valid pass or fail respon e is received.
Enabling the PrtN_SendNonBlkgSpduReq control flag will instruct the PLC to send a Non-Blocking SPDU Request that may contain one or more SPDU commands.
The MSG instruction can be changed to send an SPDU request with any valid sequence of SPDU commands.
It is very important that the SPDU response tag is of the same format as the SPDU request tag.

This rung is used to request a non-blocking response.

5
MsgActive1
Prt1_Operational
Prt1_GetNonBlkSpduResp
Prt1_GetNonBlkSpduResp
Prt1_GetNonBlkSpduRespActive
MsgActive1
Prt1_GetNonBlkSpduRespMsg.EN_CC
Prt1_GetNonBlkSpduRespMsg.ER
Prt1_GetNonBlkSpduRespMsg.DN
MSG
Type - CIP Generic
Message Control
Prt1_GetNonBlkSpduRespMsg
EN
DN
ER

This rung retrieves the port information object. It is called whenever a port first becomes operational.
The parameters are used to index into the IO data for the start of the PDI and PDO blocks.

8. Enter **IOLinkMstr** for the **Path** for all MSG instructions in all four subroutines.



Message Configuration - Prt1_SendBlkgSpduReqMsg

Path: IOLinkMstr

IOLinkMstr

Communication Method

● CIP ○ DH+ Channel: Destination Link: 0

○ CIP With Source ID Source Link: 0 Destination Node: 0 (Octal)

□ Connected ☑ Cache Connections

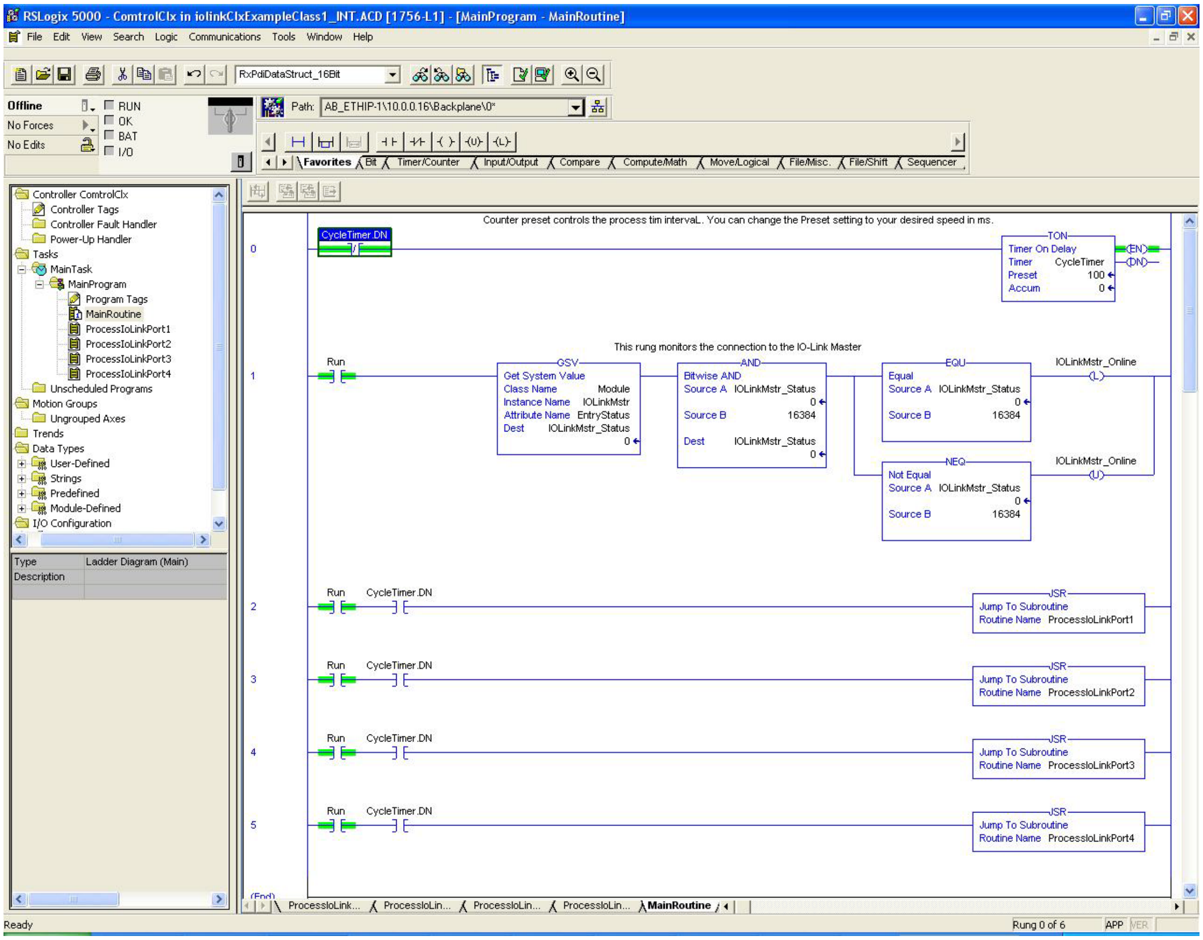
○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

9. Save the RSLogix5000 program.
10. Download to the PLC.
11. Start the PLC.
12. Click **MainRoutine** and review the RSLogix 5000 screen.



RSLogix 5000 - ControlClx in iolinkClxExampleClass1_INT.ACD [1756-L1] - [MainProgram - MainRoutine]

RxPdiDataStruct_16Bit

Offline · No Forces · No Edits · □ RUN □ OK □ BAT □ I/O

Path: AB_ETHIP-1\10.0.0.16\Backplane\0*

Controller ControlClx
- Controller Tags
- Controller Fault Handler
- Power-Up Handler

Tasks
- MainTask
  - MainProgram
    - Program Tags
    - MainRoutine
    - ProcessIoLinkPort1
    - ProcessIoLinkPort2
    - ProcessIoLinkPort3
    - ProcessIoLinkPort4
- Unscheduled Programs

Motion Groups
- Ungrouped Axes

Trends

Data Types
- User-Defined
- Strings
- Predefined
- Module-Defined

I/O Configuration

Type: Ladder Diagram (Main)
Description:

Rung 0: Counter preset controls the process tim intervaL. You can change the Preset setting to your desired speed in ms.
CycleTimer.DN — TON: Timer On Delay, Timer CycleTimer, Preset 100, Accum 0 — (EN) (DN)

Rung 1: This rung monitors the connection to the IO-Link Master
Run — GSV: Get System Value, Class Name Module, Instance Name IOLinkMstr, Attribute Name EntryStatus, Dest IOLinkMstr_Status 0 — AND: Bitwise AND, Source A IOLinkMstr_Status 0, Source B 16384, Dest IOLinkMstr_Status 0 — EQU: Equal, Source A IOLinkMstr_Status 0, Source B 16384 — IOLinkMstr_Online (L); NEQ: Not Equal, Source A IOLinkMstr_Status 0, Source B 16384 — IOLinkMstr_Online (U)

Rung 2: Run, CycleTimer.DN — JSR: Jump To Subroutine, Routine Name ProcessIoLinkPort1

Rung 3: Run, CycleTimer.DN — JSR: Jump To Subroutine, Routine Name ProcessIoLinkPort2

Rung 4: Run, CycleTimer.DN — JSR: Jump To Subroutine, Routine Name ProcessIoLinkPort3

Rung 5: Run, CycleTimer.DN — JSR: Jump To Subroutine, Routine Name ProcessIoLinkPort4

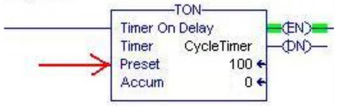
(End)

Ready · Rung 0 of 6 · APP VER

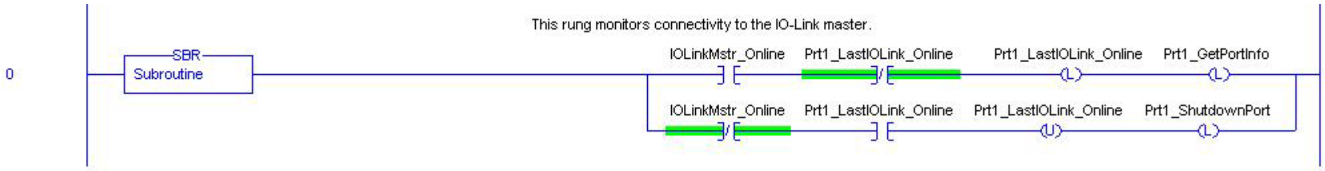
6.5. Example PLC Program Operation

The example PLC Program has been designed to operate with the default IO-Link Master settings. It provides only Input Process data, but can be modified to also transmit PDO data to the IO-Link Master. The PLC program performs the following tasks:

1. The **MainProgram** calls each of the four **ProcessIoLinkPortN** subroutines once every 100 ms. The frequency of these calls can be adjusted by changing the **CycleTimer** Preset value on rung 0.
2. Each **ProcessIoLinkPortN** subroutine is designed to handle all status and communication between the EtherNet/IP controller and one port on the IO-Link Master.

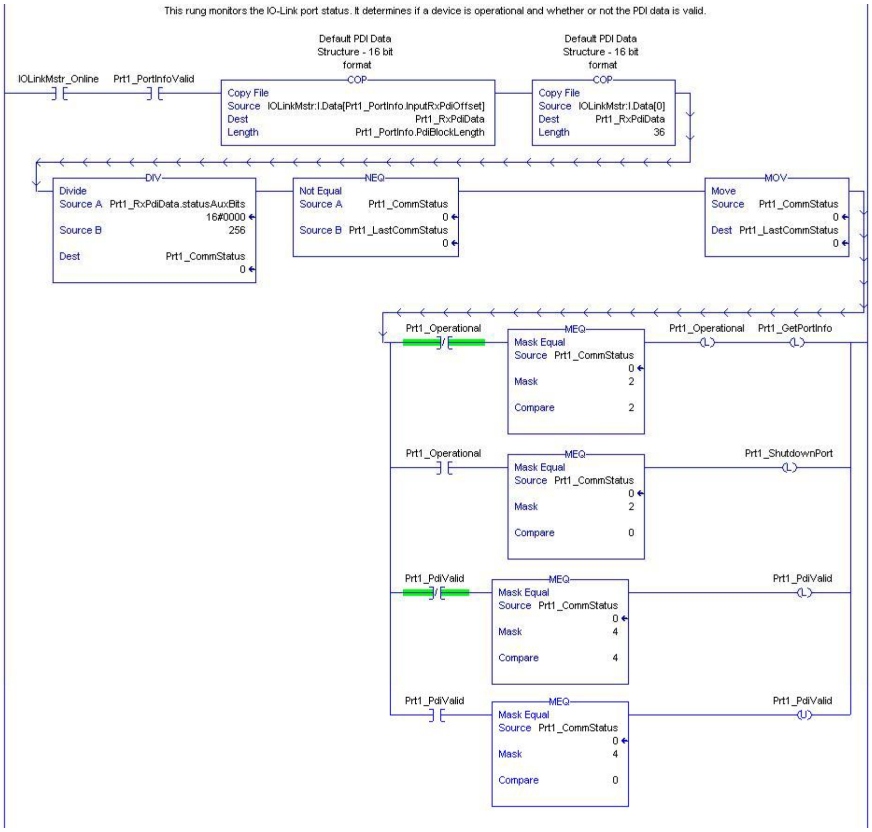


   a. Rung 0:



   This rung monitors the interface to the IO-Link. It sets the flags that control a port initialization or shutdown.

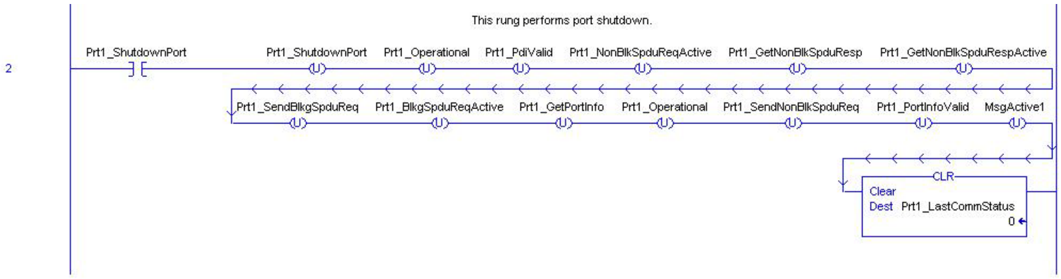
   b. Rung 1:



- Using the parameters received in the **PortInfo** tag, automatically indexes into the input data block.
- Copies the PDI data block into the **PrtN_RxPdiData** tag.
- Monitors the IO-Link port status.

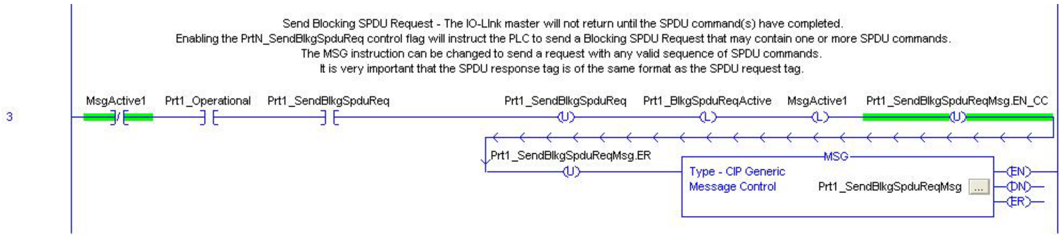
  - When the device status transitions to active (2): The **PrtN_Operational** tag is enabled (latched). This enables explicit message communication to the IO-Link Master on Rungs 3-6.
  - When the device status transitions to inactive (0) or initializing (1): The **PrtN_Shutdown** flag is enabled (latched) which causes a full shutdown of the port.

c. Rung 2:


This rung performs port shutdown.

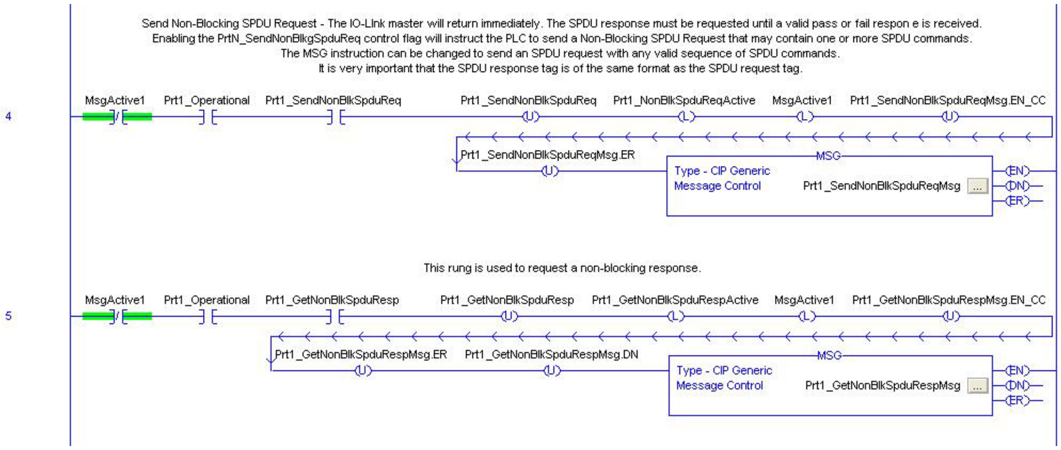
This rung clears all flags necessary to cleanly shut down a port.

d. Rung 3:


Send Blocking SPDU Request - The IO-LInk master will not return until the SPDU command(s) have completed.
Enabling the PrtN_SendBlkgSpduReq control flag will instruct the PLC to send a Blocking SPDU Request that may contain one or more SPDU commands.
The MSG instruction can be changed to send a request with any valid sequence of SPDU commands.
It is very important that the SPDU response tag is of the same format as the SPDU request tag.

When the **PrtN_SendBlkISDUReq** tag is enabled, this rung sends an explicit message to the IO-Link Master. This message starts a blocking ISDU process where the IO-Link Master will not return a MSG response until all ISDU commands have been processed.

e. Rung 4-5:


Send Non-Blocking SPDU Request - The IO-LInk master will return immediately. The SPDU response must be requested until a valid pass or fail respon e is received.
Enabling the PrtN_SendNonBlkgSpduReq control flag will instruct the PLC to send a Non-Blocking SPDU Request that may contain one or more SPDU commands.
The MSG instruction can be changed to send an SPDU request with any valid sequence of SPDU commands.
It is very important that the SPDU response tag is of the same format as the SPDU request tag.

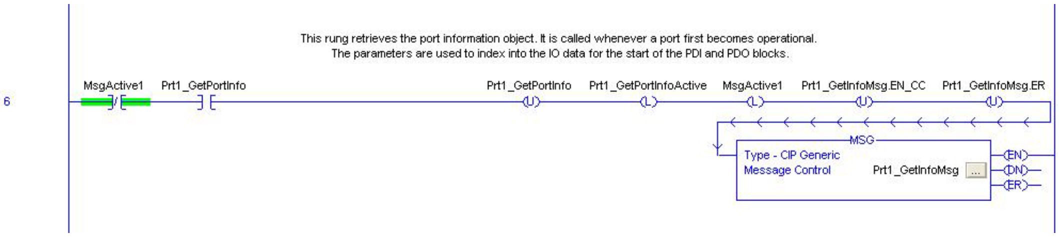
This rung is used to request a non-blocking response.
4
MsgActive1
Prt1_Operational
Prt1_SendNonBlkSpduReq
Prt1_SendNonBlkSpduReq
Prt1_NonBlkSpduReqActive
MsgActive1
Prt1_SendNonBlkSpduReqMsg.EN_CC
Prt1_SendNonBlkSpduReqMsg.ER
MSG
Type - CIP Generic
Message Control Prt1_SendNonBlkSpduReqMsg
EN
DN
ER
5
MsgActive1
Prt1_Operational
Prt1_GetNonBlkSpduResp
Prt1_GetNonBlkSpduResp
Prt1_GetNonBlkSpduRespActive
MsgActive1
Prt1_GetNonBlkSpduRespMsg.EN_CC
Prt1_GetNonBlkSpduRespMsg.ER
Prt1_GetNonBlkSpduRespMsg.DN
MSG
Type - CIP Generic
Message Control Prt1_GetNonBlkSpduRespMsg
EN
DN
ER

- When the **PrtN_SendNonBlkISDUReq** tag is enabled, this rung sends an explicit message to the IO-Link Master.
  - This message starts a blocking ISDU process where the IO-Link Master returns a MSG response immediately after verifying the ISDU request.
  - The IO-Link Master then processes all ISDU commands within the request.
  - The IO-Link returns In-Process statuses until all ISDU commands have been processed.
- When the **PrtN_GetNonBlkISDUResp** tag is enabled, this rung sends an explicit message to the IO-Link Master to retrieve the ISDU response.
- Run 7 enables (latches) **GetNonBlkISDUResp** as soon as the MSG in Rung 4 has completed

successfully.

- The ISDU response is retrieved until the response received indicates either a success (2) or error (3 or 4).

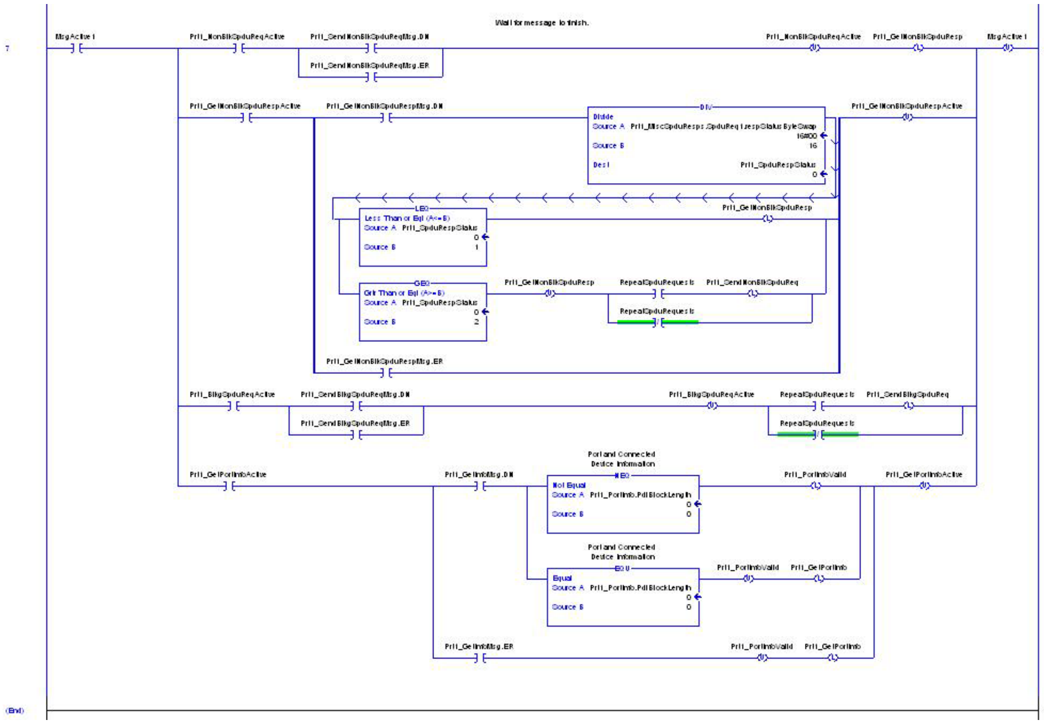
f. Rung 6:



This rung retrieves the port information object. It is called whenever a port first becomes operational.
The parameters are used to index into the IO data for the start of the PDI and PDO blocks.

- When the **PrtN_GetPortInfo** tag is enabled, this rung sends an explicit message to request the IO-Link port information block.
- The **PrtN_GetDevInfo** tag is enabled in Rung 0 whenever the IO-Link Master connection status transitions from inactive to active.

g. Rung 7:



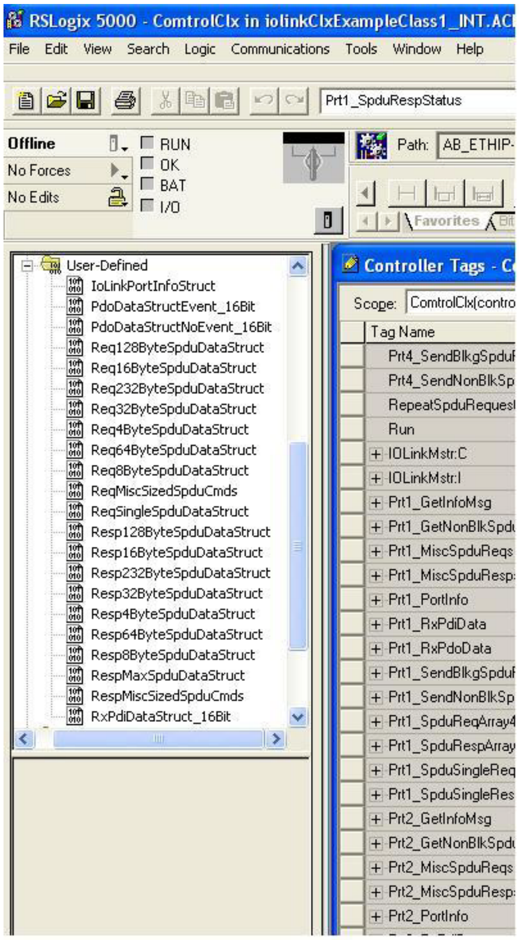
Wait for message to finish.

This rung monitors the various explicit messages for completion.

- Controls the non-blocking ISDU request process by enabling messages to retrieve the ISDU response until the request has completed.
- Sets the various flags when a get port information message has completed.

6.6. User Defined Data Structures

The example PLC program contains a number of User Defined Data Structures that may be used or modified as need be.

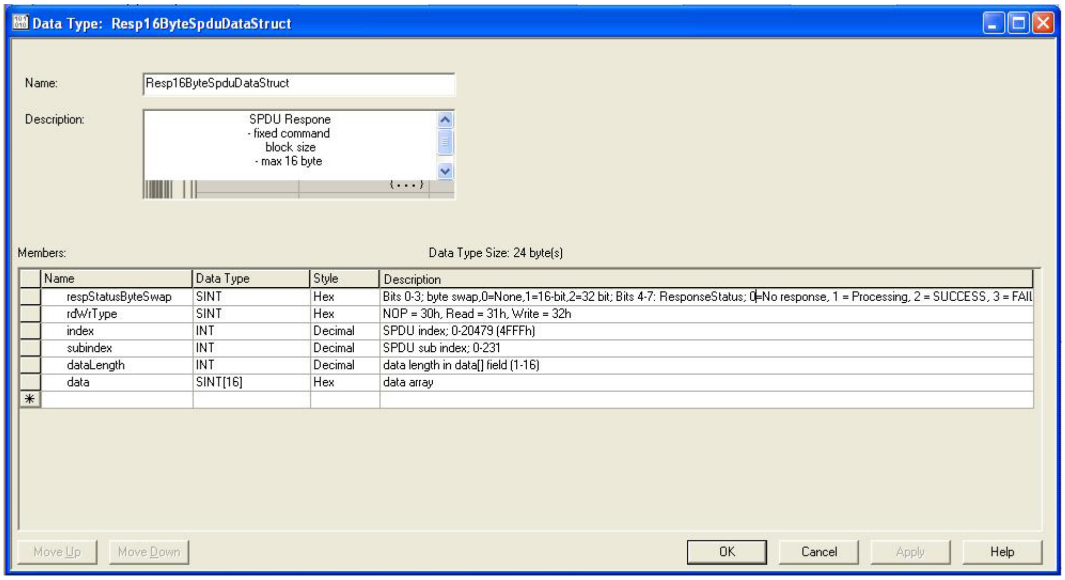


User-Defined
- IoLinkPortInfoStruct
- PdoDataStructEvent_16Bit
- PdoDataStructNoEvent_16Bit
- Req128ByteSpduDataStruct
- Req16ByteSpduDataStruct
- Req232ByteSpduDataStruct
- Req32ByteSpduDataStruct
- Req4ByteSpduDataStruct
- Req64ByteSpduDataStruct
- Req8ByteSpduDataStruct
- ReqMiscSizedSpduCmds
- ReqSingleSpduDataStruct
- Resp128ByteSpduDataStruct
- Resp16ByteSpduDataStruct
- Resp232ByteSpduDataStruct
- Resp32ByteSpduDataStruct
- Resp4ByteSpduDataStruct
- Resp64ByteSpduDataStruct
- Resp8ByteSpduDataStruct
- RespMaxSpduDataStruct
- RespMiscSizedSpduCmds
- RxPdiDataStruct_16Bit

The following illustrations show a few of the User Defined Data Structure formats.

6.6.1. User Defined Structure Example 1

This displays the first example of a User Defined Data Structure.



Name: Resp16ByteSpduDataStruct

Description: SPDU Respone
- fixed command
block size
- max 16 byte

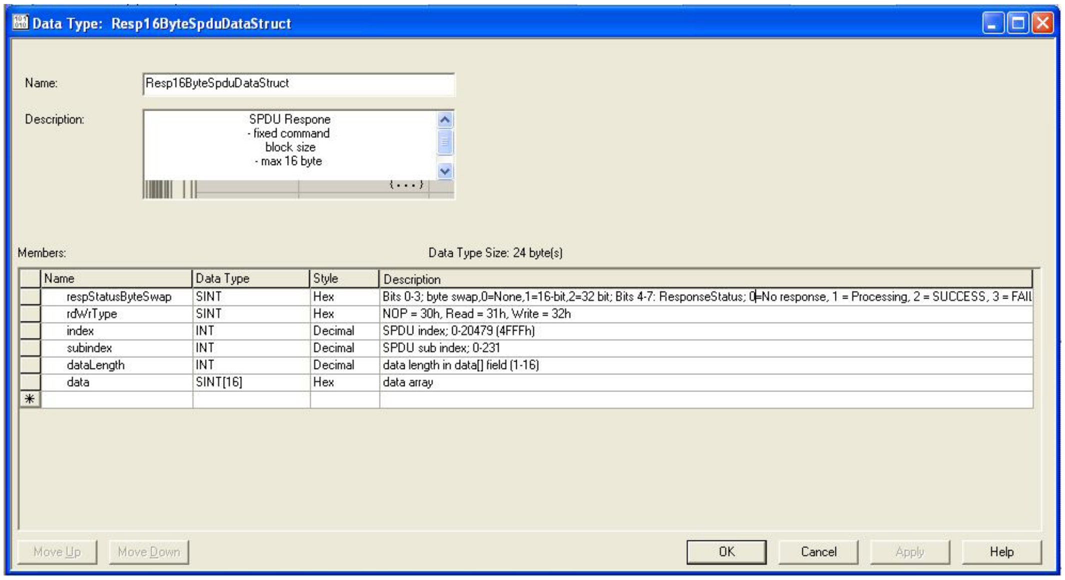
Members:

Data Type Size: 24 byte(s)

Name	Data Type	Style	Description
respStatusByteSwap	SINT	Hex	Bits 0-3; byte swap,0=None,1=16-bit,2=32 bit; Bits 4-7: ResponseStatus; 0=No response, 1 = Processing, 2 = SUCCESS, 3 = FAIL
rdWrType	SINT	Hex	NOP = 30h, Read = 31h, Write = 32h
index	INT	Decimal	SPDU index; 0-20479 (4FFFh)
subindex	INT	Decimal	SPDU sub index; 0-231
dataLength	INT	Decimal	data length in data[] field (1-16)
data	SINT[16]	Hex	data array
*			

6.6.2. User Defined Structure Example 2

This the second example of the User Defined Structure.



Name: Resp16ByteSpduDataStruct

Description: SPDU Respone
- fixed command
block size
- max 16 byte

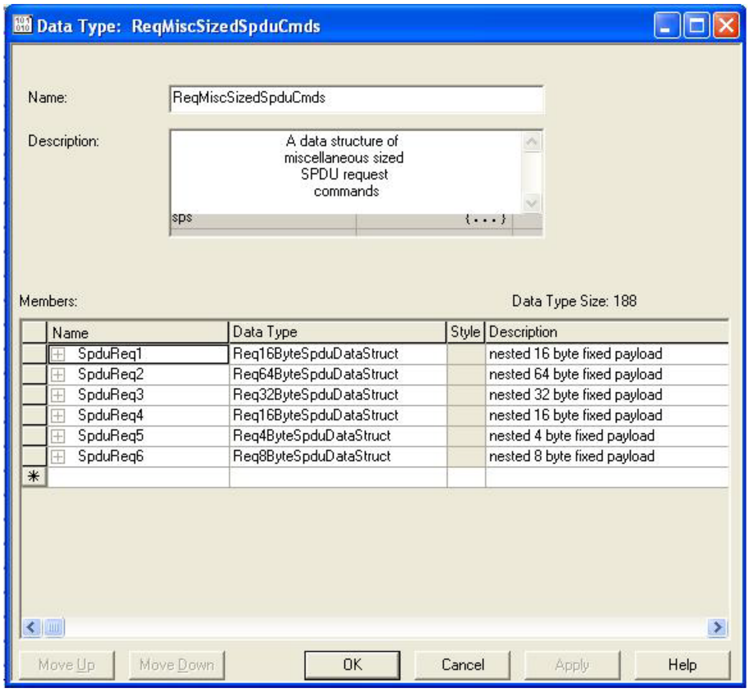
Members:

Data Type Size: 24 byte(s)

Name	Data Type	Style	Description
respStatusByteSwap	SINT	Hex	Bits 0-3; byte swap,0=None,1=16-bit,2=32 bit; Bits 4-7: ResponseStatus; 0=No response, 1 = Processing, 2 = SUCCESS, 3 = FAIL
rdWrType	SINT	Hex	NOP = 30h, Read = 31h, Write = 32h
index	INT	Decimal	SPDU index; 0-20479 (4FFFh)
subindex	INT	Decimal	SPDU sub index; 0-231
dataLength	INT	Decimal	data length in data[] field (1-16)
data	SINT[16]	Hex	data array
*			

6.6.3. User Defined Structure Example 3

This is the third example of a User Defined Structure.



Name: ReqMiscSizedSpduCmds

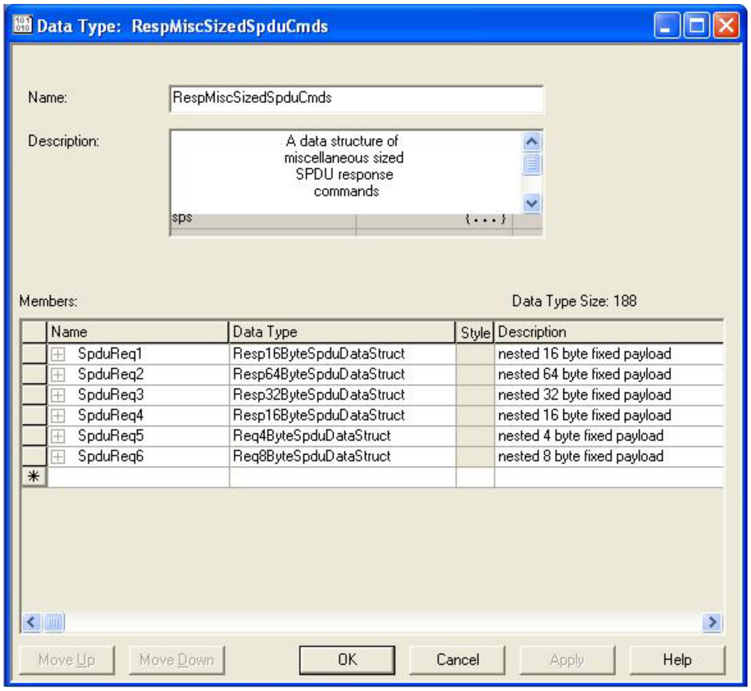
Description: A data structure of miscellaneous sized SPDU request commands

Members: Data Type Size: 188

Name	Data Type	Style	Description
SpduReq1	Req16ByteSpduDataStruct		nested 16 byte fixed payload
SpduReq2	Req64ByteSpduDataStruct		nested 64 byte fixed payload
SpduReq3	Req32ByteSpduDataStruct		nested 32 byte fixed payload
SpduReq4	Req16ByteSpduDataStruct		nested 16 byte fixed payload
SpduReq5	Req4ByteSpduDataStruct		nested 4 byte fixed payload
SpduReq6	Req8ByteSpduDataStruct		nested 8 byte fixed payload

6.6.4. User Defined Structure Example 4

This is the fourth example of a User Defined Structure.



Name: RespMiscSizedSpduCmds

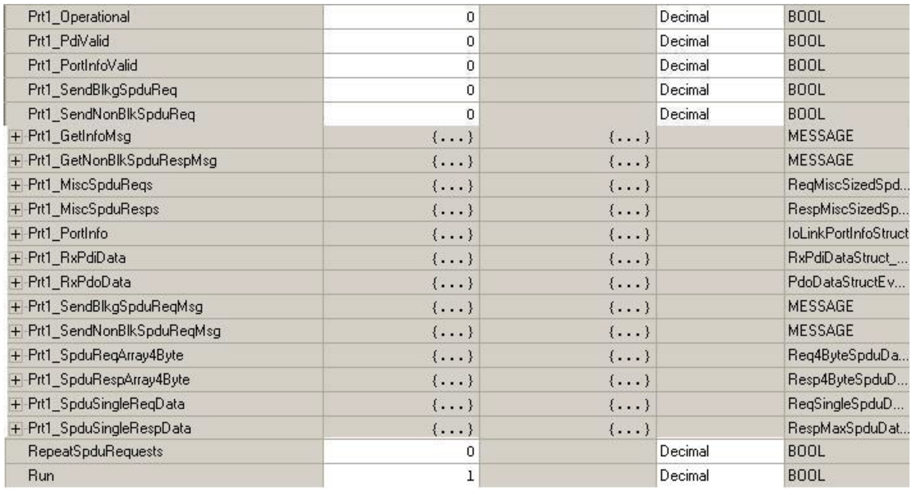
Description: A data structure of miscellaneous sized SPDU response commands

Members: Data Type Size: 188

Name	Data Type	Style	Description
SpduReq1	Resp16ByteSpduDataStruct		nested 16 byte fixed payload
SpduReq2	Resp64ByteSpduDataStruct		nested 64 byte fixed payload
SpduReq3	Resp32ByteSpduDataStruct		nested 32 byte fixed payload
SpduReq4	Resp16ByteSpduDataStruct		nested 16 byte fixed payload
SpduReq5	Req4ByteSpduDataStruct		nested 4 byte fixed payload
SpduReq6	Req8ByteSpduDataStruct		nested 8 byte fixed payload

6.7. Example PLC Program Tag Definitions

The following tag definitions apply to the example IO-Link Master PLC program.



Prt1_Operational	0		Decimal	BOOL
Prt1_PdiValid	0		Decimal	BOOL
Prt1_PortInfoValid	0		Decimal	BOOL
Prt1_SendBlkgSpduReq	0		Decimal	BOOL
Prt1_SendNonBlkSpduReq	0		Decimal	BOOL
Prt1_GetInfoMsg	{...}	{...}		MESSAGE
Prt1_GetNonBlkSpduRespMsg	{...}	{...}		MESSAGE
Prt1_MiscSpduReqs	{...}	{...}		ReqMiscSizedSpd...
Prt1_MiscSpduResps	{...}	{...}		RespMiscSizedSp...
Prt1_PortInfo	{...}	{...}		IoLinkPortInfoStruct
Prt1_RxPdiData	{...}	{...}		RxPdiDataStruct_...
Prt1_RxPdoData	{...}	{...}		PdoDataStructEv...
Prt1_SendBlkgSpduReqMsg	{...}	{...}		MESSAGE
Prt1_SendNonBlkSpduReqMsg	{...}	{...}		MESSAGE
Prt1_SpduReqArray4Byte	{...}	{...}		Req4ByteSpduDa...
Prt1_SpduRespArray4Byte	{...}	{...}		Resp4ByteSpduD...
Prt1_SpduSingleReqData	{...}	{...}		ReqSingleSpduD...
Prt1_SpduSingleRespData	{...}	{...}		RespMaxSpduDat...
RepeatSpduRequests	0		Decimal	BOOL
Run	1		Decimal	BOOL

Tag Name	Value Range	Description
PrtN_Operational (init state = false)	BOOL	Controlled by the subroutine, the port operational status. The port must be operational before communication to the IO-Link port is allowed. • 0 = false • 1= true
PrtN_PdiValid (init state = false)	BOOL	Controlled by the subroutine, the PDI (Input Process data block) valid status. • 0 = false • 1= true
PrtN_PortInfoValid (init state = false)	BOOL	Controlled by the subroutine, the port information valid status. The port information must be retrieved before the device can become operational. • 0 = false • 1= true
PrtN_SendBlkgISDUReq (init state = false)	BOOL	*Controlled by the User or some other part of a PLC program*, directs the subroutine whether to send a blocking ISDU request to the IO-Link Master. • 0 = false (do not send message) • 1= true (send message)
PrtN_SendNonBlkISDUReq (init state = false)	BOOL	Controlled by the User or some other part of a PLC program, directs the subroutine whether to begin the non-blocking ISDU request process. If true, the subroutine sends a non-blocking ISDU request to the IO-Link Master. • 0 = false (do not send message) • 1= true (send message)

Tag Name	Value Range	Description
PrtN_GetInfoMsg	MSG instruction parameters	Used by the subroutine, the message data used to get the port information from the IO-Link Master. ***Note:*** *This tag should not be modified by any other part of the PLC program or through the RSLogix 5000 user interface.*
PrtN_GetNonBlkISDURespMsg	MSG instruction parameters	Used by the subroutine, the message data used to get the non-blocking ISDU response from the IO-Link Master. ***Note:*** *This tag should not be modified by any other part of the PLC program or through the RSLogix 5000 user interface.*
PrtN_MiscISDUReqs	User defined data structure	Group of ISDU commands used as the default ISDU request format for the example PLC program. Can be modified by the user or other part of a PLC program. Refer to *6.7.3. PrtN_MiscISDUReqs* on Page 100 for more information.
PrtN_MiscISDUResps	User defined data structure	Group of ISDU command responses that is returned by the IO-Link Master after and ISDU request completion. Must be in same overall format as **PrtN_MiscISDUReqs**. Refer to *6.7.4. PrtN_MiscISDUResp* on Page 101 for a complete description.
PrtN_PortInfo	User defined data structure	Contains common device information parameters automatically read by the IO-Link Master during initialization of the IO-Link device interface.
PrtN_RxPdiData	User defined data structure	This tag contains the latest PDI data block as received from the Class 1 interface. It is updated with every **ProcessIoLinkPortN** subroutine call. Refer to *6.7.2. PrtN_RxPdiData Definition* on Page 99 for more information.
PrtN_SendBlkgISDUReqMsg	MSG instruction parameters	MSG instruction parameters used to send a blocking ISDU Request message. ***Note:*** *This tag should not be modified by any other part of the PLC program or through the RSLogix 5000 user interface.*
PrtN_SendNonBlkISDUReqMsg	MSG instruction parameters	MSG instruction parameters used to send a non-blocking ISDU Request message. ***Note:*** *This tag should not be modified by any other part of the PLC program or through the RSLogix 5000 user interface.*
PrtN_ISDUReqArray4Byte	ISDU command parameters	An alternative ISDU request format.
PrtN_ISDURespArray4Byte	ISDU response parameters	An alternative ISDU response format. Must be used with **PrtN_ISDUReqArray4Byte**.
PrtN_ISDUSingleReqData	ISDU command parameters	An alternative ISDU request format.

Tag Name	Value Range	Description
PrtN_ ISDUSingleRespData	ISDU response parameters	An alternative ISDU response format. Must be used with **PrtN_ISDUReqArray4Byte**.
RepeatISDURequests	BOOL	If enabled, instructs all subroutines to repeat any ISDU requests upon completion. Intended for testing purposes. May be enabled by end user.
Run	BOOL	**MainProgram** only. Allows the **ProcessIoLinkPortN** subroutine calls if enabled (1). Prevents the **ProcessIoLinkPortN** subroutine calls if disabled (0).

6.7.1. PrtN_DeviceInformation Definition

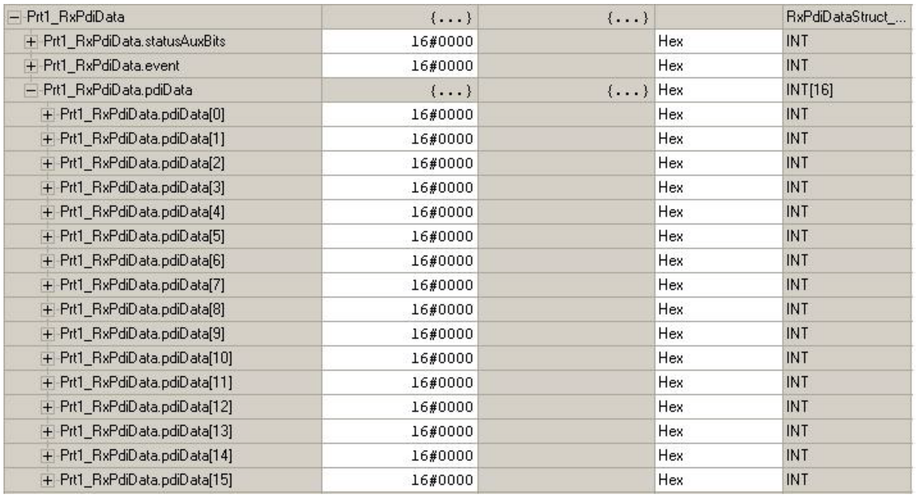
The IO-Link Master requests this information from the IO-Link device during the IO-Link device initialization process. It is then made accessible via explicit messages. The example PLC program automatically requests this information block when the device status transitions to active.

Parameter Name	Data	Description
VendorName	64 ASCII characters	Requested from ISDU data block index 16, contains the Vendor Name description of the IO-Link device.
VendorText	64 ASCII characters	Requested from ISDU data block index 17, contains the Vendor Text description of the IO-Link device.
ProductName	64 ASCII characters	Requested from ISDU data block index 18, contains the Product Name description of the IO-Link device.
ProductId	64 ASCII characters	Requested from ISDU data block index 19, contains the Product ID description of the IO-Link device.
ProductText	64 ASCII characters	Requested from ISDU data block index 20, contains the Product Text description of the IO-Link device.
SerialNum	16 ASCII characters	Requested from ISDU data block index 21, contains the Vendor Specific Serial Number of the IO-Link device.
HardwareRev	64 ASCII characters	Requested from ISDU data block index 22, contains the Hardware Revision of the IO-Link device.
FirmwareRev	64 ASCII characters	Requested from ISDU data block index 23, contains the Firmware Revision of the IO-Link device.
DevicePdiLength	INT	Length of valid PDI data from IO-Link device or port (if not in I/O Link mode).
DevicePdoLength	INT	Length of valid PDO data that can accepted by the IO-Link device or port (if not in I/O Link mode).
PdiBlockLength	INT	The configured PDI data block length. This includes the header bytes and any PDI data.
PdoBlockLength	INT	The configured PDO data block length. This includes the header bytes and any PDO data.
InputRxPdiOffset	INT	Provides the index into the Class 1 I/O input data received from the IO-Link Master. The index corresponds to the configured PDI data format of the port on the IO-Link Master. Used to automatically index into the input data and retrieve the PDI data block.

Parameter Name	Data	Description
InputRxPdoOffset	INT	Provides the index into the Class 1 I/O input data received from the IO-Link Master. The index corresponds to the configured PDO data format of the port on the IO-Link Master. Used to automatically index into the input data and retrieve the PDO data block.
OutputPdoOffset	INT	Provides the index into the Class 1 I/O output data sent to the IO-Link Master. The index corresponds to the configured PDO data format of the port on the IO-Link Master. Used to automatically index into the output data and transmit the PDO data block.
ControlFlags	Bit-mapped INT	Bit 0 (01h): 1 =Indicates that the event code to clear is expected in the PDO block 0 =Indicates that the event code to clear is not expected in the PDO block. The PDO data block only contains PDO data. Bit 1 (02h): 1 =Indicates that the IO-Link device is SIO mode capable 0 =Indicates that the IO-Link device is not SIO mode capable

6.7.2. PrtN_RxPdiData Definition

The PDI data block is received from the IO-Link Master over a Class 1 I/O connection. The data is then copied into the PDI data block in each subroutine.

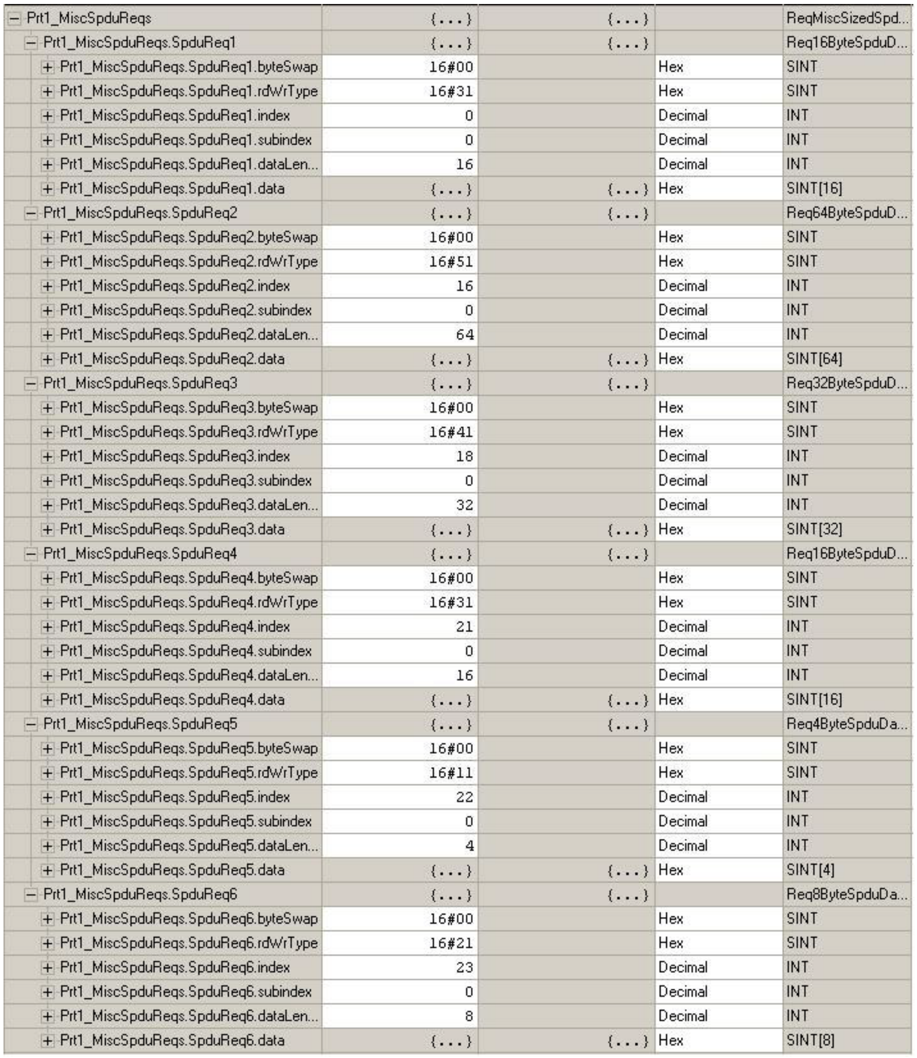


− Prt1_RxPdiData	{...}	{...}		RxPdiDataStruct_...
+ Prt1_RxPdiData.statusAuxBits	16#0000		Hex	INT
+ Prt1_RxPdiData.event	16#0000		Hex	INT
− Prt1_RxPdiData.pdiData	{...}	{...}	Hex	INT[16]
+ Prt1_RxPdiData.pdiData[0]	16#0000		Hex	INT
+ Prt1_RxPdiData.pdiData[1]	16#0000		Hex	INT
+ Prt1_RxPdiData.pdiData[2]	16#0000		Hex	INT
+ Prt1_RxPdiData.pdiData[3]	16#0000		Hex	INT
+ Prt1_RxPdiData.pdiData[4]	16#0000		Hex	INT
+ Prt1_RxPdiData.pdiData[5]	16#0000		Hex	INT
+ Prt1_RxPdiData.pdiData[6]	16#0000		Hex	INT
+ Prt1_RxPdiData.pdiData[7]	16#0000		Hex	INT
+ Prt1_RxPdiData.pdiData[8]	16#0000		Hex	INT
+ Prt1_RxPdiData.pdiData[9]	16#0000		Hex	INT
+ Prt1_RxPdiData.pdiData[10]	16#0000		Hex	INT
+ Prt1_RxPdiData.pdiData[11]	16#0000		Hex	INT
+ Prt1_RxPdiData.pdiData[12]	16#0000		Hex	INT
+ Prt1_RxPdiData.pdiData[13]	16#0000		Hex	INT
+ Prt1_RxPdiData.pdiData[14]	16#0000		Hex	INT
+ Prt1_RxPdiData.pdiData[15]	16#0000		Hex	INT

Refer to section on Input Process data format.

6.7.3. PrtN_MiscISDUReqs

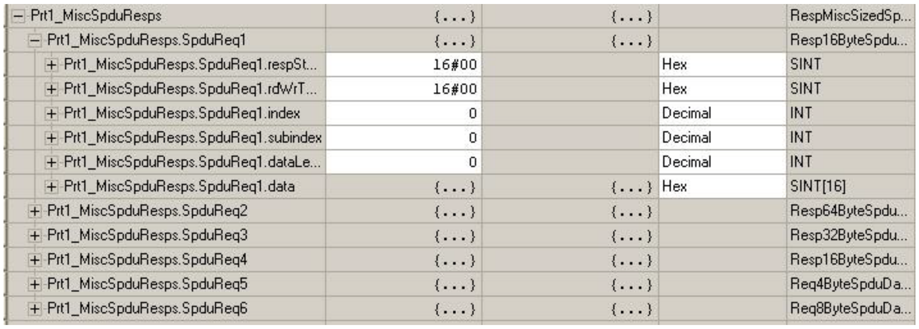
This tag is used as the default ISDU request. It contains several ISDU commands that are configured to read standard ISDU blocks supported by most IO-Link devices. This User Defined Structure may be changed to include any set of ISDU commands. The only constraint is that the entire Request and response must be no larger than the maximum MSG instruction payload of 500 bytes.



⊟ Prt1_MiscSpduReqs	{...}	{...}		ReqMiscSizedSpd...
⊟ Prt1_MiscSpduReqs.SpduReq1	{...}	{...}		Req16ByteSpduD...
⊞ Prt1_MiscSpduReqs.SpduReq1.byteSwap	16#00		Hex	SINT
⊞ Prt1_MiscSpduReqs.SpduReq1.rdWrType	16#31		Hex	SINT
⊞ Prt1_MiscSpduReqs.SpduReq1.index	0		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq1.subindex	0		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq1.dataLen...	16		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq1.data	{...}	{...}	Hex	SINT[16]
⊟ Prt1_MiscSpduReqs.SpduReq2	{...}	{...}		Req64ByteSpduD...
⊞ Prt1_MiscSpduReqs.SpduReq2.byteSwap	16#00		Hex	SINT
⊞ Prt1_MiscSpduReqs.SpduReq2.rdWrType	16#51		Hex	SINT
⊞ Prt1_MiscSpduReqs.SpduReq2.index	16		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq2.subindex	0		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq2.dataLen...	64		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq2.data	{...}	{...}	Hex	SINT[64]
⊟ Prt1_MiscSpduReqs.SpduReq3	{...}	{...}		Req32ByteSpduD...
⊞ Prt1_MiscSpduReqs.SpduReq3.byteSwap	16#00		Hex	SINT
⊞ Prt1_MiscSpduReqs.SpduReq3.rdWrType	16#41		Hex	SINT
⊞ Prt1_MiscSpduReqs.SpduReq3.index	18		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq3.subindex	0		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq3.dataLen...	32		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq3.data	{...}	{...}	Hex	SINT[32]
⊟ Prt1_MiscSpduReqs.SpduReq4	{...}	{...}		Req16ByteSpduD...
⊞ Prt1_MiscSpduReqs.SpduReq4.byteSwap	16#00		Hex	SINT
⊞ Prt1_MiscSpduReqs.SpduReq4.rdWrType	16#31		Hex	SINT
⊞ Prt1_MiscSpduReqs.SpduReq4.index	21		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq4.subindex	0		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq4.dataLen...	16		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq4.data	{...}	{...}	Hex	SINT[16]
⊟ Prt1_MiscSpduReqs.SpduReq5	{...}	{...}		Req4ByteSpduDa...
⊞ Prt1_MiscSpduReqs.SpduReq5.byteSwap	16#00		Hex	SINT
⊞ Prt1_MiscSpduReqs.SpduReq5.rdWrType	16#11		Hex	SINT
⊞ Prt1_MiscSpduReqs.SpduReq5.index	22		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq5.subindex	0		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq5.dataLen...	4		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq5.data	{...}	{...}	Hex	SINT[4]
⊟ Prt1_MiscSpduReqs.SpduReq6	{...}	{...}		Req8ByteSpduDa...
⊞ Prt1_MiscSpduReqs.SpduReq6.byteSwap	16#00		Hex	SINT
⊞ Prt1_MiscSpduReqs.SpduReq6.rdWrType	16#21		Hex	SINT
⊞ Prt1_MiscSpduReqs.SpduReq6.index	23		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq6.subindex	0		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq6.dataLen...	8		Decimal	INT
⊞ Prt1_MiscSpduReqs.SpduReq6.data	{...}	{...}	Hex	SINT[8]

6.7.4. PrtN_MiscISDUResp

This tag contains the response to the ISDU request. It must be of the same size and structure as the request **structure**.

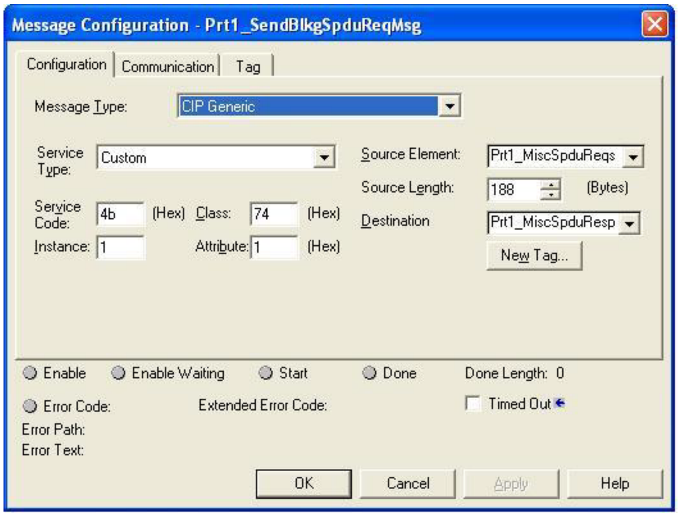


− Prt1_MiscSpduResps	{...}	{...}		RespMiscSizedSp...
− Prt1_MiscSpduResps.SpduReq1	{...}	{...}		Resp16ByteSpdu...
+ Prt1_MiscSpduResps.SpduReq1.respSt...	16#00		Hex	SINT
+ Prt1_MiscSpduResps.SpduReq1.rdWrT...	16#00		Hex	SINT
+ Prt1_MiscSpduResps.SpduReq1.index	0		Decimal	INT
+ Prt1_MiscSpduResps.SpduReq1.subindex	0		Decimal	INT
+ Prt1_MiscSpduResps.SpduReq1.dataLe...	0		Decimal	INT
+ Prt1_MiscSpduResps.SpduReq1.data	{...}	{...}	Hex	SINT[16]
+ Prt1_MiscSpduResps.SpduReq2	{...}	{...}		Resp64ByteSpdu...
+ Prt1_MiscSpduResps.SpduReq3	{...}	{...}		Resp32ByteSpdu...
+ Prt1_MiscSpduResps.SpduReq4	{...}	{...}		Resp16ByteSpdu...
+ Prt1_MiscSpduResps.SpduReq5	{...}	{...}		Req4ByteSpduDa...
+ Prt1_MiscSpduResps.SpduReq6	{...}	{...}		Req8ByteSpduDa...

6.7.5. Using Other ISDU Request/Response Command Formats

Other ISDU request/response formats may be used instead of the default request command set. The following steps demonstrate how to change the ISDU request/response formats:

1. If one ISDU request/response is required, create a new request and response tag with any of the defined ISDU User Defined structures. The one requirement is that the request and response formats must be the same. For example, if a 16 byte nested format is use for the request, then a 16 byte nested response structure must be used.
2. If multiple ISDU requests of the same nested lengths are required, created request and response arrays of the same User Defined format.
3. If multiple ISDU requests of different nested lengths are required, create new User Defined Data Structures for the request and response containing user defined command structures. Then create tags using the new user Defined data structures. You may also want to modify the **ReqMiscSizedISDUCmds** and **RespMiscSizedISDUCmds** User Defined data structures.
4. Modify the appropriate **MSG** instruction settings:



Message Configuration - Prt1_SendBlkgSpduReqMsg

Message Type: CIP Generic

Service Type: Custom

Service Code: 4b (Hex) Class: 74 (Hex)

Instance: 1 Attribute: 1 (Hex)

Source Element: Prt1_MiscSpduReqs

Source Length: 188 (Bytes)

Destination: Prt1_MiscSpduResp

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

○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

   a. Change the **Source Element** to that of the new ISDU request tag.
   b. Change the **Source Length** to that of the new **Source Element**. That information is often displayed on the User Defined Structure definition pane.
   c. Change the **Destination** to that of the new response tag.

Chapter 7. SLC/PLC-5/MicroLogix Interface

The IO-Link Master provides support for the SLC, PLC-5 and MicroLogix PLCs. The following features are supported:

- Rx PDI data, both Polling and Write-to-File modes.
- Tx PDO data, both PLC-Writes and Read-From-File modes.
- PCCC based messages transferred by means of the PCCC CIP 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)
- Receive, transmit and statistics data.
- Standard PLC-5/SLC file naming conventions.
- 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 Read-from-File communication methods in place of Write-to-Tag and Read-from-Tag communication methods. The Write-to-File methods operate in a very similar manner to the Write-to-Tag method 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 IO-Link Master to operate in Write-to-File or Read-from-File, enter the file name starting with an **N** (i.e. N10:0).

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.*

7.1. Requirements

Your PLC-5/SLC/MicroLogix PLC 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 IO-Link Master.*

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

7.2. PLC-5 and SLC 5/05 PLC Requirements

The following PLCs support Ethernet/IP.

7.2.1. 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.

7.2.2. 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 and later • Series D: Revision E and later • Series E: Revision D and later *Full Ethernet/IP Compliance:* • Series C: Revision R and later • Series D: Revision H and later • Series E: Revision G and 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-L20C15 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 and 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
- Ethernet Interface Module Series B, Revision D Product Release Notes, Rockwell Publication 1785-RN191E-EN-P

Note: *Older versions of firmware may or may not provide Ethernet/IP functionality.*

7.3. PLC-5 and SLC Messages

The following PCCC messages are supported for the PLC-5 and SLC 5/05 PLCs.

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)

The Receive Port Information is provided in one continuous file. The following file addresses are used to retrieve the various parameters.

	IO-Link Port 1	IO-Link Port 2	IO-Link Port 3	IO-Link Port 4	Access	Length
PDI Data Block	N10:0	N20:0	N30:0	N40:0	Read-Only	Configurable per port ***Note:*** *See the table below for details.*
Receive PDO Data Block	N11:0	N21:0	N31:0	N41:0	Read-Only	Configurable per port ***Note:*** *See the table below for details.*
Transmit PDO Data Block	N12:0	N22:0	N32:0	N42:0	Write-Only	Configurable per port ***Note:*** *See the table below for details.*
Receive ISDU Response	N13:0	N23:0	N33:0	N43:0	Read-Only	4 INTs to Max Msg Size
Transmit ISDU Request	N14:0	N24:0	N34:0	N44:0	Write-Only	4 INTs to Max Msg Size
Port Information Block (Continuous Block)						464 Bytes (232 INTs)

	IO-Link Port 1	IO-Link Port 2	IO-Link Port 3	IO-Link Port 4	Access	Length
Vendor Name	N15:0	N25:0	N35:0	N45:0	Read	64 Chars (32 INTs)
Vendor Text	N15:32	N25:32	N35:32	N45:32	Read	64 Chars (32 INTs)
Product Name	N15:64	N25:64	N35:64	N45:64	Read	64 Chars (32 INTs)
Product ID	N15:96	N25:96	N35:96	N45:96	Read	64 Chars (32 INTs)
Product Text	N15:128	N25:128	N35:128	N45:128	Read	64 Chars (32 INTs)
Serial Number	N15:160	N25:160	N35:160	N45:160	Read	16 Chars (8 INTs)
Hardware Revision	N15:168	N25:168	N35:168	N45:168	Read	64 Chars (32 INTs)
Firmware Revision	N15:200	N25:200	N35:200	N45:200	Read	64 Chars (32 INTs)

7.4. Process Data (PDI and PDO) Access via PCCC Messages

The process data has been grouped together in order to minimize the number of PCCC messages required to interface to the IO-Link Master. The PDI and PDO data for multiple ports can be received or transmitted by one message.

	File Number	Controller Port 1 Access		Controller Port 2 Access		Controller Port 3 Access		Controller Port 4 Access	
		Read (Input)	**Write (Output)**	**Read (Input)**	**Write (Output)**	**Read (Input)**	**Write (Output)**	**Read (Input)**	**Write (Output)**
Read (Input) Process Data Input	N10:0 (Port 1)	▨							
	N20:0 (Port 2)	▨		▨					
	N30:0 (Port 3)	▨		▨		▨			
	N40:0 (Port 4)	▨		▨		▨		▨	
Read (Input) Process Data Output	N11:0 (Port 1)	▨							
	N21:0 (Port 2)	▨		▨					
	N31:0 (Port 3)	▨		▨		▨			
	N41:0 (Port 4)	▨		▨		▨		▨	
Write (Output) Process Data Output	N12:0 (Port 1)		▨						
	N22:0 (Port 2)		▨		▨				
	N32:0 (Port 3)		▨		▨		▨		
	N33:0 (Port 4)		▨		▨		▨		▨

PCCC Read/Write Access *where*:

- All PDI data can be read with one PCCC read message.
- All PDO data can be read with one PCCC read message.
- All PDO data can be written with one PCCC write message.
- Controller Read access:
  - The PDI data from one or more ports may be read with one message. (That is, if addressing Port 1, N10:0, ports one to four may be read in one message.)

  - The PDO data from one or more ports may be read with one message. (That is, if addressing Port 1, N11:0, ports one to four may be read in one message.)
  - Partial PDI and PDO data reads are allowed.
  - The length of the Read message can range from 1 to the total, configured PDI or PDO length for all ports starting at the addressed port.
- Controller Write (Output) access:
  - Only PDO data may be written.
  - The PDO data for one or more ports may be written with one message.
  - Partial PDO data writes are not allowed.
  - The length of the Write message must be equal to the total of the configured PDO lengths for all ports to be written. The one exception is that the data length of the last port to be written must be equal to or greater than the device PDO length for that port.

Chapter 8. Troubleshooting and Technical Support

This section provides the following information:

- *8.1. Troubleshooting*
- *8.2. Contacting Technical Support* on Page 110
- *8.3. Using Log Files* on Page 111

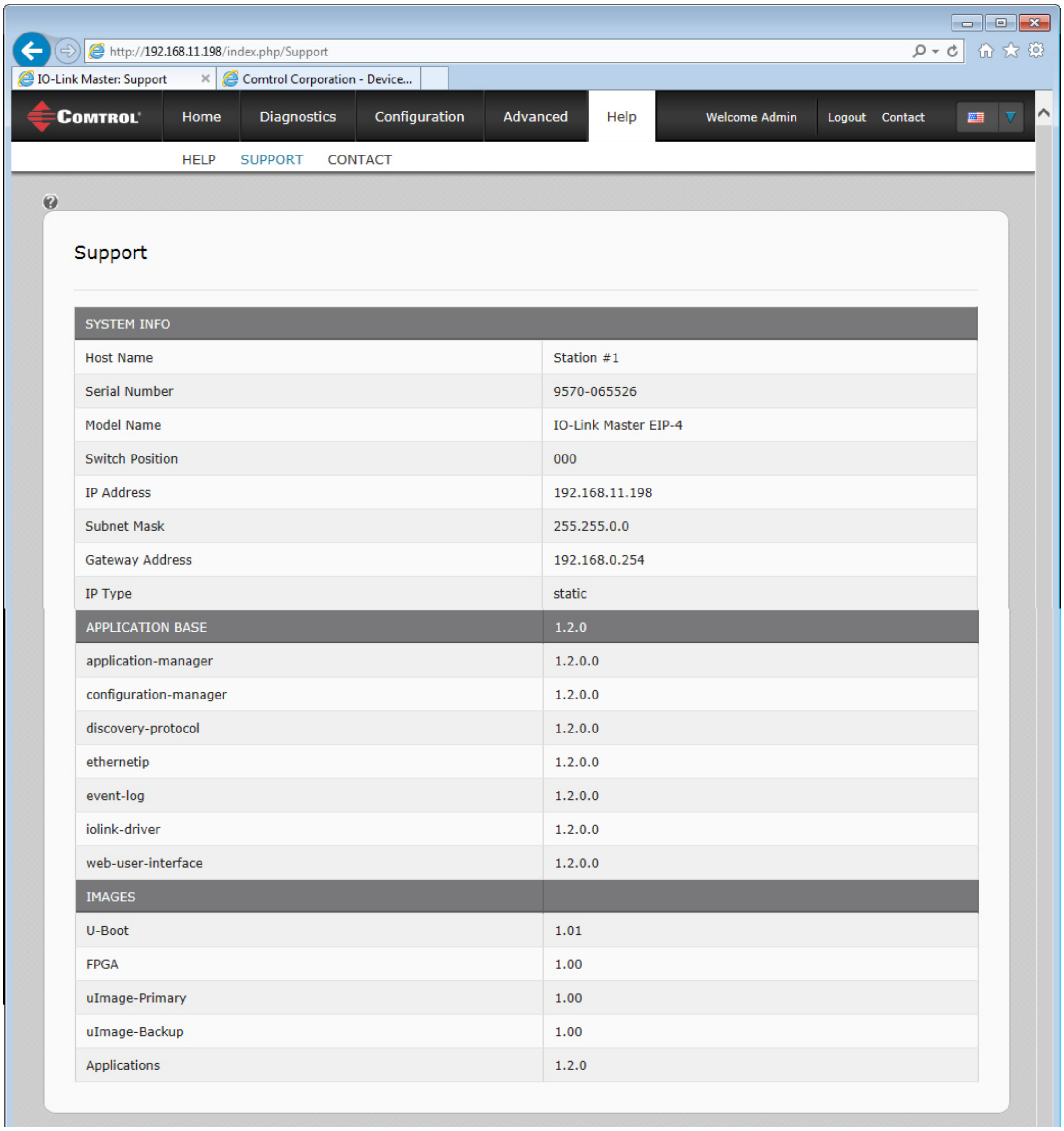
8.1. Troubleshooting

Before contacting Technical Support, you may want to try the following:

- Open the IO-Link Master web interface and review the following web pages:
  - IO-Link Diagnostics
  - EtherNet/IP Diagnostics
  - Modbus/TCP Diagnostics
- Reboot the IO-Link Master
- Verify that you are using the correct types of cables on the correct connectors and that all cables are connected securely.
- Check to make sure LEDs are not reporting an issue using the *IO-Link Master EIP4 User Guide*
- Verify that the network IP address, subnet mask, and gateway are correct and appropriate for the network. Make sure that the IP address programmed into the IO-Link Master matches the unique reserved IP configured address assigned by the system administrator.
- If using DHCP, the host system needs to provide the subnet mask. The gateway is optional and is not required for a purely local network.
- Remember that if the rotary switches are set to a non-default position, the rotary switches override the lower 3 digits (8 bits) of the static IP address configured in the **Network** page or in PortVision DX.
- Verify that the Ethernet hub and any other network devices between the system and the IO-Link Master are powered up and operating.
- If you have a spare IO-Link Master, try replacing the IO-Link Master.

8.2. Contacting Technical Support

You may want to access the **Help/SUPPORT** page when you call Technical Support, as they may request the information displayed on the **SUPPORT** page.



Support

SYSTEM INFO	
Host Name	Station #1
Serial Number	9570-065526
Model Name	IO-Link Master EIP-4
Switch Position	000
IP Address	192.168.11.198
Subnet Mask	255.255.0.0
Gateway Address	192.168.0.254
IP Type	static
APPLICATION BASE	**1.2.0**
application-manager	1.2.0.0
configuration-manager	1.2.0.0
discovery-protocol	1.2.0.0
ethernetip	1.2.0.0
event-log	1.2.0.0
iolink-driver	1.2.0.0
web-user-interface	1.2.0.0
IMAGES	
U-Boot	1.01
FPGA	1.00
uImage-Primary	1.00
uImage-Backup	1.00
Applications	1.2.0

Comtrol Technical Support is available from 8:00AM to 6:00PM (CST), Monday through Friday, excluding major USA holidays.

Contact	Information
Phone	763.957.6000
Downloads	ftp://ftp.comtrol.com/html/default.htm
Web Site	http://www.comtrol.com

8.3. Using Log Files

Log files are available in the IO-Link Master web page. The IO-Link Master provides four different log files that you can view, export, or clear:

- **Syslog** (system log) displays line-by-line activity records.
- **dmesg** displays Linux kernel messages.
- **top** displays which programs are using most of the memory and CPU.
- **ps** displays the running programs
- All log files start up automatically during the startup cycle. Each log file has a size limit of 100KB.

Note: *Typically, log files are intended to be used by Technical Support in the event there is a problem.*

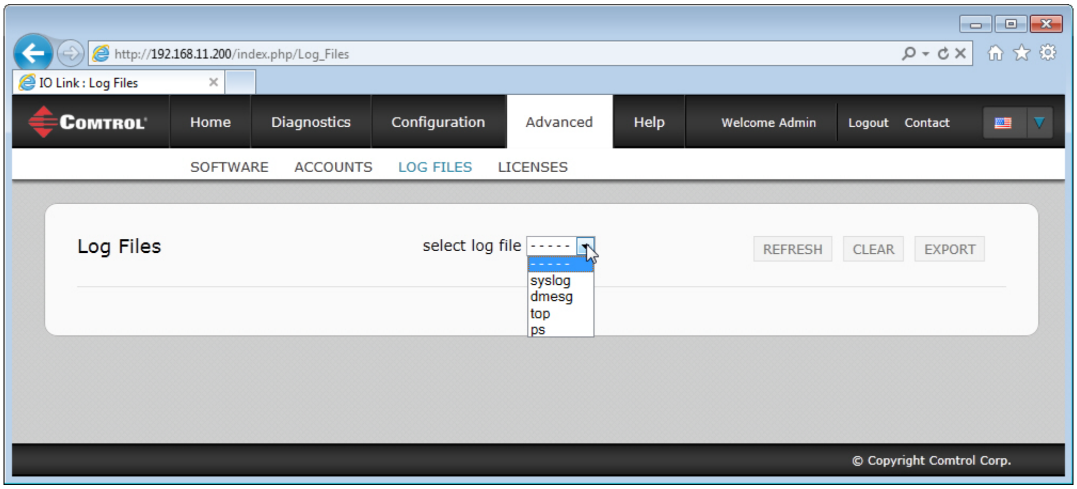
You can use the following procedures to:

- *8.3.1. View a Log File* on Page 111
- *8.3.2. Clear a Log File* on Page 111
- *8.3.3. Export a Log File* on Page 112

8.3.1. View a Log File

Use this procedure to view a log file.

1. Open the IO-Link Master web interface using one of these method:
   - From PortVision DX, highlight the IO-Link Master and click the **Webpage** button or right-click the IO-Link Master in the *Device List* pane and click **Webpage**.
   - Open your browser and enter the IP address of the IO-Link Master.
2. Click **Advanced** and then **LOG FILES**.
3. Select the log file type from the drop-list.



4. Optionally, click the **REFRESH** button to get the latest information.
5. Optionally, export the log file.

8.3.2. Clear a Log File

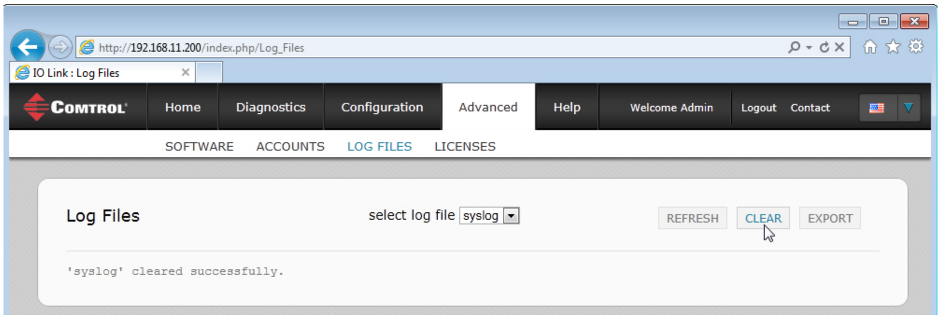
Use this procedure to clear a log file.

1. Open the IO-Link Master web interface using one of these method:
   - From PortVision DX, highlight the IO-Link Master and click the **Webpage** button or right-click the IO-

Link Master in the *Device List* pane and click **Webpage**.

- Open your browser and enter the IP address of the IO-Link Master.

2. Click **Advanced** and then **LOG FILES**.
3. Optionally, export the log file.
4. Select the log file type from the drop-list.
5. Click the **CLEAR** button.



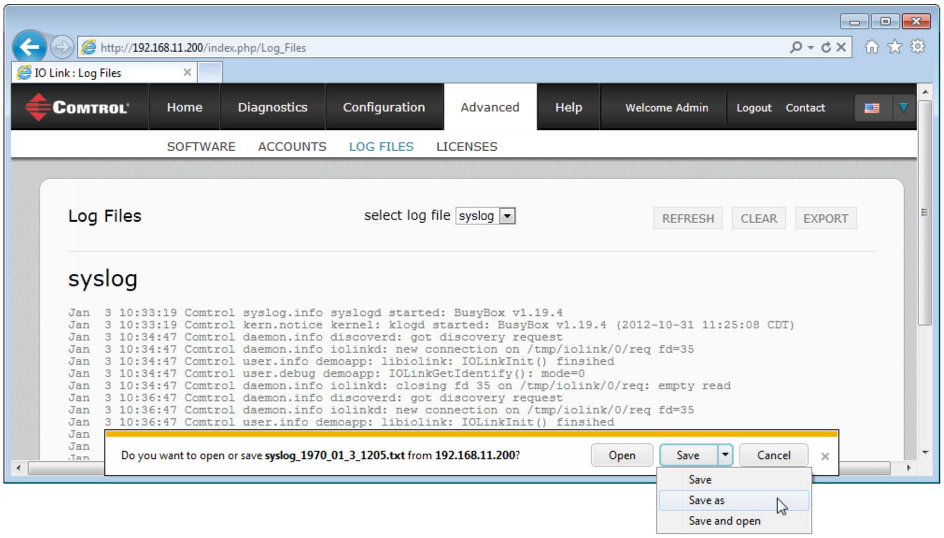
```
'syslog' cleared successfully.
```

The log file automatically starts logging the latest information.

8.3.3. Export a Log File

Use the following procedure to export a log file.

1. Open the IO-Link Master web interface using one of these method:
   - From PortVision DX, highlight the IO-Link Master and click the **Webpage** button or right-click the IO-Link Master in the *Device List* pane and click **Webpage**.
   - Open your browser and enter the IP address of the IO-Link Master.
2. Click **Advanced** and then **LOG FILES**.
3. Select the log file type from the drop-list.
4. Click the **EXPORT** button.
5. Click the **Save** button drop-list and click **Save** to save it to your user folder or **Save as** to browse to or create a new folder in which to place the log file.



```
Jan  3 10:33:19 Comtrol syslog.info syslogd started: BusyBox v1.19.4
Jan  3 10:33:19 Comtrol kern.notice kernel: klogd started: BusyBox v1.19.4 (2012-10-31 11:25:08 CDT)
Jan  3 10:34:47 Comtrol daemon.info discoverd: got discovery request
Jan  3 10:34:47 Comtrol daemon.info iolinkd: new connection on /tmp/iolink/0/req fd=35
Jan  3 10:34:47 Comtrol user.info demoapp: libiolink: IOLinkInit() finsihed
Jan  3 10:34:47 Comtrol user.debug demoapp: IOLinkGetIdentify(): mode=0
Jan  3 10:34:47 Comtrol daemon.info iolinkd: closing fd 35 on /tmp/iolink/0/req: empty read
Jan  3 10:36:47 Comtrol daemon.info discoverd: got discovery request
Jan  3 10:36:47 Comtrol daemon.info iolinkd: new connection on /tmp/iolink/0/req fd=35
Jan  3 10:36:47 Comtrol user.info demoapp: libiolink: IOLinkInit() finsihed
Jan
Jan
Jan
```

Do you want to open or save **syslog_1970_01_3_1205.txt** from **192.168.11.200**?

6. Depending on your operating system, you may need to close the pop-up window.