Quantum EIO

Distributed I/O Network Installation and Configuration Guide

Original instructions

10/2019

S1A48986.12

www.schneider-electric.com

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

Table of Contents

Safety Information

Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

DANGER
DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

WARNING
WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

CAUTION
CAUTION indicates a hazardous situation which, if not avoided, **could result** in minor or moderate injury.

NOTICE
NOTICE is used to address practices not related to physical injury.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

BEFORE YOU BEGIN

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

⚠ WARNING

UNGUARDED EQUIPMENT

- Do not use this software and related automation equipment on equipment which does not have point-of-operation protection.
- Do not reach into machinery during operation.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, you should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the Function Block Library, System User Guide, or other implementation referenced in this documentation.

START-UP AND TEST

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check be made and that enough time is allowed to perform complete and satisfactory testing.

⚠ WARNING

EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and temporary grounds that are not installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:
- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove all temporary grounds from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

OPERATION AND ADJUSTMENTS

The following precautions are from the NEMA Standards Publication ICS 7.1-1995 (English version prevails):

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.
- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer’s instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer’s instructions and the machinery used with the electrical equipment.
- Only those operational adjustments actually required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

About the Book

At a Glance

Document Scope

PlantStruxure is a Schneider Electric program designed to address the key challenges of many different types of users, including plant managers, operations managers, engineers, maintenance teams, and operators, by delivering a system that is scalable, flexible, integrated, and collaborative.

This document presents one of the PlantStruxure features, using Ethernet as the backbone around the Quantum PLC offer and connecting a *Quantum local rack* to Quantum and Modicon X80 *remote I/O drops* and distributed I/O devices. This feature is known as Quantum Ethernet I/O or Quantum EIO. (*NOTE: Modicon X80 is the generic name given to the M340 I/O modules when they are connected remotely to a Quantum controller or module in a PlantStruxure architecture. The M340 I/O name is still used when the module is connected to a M340 controller. The product references remains unchanged; only the range name changes.*)

This guide describes the 140NOC78000 distributed I/O head module and its role in a Quantum EIO system.

NOTE: The specific configuration settings contained in this guide are for instructional purposes only. The settings required for your specific application may differ from the examples presented in this guide.

Validity Note

This document is valid for the Quantum EIO system when used with EcoStruxure™ Control Expert 14.1 or later.

The technical characteristics of the devices described in the present document also appear online. To access the information online:

Step	Action
1	Go to the Schneider Electric home page *www.schneider-electric.com*.
2	In the **Search** box type the reference of a product or the name of a product range. ● Do not include blank spaces in the reference or product range. ● To get information on grouping similar modules, use asterisks (*).
3	If you entered a reference, go to the **Product Datasheets** search results and click on the reference that interests you. If you entered the name of a product range, go to the **Product Ranges** search results and click on the product range that interests you.
4	If more than one reference appears in the **Products** search results, click on the reference that interests you.
5	Depending on the size of your screen, you may need to scroll down to see the datasheet.
6	To save or print a datasheet as a .pdf file, click **Download XXX product datasheet**.

The characteristics that are presented in the present document should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the document and online information, use the online information as your reference.

Related Documents

Title of Documentation	Reference Number
Quantum EIO Global System Planning Guide	S1A48959 (English), S1A48961 (French), S1A48962 (German), S1A48964 (Italian), S1A48965 (Spanish), S1A48966 (Chinese)
Quantum EIO Remote I/O Modules Installation and Configuration Guide	S1A48978 (English), S1A48981 (French), S1A48982 (German), S1A48983 (Italian), S1A48984 (Spanish), S1A48985 (Chinese)
Quantum EIO Control Network Installation and Configuration Guide	S1A48993 (English), S1A48994 (French), S1A48995 (German), S1A48997 (Italian), S1A48998 (Spanish), S1A48999 (Chinese)
Modicon Quantum Change Configuration on the Fly User Guide	S1A48967 (English), S1A48968 (French), S1A48969 (German), S1A48970 (Italian), S1A48972 (Spanish), S1A48976 (Chinese)
Modicon Quantum Hot Standby System User Manual	35010533 (English), 35010534 (French), 35010535 (German), 35010536 (Spanish), 35013993 (Italian), 35012188 (Chinese)
Modicon M340/X80 BMX NRP 020• Fiber Optic Repeater Module User Guide	EIO0000001108 (English), EIO0000001109 (French), EIO0000001110 (German), EIO0000001111 (Spanish), EIO0000001112 (Italian), EIO0000001113 (Chinese)

Title of Documentation	Reference Number
Modicon M340/X80 with Control Expert Analog Input/Output Modules User Manual	35011978 (English), 35011979 (German), 35011980 (French), 35011981 (Spanish), 35011982 (Italian), 35011983 (Chinese)
Modicon M340/X80 with Control Expert Discrete Input/Output Modules User Manual	35012474 (English), 35012475 (German), 35012476 (French), 35012477 (Spanish), 35012478 (Italian), 35012479 (Chinese)
Modicon M340/X80 with Control Expert BMX EHC 0200 Counting Module User Manual	35013355 (English), 35013356 (German), 35013357 (French), 35013358 (Spanish), 35013359 (Italian), 35013360 (Chinese)
Control Expert Program Languages and Structure Reference Manual	35006144 (English), 35006145 (French), 35006146 (German), 35006147 (Spanish), 35013361 (Italian), 35013362 (Chinese)
EcoStruxure™ Control Expert, System Bits and Words, Reference Manual	EIO0000002135 (English), EIO0000002136 (French), EIO0000002137 (German), EIO0000002138 (Italian), EIO0000002139 (Spanish), EIO0000002140 (Chinese)
Control Expert Operating Modes	33003101 (English), 33003102 (French), 33003103 (German), 33003104 (Spanish), 33003696 (Italian), 33003697 (Chinese)
Quantum with Control Expert Hardware Reference Manual	35010529 (English), 35010530 (French), 35010531 (German), 35010532 (Spanish), 35013975 (Italian), 35012184 (Chinese)

Title of Documentation	Reference Number
Control Expert Installation Manual	35014792 (French), 35014793 (English), 35014794 (German), 35014795 (Spanish), 35014796 (Italian), 35012191 (Chinese)

You can download these technical publications and other technical information from our website at https://www.schneider-electric.com/en/download

Chapter 1
Characteristics of the 140NOC78000 Module

Introduction

This chapter describes the 140NOC78000 head module for distributed I/O communications in a Quantum EIO system.

This chapter includes physical characteristics, port descriptions, and agency specifications for the 140NOC78000 module.

What Is in This Chapter?

This chapter contains the following topics:

140NOC78000 Module Description

Introduction

140NOC78000 distributed I/O head modules are installed on the local rack of a Quantum EIO system. They provide the interfaces to communicate with distributed I/O devices and client applications on an Ethernet remote I/O network.

Functionality

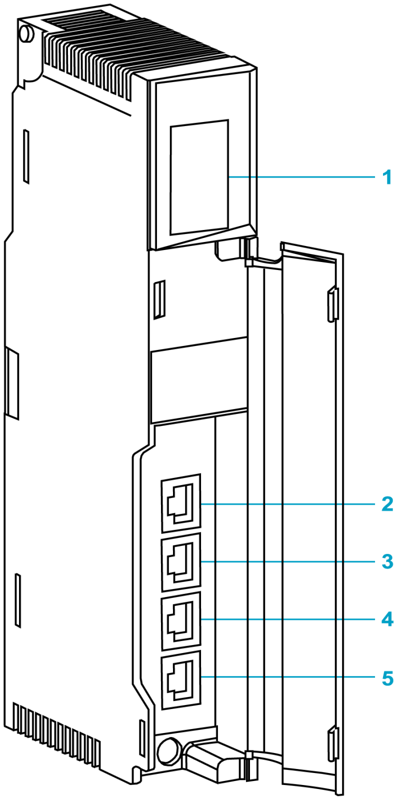
The main purpose of the 140NOC78000 module is to provide I/O scanning services to distributed I/O devices on the device network and distributed I/O network. In addition, the 140NOC78000 module (when interlinked with the 140NOC78100 control head module) provides services to communicate with PLC applications running on the control network.

You can configure a maximum of six 140NOC78000 modules on the local rack. If you configure a 140NOC78100 module on the local rack to manage a control network, you can only configure five 140NOC78000 modules on the local rack.

To communicate with...	Interlink the 140NOC78000 module with...
distributed I/O devices on a device network	the 140CRP31200 module **NOTE:** You may interlink the 140NOC78000 module with up to 2 additional 140NOC78000 modules on the local rack.
distributed I/O devices on an isolated distributed I/O network	other 140NOC78000 modules only on the local rack
distributed I/O devices on an independent distributed I/O network	the 140NOC78100 module **NOTE:** Neither the 140NOC78000 nor the 140NOC78100 modules can be interlinked with the 140CRP31200 module on the local rack. **NOTE:** You may interlink the 140NOC78000 module with up to 2 additional 140NOC78000 modules on the local rack.
distributed I/O devices on an extended distributed I/O network	• the 140CRP31200 module – and – • the 140NOC78100 module (when the IP forwarding service is configured) **NOTE:** You may interlink the 140NOC78000 module with up to 2 additional 140NOC78000 modules on the local rack.
the control network	the 140NOC78100 module

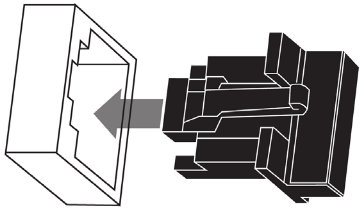
External Features

140NOC78000:



1 LED display
2 SERVICE port (ETH 1)
3 INTERLINK port (ETH 2)
4 DEVICE NETWORK port (ETH 3)
5 DEVICE NETWORK port (ETH 4)

NOTE: To help prevent dust from entering the unused Ethernet ports on this module, cover the port with the stopper:



External Ports

The 140NOC78000 module monitors the functionality of network links depending on which links are connected to the network. The module has 4 external ports (but only 1 IP address).

Port	Quantity	Description
SERVICE	1	The SERVICE port allows the diagnosis of Ethernet ports and provides access to external tools and devices (Control Expert, ConneXium Network Manager, HMI, etc.). The port supports these modes: ● access port (default): This mode supports Ethernet communications. ● port mirroring: In this mode, data traffic from one or more of the other 3 ports is copied to this port. This allows a connected tool to monitor and analyze the port traffic. ● disabled **NOTE:** ● If the device, which is connected to the service port, is configured for a speed that exceeds 100 Mbps, the Ethernet link may not be established between the device and the module through the SERVICE port. ● You can configure the SERVICE port either online or offline. ● In port mirroring mode, the SERVICE port acts like a read-only port. That is, you cannot access devices (ping, connect to Control Expert, etc.) through the SERVICE port. Refer to Configuring the Service Port *(see page 145)*.
INTERLINK	1	The INTERLINK port provides connectivity to other Quantum EIO head modules on the local rack.
DEVICE NETWORK	2	These 2 copper ports provide: ● connections for distributed I/O communications ● cable redundancy via the daisy chain loop architecture

Module Specifications

Product Certification

The Quantum EIO head/adapter modules meet these standards:

UL (UL508)
CSA (CSA22.2 no. 142)
C-tick
Hazardous locations (Cl1 div 2)
IEC61000-4-16
EMI EN 55011
CE
EN 61131-1
IEC 61131-2 (zone B and zone C, except surges on AC: zone B only)

Quantum EIO modules conform to these product certification and marine classification authorities:

Key	Certification Body	Country
ABS	American Bureau of Shipping	United States
BV	Bureau Veritas	France
DNV	Det Norske Veritas	Norway
GOST	Gosudarstvennyy Standart	Russia
GL	Germanischer Lloyd	Germany
LR	Lloyd's Register	United Kingdom
RINA	Registro Italiano Navale	Italy

The electrical isolation within a Quantum EIO system modules complies with the 1500 Vac/2250 Vdc 60s from IEEE 802.3 2008.

Environmental Requirements

Parameter	Reference	Specification
protection	EN 61131-2	IP20
	IEC 60527	
protection class	EN 61131-2	protection class 1
over voltage class	EN 61131-2	category II
operating temperature	IEC 60068-2-1	0 ... 60° C
	Ab&Ad (cold)	
	IEC 60068-2-2	
	Bb&Bd (cold)	
storage temperature	IEC 60068-2-1	-40 ... 85° C
	Ab&Ad (cold)	
	IEC 60068-2-2	
	Bb&Bd (cold)	
sinusoidal vibration	IEC 60068-2-6fC	• .5 ... 8.4 Hz at 3.5 mm constant amplitude • 8.4 ... 150 Hz at 1g constant acceleration • 10 cycles at sweep rate of 1 oct/min
	EN 61131-2	
operating shock	IEC 60068-2-27Ea	30 g peak, 11 ms, half-sine wave, 3 shocks in each direction (+ and -) for each of the 3 principle axes
altitude		0 ... 5000 m maximum during operation. For altitudes > 2000 m, reduce the operating temperature by 6° C for each additional 1000 m.
free fall, random (packaged)	EN 61131-2	5 random drops from 1 m onto flat surfaces
	IEC 60068-2-32	
	test ed., method 1	
free fall, flat drop (unpackaged)	EN 61131-2	• 2 random drops from 1 m onto flat surfaces • 5 drops from 0.1 m onto flat surfaces
	IEC 60068-2-32	
	test ed., method 1	
free fall, angled (unpackaged)	EN 61131-2	5 drops from 0.1 m onto each corner
	IEC 60068-2-31	
relative humidity (operating)	IEC 60068-2-78Ca	93% (+/- 2%, noncondensing) at 60° C for conformally coated modules 140CRA31200C and 140CRP31200C
relative humidity (nonoperating)	IEC 60068	93% (+/- 2%, noncondensing) at 60° C for conformally coated modules 140CRA31200C and 140CRP31200C

NOTE: The BMXCRA31210 is also available in a coated version.

Communication Specifications

Introduction

The following specifications describe both the I/O communication and the explicit messaging capacities of the 140NOC78000 distributed I/O head module.

I/O Communication Specifications

The 140NOC78000 module presents the following I/O communication features:

Communication Type	Feature		Capacity
EtherNet/IP (CIP Implicit Messaging)	scanner		
		maximum number of devices	128 devices (125 devices as scanner + 3 devices as adapter) shared with Modbus TCP
		maximum message size	511 bytes
	adapter		
		maximum number of instances	3 adapter instances
		maximum number of connections	2 connections per instance
		maximum message size	511 bytes including header
		inputs	505 bytes excluding header
		outputs	509 bytes excluding header
Modbus TCP (Modbus Scanner)	maximum number of registers		
		read	125 registers
		write	120 registers
		maximum number of devices	128 devices shared with EtherNet/IP
	maximum message size		
		read	250 bytes (125 words) excluding header
		write	240 bytes (120 words) excluding header

I/O Data Exchange with the CPU		
Feature	**Capacity**	**Comments**
maximum total input data size	4 kb	4 kb of data includes user configurable data and overhead. The overhead includes module diagnostic data, data object headers, and the number of headers depending on the user configuration. As a result, the user configurable data size is less than 4kb, but more than 3.5 kb.
maximum total output data size	4 kb	4 kb of data includes user configurable data and overhead. The overhead includes module control data, data object headers, and the number of headers depending on the user configuration. As a result, the user configurable data size is less than 4kb, but more than 3.5 kb.

Explicit Messaging Specifications

The 140NOC78000 module presents the following explicit messaging features:

Communication Type	Feature		Capacity
EtherNet/IP (CIP Explicit Messaging)	client		
		maximum number of simultaneous connections	16 connections
		maximum number of concurrent requests	16 requests, shared with Modbus TCP
	server		
		maximum number of simultaneous connections	32 connections
		maximum messaging size	1023 bytes
Modbus TCP (Modbus Explicit Messaging)	client		
		maximum number of simultaneous connections	16 connections
		maximum number of concurrent requests	16 requests, shared with EtherNet/IP
	server		
		maximum number of requests that can be transferred to the CPU per scan	12 connections
		maximum number of simultaneous connections	32 connections
	maximum message size		
		read	250 bytes (125 words) excluding header
		write	240 bytes (120 words) excluding header

Chapter 2
Installing the 140NOC78000 Module

Introduction

This chapter describes the installation process of the 140NOC78000 module and other distributed I/O network devices within a Quantum EIO system.

What Is in This Chapter?

This chapter contains the following topics:

Mounting a Quantum EIO-Compatible Module on the Backplane

Introduction

Use these instructions to install Quantum and Modicon X80 modules that operate in a Quantum EIO system:

- 140CRP31200 remote I/O head module (on the local rack)
- remote I/O adapter module:
  - 140CRA31200 remote I/O adapter module (on a Quantum remote I/O drop)
  - BMXCRA312•0 adapter module (on a Modicon X80 remote I/O drop)
  - 140CRA31908 adapter module facilitates the use of S908 hardware and applications in M580 Ethernet I/O architectures.
- 140NOC78000 distributed I/O head module (on the local rack)
- 140NOC78100 control head module (on the local rack)

Grounding Considerations

Do not apply power to a Quantum rack until connections are made at both ends of the Ethernet cable. For example, connect the cable to both the 140CRP31200 and another device (adapter module) or ConneXium dual-ring switch DRS before you turn on the power.

Refer to the *Quantum EIO System Planning Guide* for details on dual-ring switches (DRSs).

DANGER

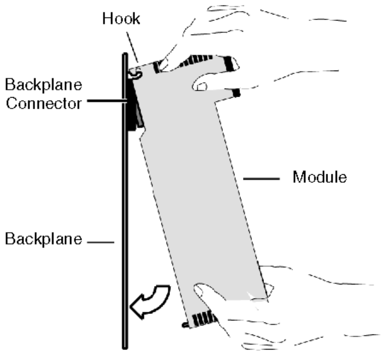
ELECTRICAL SHOCK HAZARD

- Switch off the power supply to the automation controller stations at both ends of the connection before inserting or removing an Ethernet cable.
- Use suitable insulation equipment when inserting or removing all or part of this equipment.

Failure to follow these instructions will result in death or serious injury.

Use fiber-optic cable to establish a communications link when it is not possible to master potential between distant grounds.

Mounting a Module

Step	Action
1	Hold the module at an angle and mount it on the 2 hooks near the top of the backplane. The figure shows the correct way to hold the module: 
2	Swing the module down so the connector engages the backplane connector.
3	Use a Phillips-head screw driver to tighten the screw at the bottom of the module from 2 to 4 in-lbs or from .22 through .45 N•m of torque.

NOTE: The figure above shows a **Quantum** module being mounted from top to bottom. Modicon X80 modules mount from bottom to top.

Replacing a Module

You can replace a Quantum EIO module at any time using another module with compatible firmware. The replacement module obtains its operating parameters over the backplane connection from the CPU. The transfer occurs immediately at the next cycle to the device.

The operating parameters that the CPU sends to a replacement module do not include any parameter values that were edited in the original module using explicit messaging **SET** commands.

Quantum EIO-Compatible Module Installation Considerations

Introduction

Observe the following guidelines when you install these Quantum and Modicon X80 modules in a Quantum EIO system:

- 140CRP31200 remote I/O head module (on the local rack)
- remote I/O adapter module (on the remote I/O drop)
  - 140CRA31200 module on a Quantum drop
  - 140CRA31908 adapter module on a Quantum drop
  - BMXCRA312•0 module on a Modicon X80 drop
- 140NOC78000 distributed I/O head module (on the local rack)
- 140NOC78100 control network head module (on the local rack)

Grounding Considerations

DANGER

ELECTRICAL SHOCK HAZARD

- Switch off the power supply to the automation controller stations at both ends of the connection before inserting or removing an Ethernet cable.
- Use suitable insulation equipment when inserting or removing all or part of this equipment.

Failure to follow these instructions will result in death or serious injury.

Use fiber-optic cable to establish a communications link when it is not possible to master potential between distant grounds.

NOTE: Refer to the ground connections information in *Electrical installation guide*.

Installation

You can apply power to the Quantum EIO controller rack after the 140CRP31200 head module or adapter module is inserted:

- Successful installation:
  - Initialization is finished.
  - Interconnections to other modules are validated (drop adapter module only).
- Unsuccessful installation:
  - Initialization does not finish.
  - Interconnections to other modules are not validated (adapter modules only).

You can see the status of the installation on the LED display.

NOTE: Because all modules on the local rack are initialized when power is applied, the 140CRP31200 remote I/O head module can only validate the interconnections with 140NOC78000 and 140NOC78100 head modules after these modules have been initialized. Therefore, the adapter module waits until its queries about the interconnected port information are answered.

NOTE: These guidelines pertain to the installation of a single head module or adapter module, not the entire network. For network power-up guidelines, refer to *Quantum EIO, System Planning Guide*.

Cable Installation

Introduction

We recommend the use of shielded twisted 4-pair CAT5e (10/100 Mbps) cables for the following connection types in a Quantum EIO system:

- the interlink connection between 140NOC78•00 modules and a 140CRP31200 remote I/O head module on the local rack
- the connection between 140NOC78•00 modules and DRSs on the main ring
- the connection between a 140NOC78000 distributed I/O head module and an isolated or extended distributed I/O network

NOTE: We recommend that copper shielded twisted 2-pair CAT5e (10/100 Mbps) and CAT6 (10/100/1000 Mbps) cables not be used. Rather, we recommend that you use copper shielded twisted 4-pair CAT5e (10/100 Mbps and CAT6 (10/100/1000 Mbps) cables.

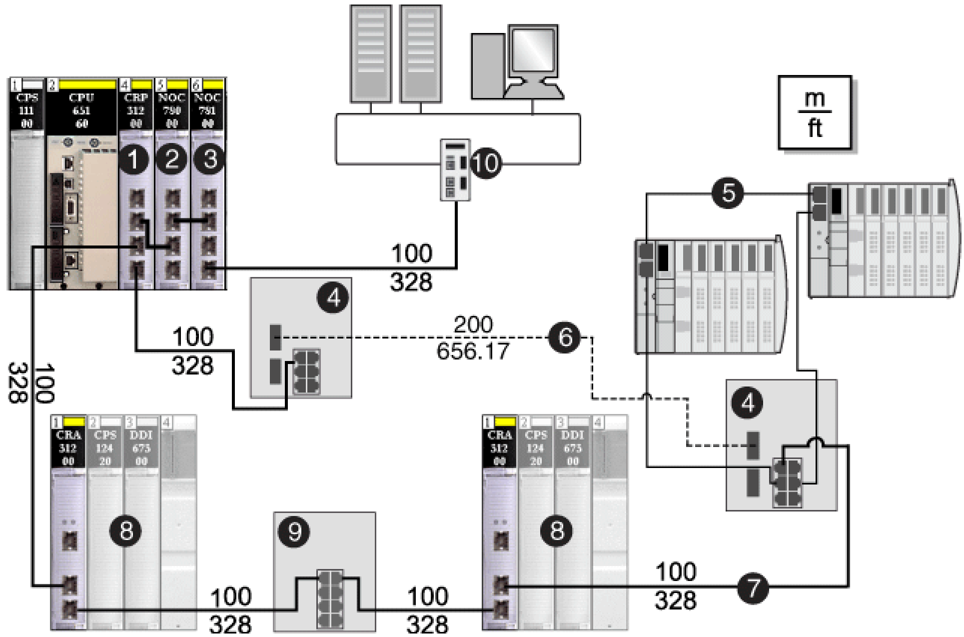
NOTE: We recommend the use of CAT6 (10/100/1000 Mbps) copper shielded twisted 4-pair cables for the connection between a 140NOC78100 control head module and a control network.

NOTE: Regarding shielded twisted 4-pair CAT5e (10/100 Mbps) cables, we recommend ConneXium 490NT•000•• cables.

Connections Between Devices

This example shows the maximum cable lengths between remote I/O and distributed I/O devices and a control network in a Quantum EIO installation.

NOTE: Use copper cable for distances less than or equal to 100 m. User fiber cable for distances greater than 100 m.



1 140CRP31200 remote I/O head module to manage remote I/O devices
2 140NOC78000 distributed I/O head module to manage distributed I/O devices
3 140NOC78100 control head module to provide transparency between the device network and the control network (10)
4 DRSs (with copper and fiber ports): These DRSs extend the distance between devices (up to 15 km).
5 Ethernet distributed I/O sub-ring
6 fiber portion of the main ring
7 copper portion of the main ring
8 Ethernet remote I/O drops on the main ring
9 DRS (with copper ports): This DRS serves to extend the distance between other devices.
10 control network

Chapter 3
Planning and Designing a Distributed I/O Network within a Quantum EIO System

Introduction

In a Quantum EIO system, you can design an architecture that contains both remote I/O and distributed I/O devices operating on the same Ethernet remote I/O network. Through the use of a distributed I/O head module installed on the local rack and DRSs installed on the main ring, you can connect distributed I/O devices to the remote I/O network. The head module scans the distributed I/O devices, and the DRSs control the data flow from distributed I/O devices so that the high priority of remote I/O traffic is maintained.

NOTE: The architectures described in this document have been tested and validated in various scenarios. If you intend to use architectures different than the ones described in this document, test and validate them thoroughly before implementing.

What Is in This Chapter?

This chapter contains the following sections:

Section 3.1
How Distributed I/O Networks Participate in the Quantum Ethernet I/O System

Introduction

A distributed I/O network within a Quantum EIO system is an Ethernet-based network that contains distributed I/O devices, adapter class devices, DRSs, HMIs, SCADA, SNTP server, etc. This network communicates with one or several 140NOC78000 head modules on the Quantum local rack, and allows distributed I/O devices to exist on an Ethernet remote I/O network.

To provide determinism on the Ethernet remote I/O network, use DRSs to connect the distributed I/O devices. These switches control the data flow from distributed I/O devices so that the high priority of the remote I/O network is not disturbed.

For more information about network performance and determinism, refer to the *Quantum Ethernet I/O Global System Planning Guide*.

What Is in This Section?

This section contains the following topics:

How Distributed I/O Networks Co-exist with Remote I/O Networks

Introduction

Remote I/O devices and distributed I/O devices can co-exist on a device network within a Quantum EIO system.

A *device network* is an Ethernet remote I/O network that contains for example:
- a local rack
- remote I/O drops
- distributed I/O devices
- DRSs
- adapters

In this type of network, remote I/O traffic has the highest priority on the network, so it is delivered ahead of distributed I/O traffic in the event of network congestion, providing deterministic remote I/O exchanges.

A 140NOC78000 distributed I/O head module installed on the local rack communicates with distributed I/O devices and client applications on the device network *(see page 56)*, isolated distributed I/O networks *(see page 59)*, independent distributed I/O networks *(see page 61)*, and extended distributed I/O networks *(see page 63)*. In addition, this head module provides services to communicate with PLC applications running on the control network.

DRSs connect the distributed I/O devices on the device network and provide determinism on the Ethernet remote I/O network. Download the specific predefined configuration files to these dual-ring switches, which control the data flow from distributed I/O devices so that the high priority of the remote I/O network is maintained.

Distributed I/O Devices

In a Quantum EIO system, *distributed I/O devices* can be:

- **Connected** to the Ethernet remote I/O network. A 140CRP31200 remote I/O head module is interlinked with a 140NOC78000 distributed I/O head module on the local rack to create a *device network*. The distributed I/O devices are connected via a DRS located on the main ring. Special types of distributed I/O devices that have 2 Ethernet ports and support *RSTP* may be connected directly to a *sub-ring*. Many types of distributed I/O devices may be connected to the DRS as distributed I/O clouds.
  (*Refer to the distributed I/O clouds topic (see page 40) to see the graphic showing the 140 CRP 312 00 head module interlinked with a 140 NOC 780 00 head module on the local rack to support distributed I/O devices*.)
  You can also connect distributed I/O devices that are part of an existing distributed I/O network, which you may not want to reconfigure.
  - Connect the interlink port (ETH2) of the 140NOC78000 module to the extended port (ETH1) of the 140NOC78100 module for the existing distributed I/O network to be a physical part of the Quantum EIO system. This network is called an extended distributed I/O network *(see page 63)*.
  - Connect the interlink port (ETH2) of the 140NOC78000 module to the extended port (ETH2) of the 140NOC78100 module. The distributed I/O devices are not a physical part of the Quantum EIO device network, but they do communicate with the *control network*. This network is called an independent distributed I/O network *(see page 61)*.

  – and/or –
- **Not connected** to the Ethernet remote I/O network. The devices are connected directly to a 140NOC78000 head module on the local rack as distributed I/O clouds, consisting of devices such as TeSys T motor drives, islands of STB devices, SCADA and HMI devices, and PCs. If you use a device that has 2 Ethernet ports and supports RSTP, you can connect the device in a star, a daisy chain, or a ring topology. In this instance, these distributed I/O devices are isolated and are not a physical part of the Ethernet remote I/O network.
  (*Refer to the distributed I/O clouds topic (see page 40) to see the graphic showing the 140 CRP 312 00 head module **not** interlinked with the 140 NOC 780 00 head module on the local rack, where there are no distributed I/O devices physically connected to the Ethernet remote I/O network*.)

NOTE: Distributed I/O devices can be connected to the Quantum EIO network via DRSs or via the service ports on the 140CRP31200 head module or ••• CRA 312 00 adapter module. They cannot be connected directly to the main remote I/O ring.

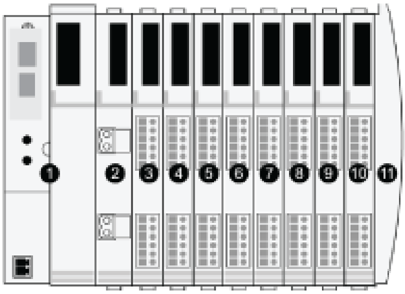
NOTE: If the device, which is connected to the service port, is configured for a speed that exceeds 100 Mbps, the Ethernet link may not be established between the device and the module through the service port.

The maximum load the network can process from distributed I/O devices is:

- 5 Mbps per DRS port or service port
- 20 Mbps for distributed I/O traffic on the main ring

Example of a Distributed I/O Device:

Shown throughout this documentation is an Advantys STB island. When an STB island is used with an STB NIP 2311 network interface module (NIM), the island can be connected directly to a distributed I/O sub-ring. The STB NIP 2311 NIM has 2 Ethernet ports and it supports RSTP, making it able to operate on a sub-ring.



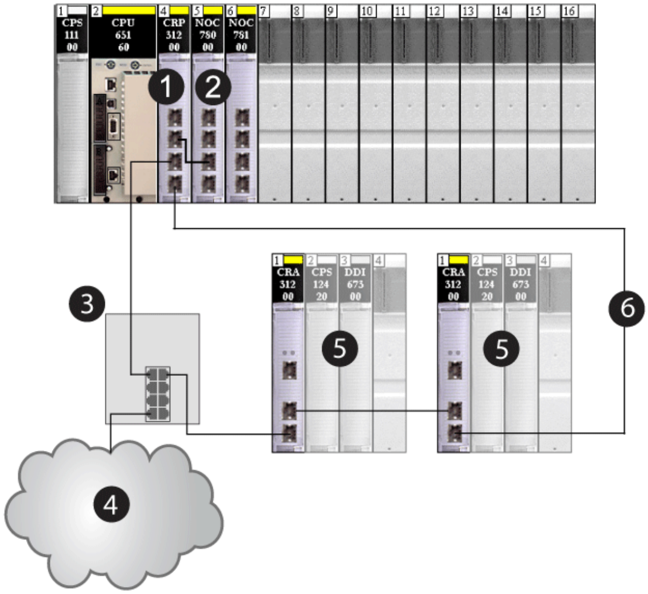
1 STB NIP 2311 NIM
2 STB PDT 3100 (24 VDC power distribution module)
3 STB DDI 3230 24 VDC (2-channel digital input module)
4 STB DDO 3200 24 VDC (2-channel digital output module)
5 STB DDI 3420 24 VDC (4-channel digital input module)
6 STB DDO 3410 24 VDC (4-channel digital output module)
7 STB DDI 3610 24 VDC (6-channel digital input module)
8 STB DDO 3600 24 VDC (6-channel digital output module)
9 STB AVI 1270 +/-10 VDC (2-channel analog input module)
10 STB AVO 1250 +/-10 VDC (2-channel analog output module)
11 STB XMP 1100 (island bus termination plate)

Distributed I/O Clouds

A *distributed I/O cloud* is a group of distributed I/O devices that are not required to support *RSTP*. Distributed I/O clouds require only a single (non-ring) copper wire connection. They can be connected to some of the copper ports on DRSs, or they can be connected directly to 140NOC78000 modules in the *local rack*. Distributed I/O clouds cannot be connected to *sub-rings*.

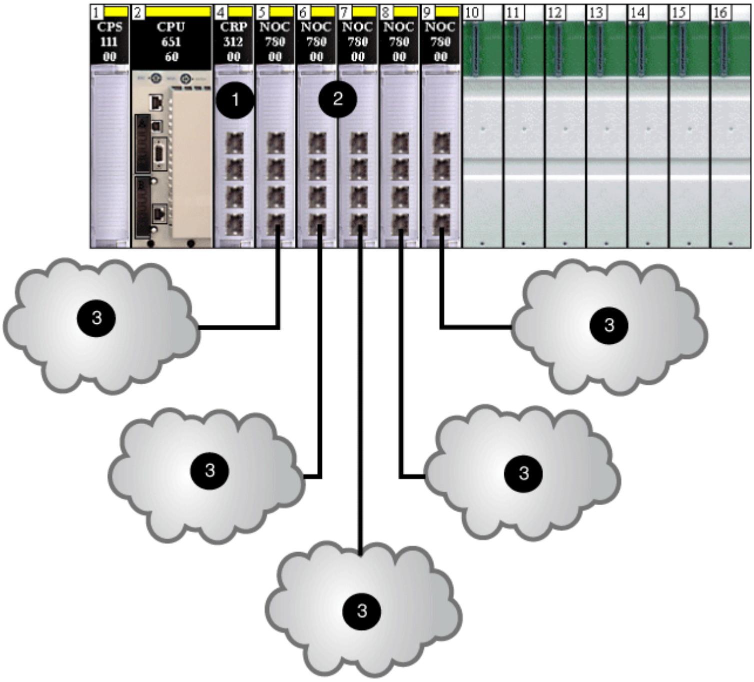
When a distributed I/O cloud is connected to a DIO cloud port on a DRS in the *main ring*, the distributed I/O devices within the cloud are a physical part of the Quantum EIO network.

In this instance, interlink the 140 CRP 312 00 remote I/O head module with a 140NOC78000 module in the local rack, since the 140NOC78000 module manages the distributed I/O devices.



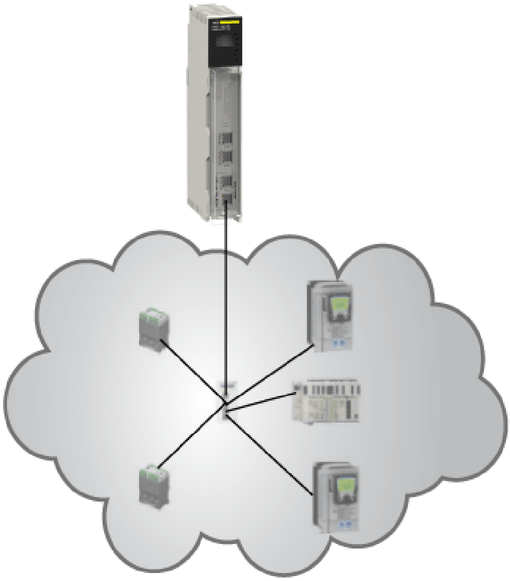
1 140CRP31200 remote I/O head module
2 140NOC78000 distributed I/O head module
3 DRS
4 distributed I/O cloud
5 remote I/O drop
6 main ring

When a distributed I/O cloud is connected directly to a 140NOC78000 module in the local rack, the distributed I/O devices are *isolated* from the remote I/O network.



1 140CRP31200 remote I/O head module
2 140NOC78000 distributed I/O head modules
3 distributed I/O clouds

Distributed I/O clouds contain either a single device or several devices designed in daisy chain topologies. The following example shows a distributed I/O cloud with daisy-chained devices.

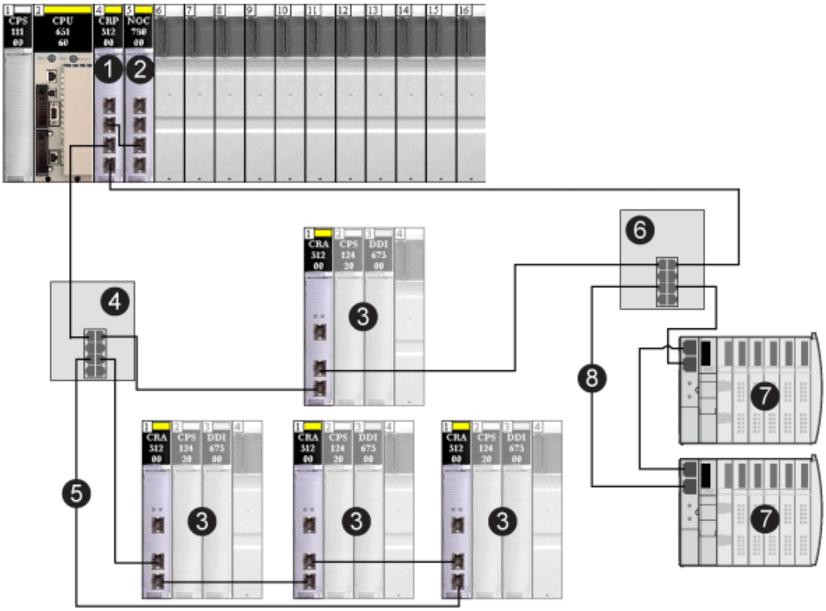


Distributed I/O Sub-rings

Distributed I/O sub-rings are connected to the *main ring* via DRSs. They contain only distributed I/O devices that have 2 Ethernet ports and support *RSTP*. A distributed I/O sub-ring supports a maximum of 128 distributed I/O devices.

NOTE: You cannot combine remote I/O devices and distributed I/O devices in the same sub-ring.

The following graphic shows a remote I/O sub-ring (5) and a distributed I/O sub-ring (8).



1 140CRP31200 remote I/O head module
2 140NOC78000 distributed I/O head module
3 remote I/O drop
4 DRS configured to support a remote I/O sub-ring
5 remote I/O sub-ring
6 DRS configured to support a distributed I/O sub-ring
7 distributed I/O devices (STB island)
8 distributed I/O sub-ring

NOTE: DRSs may have different predefined configuration files to support devices in a Quantum EIO system. For details, refer to the *Quantum EIO System Planning Guide*.

Rules for Interconnectivity

Introduction

The Quantum local rack within a Quantum EIO system can have different combinations of Ethernet head modules. This topic describes the types of networks created when the 140NOC78000 distributed I/O head modules interconnect with other head modules on the local rack.

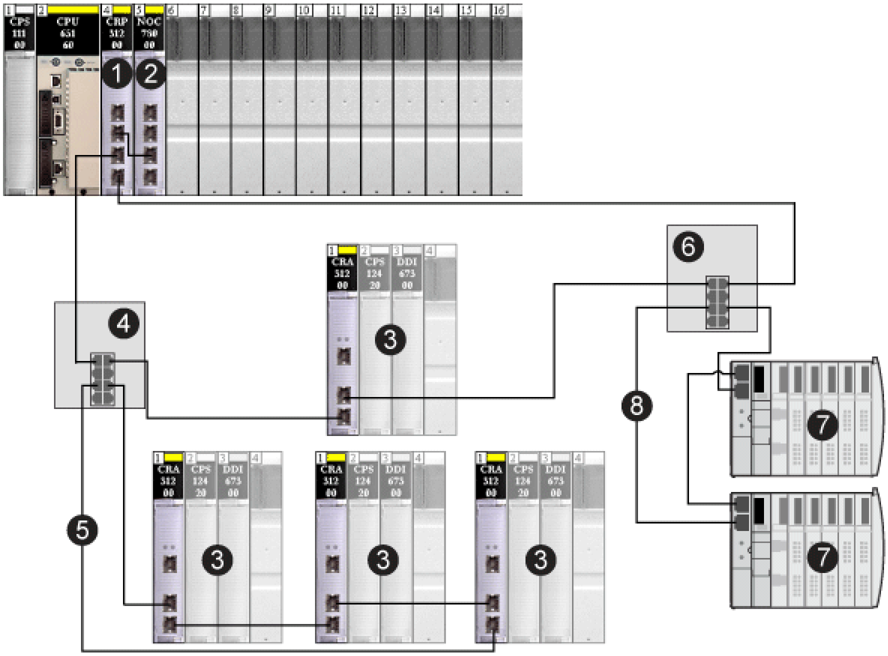
A local rack contains one 140CRP31200 remote I/O head module and up to 6 communication modules (only one of which can be a 140NOC78100 control head module). Use the interlink port on the 140NOC78000 module to interlink with the other head modules on the local rack for various network combinations:

140NOC78000 Module Interconnectivity	Network Type	Description
up to three 140NOC78000 modules interlinked with the 140CRP31200 module	device network	combination of remote I/O and distributed I/O devices
up to six 140NOC78000 modules <u>not</u> interlinked with any other head modules	isolated distributed I/O network	distributed I/O devices that are not a physical part of the Quantum EIO system
up to five 140NOC78000 modules, only one of which is interlinked with the 140NOC78100 module	independent distributed I/O network	distributed I/O devices located on an existing distributed I/O network that utilize control network transparency in the Quantum EIO system
• up to three 140NOC78000 modules interlinked with the 140CRP31200 module • one of these three modules is interlinked with the *service/extend port* of the 140NOC78100 module	extended distributed I/O network	distributed I/O devices located on an existing distributed I/O network that are a physical part of the Quantum EIO system
up to three 140NOC78000 modules interlinked with the 140CRP31200 module and the 140NOC78100 module	control network transparency	provides Ethernet transparency between the device network and the control network

NOTE: The types of distributed I/O topologies you can use in a Quantum EIO system are discussed in the Selecting the Correct Distributed I/O Topology *(see page 55)* topic.

Interlinking the 140NOC78000 Module in a Device Network

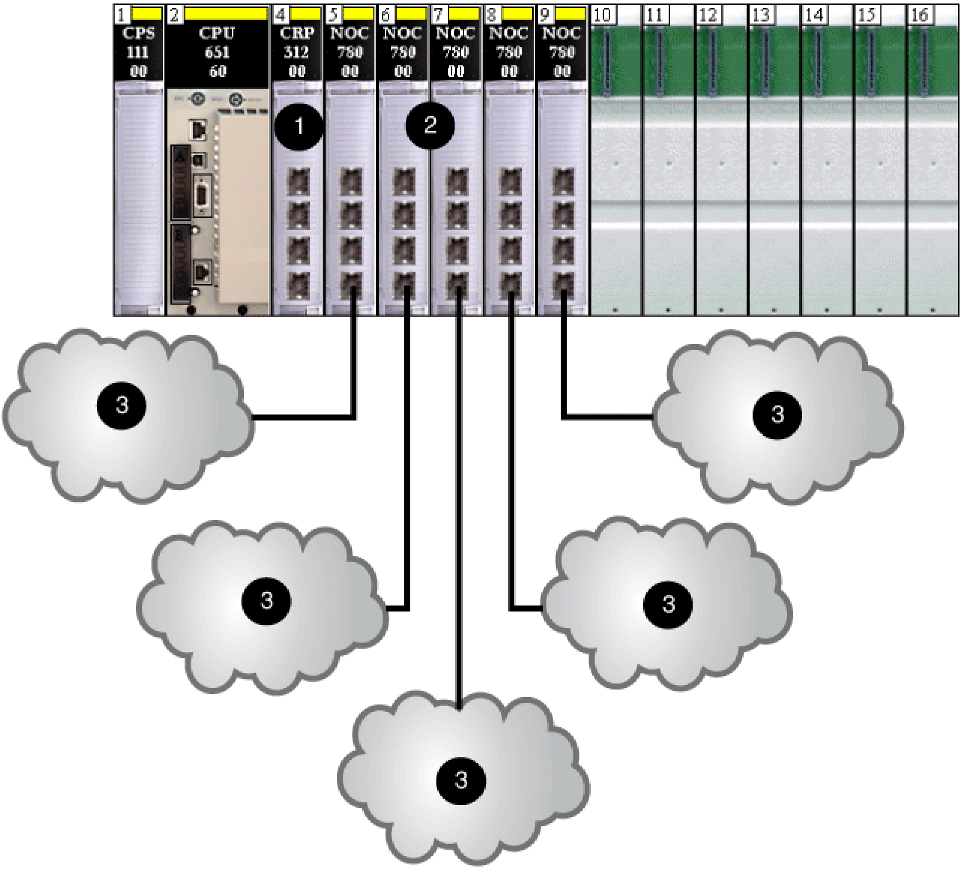
Interlink the 140NOC78000 module with the 140CRP31200 module to create a device network:



1 140CRP31200 remote I/O head module
2 140NOC78000 distributed I/O head module
3 remote I/O drop on the main ring
4 DRS on the main ring connecting the remote I/O sub-ring
5 remote I/O sub-ring
6 DRS on the main ring connecting the distributed I/O sub-ring
7 distributed I/O devices
8 distributed I/O sub-ring

Interlinking the 140NOC78000 Module in an Isolated Distributed I/O Network

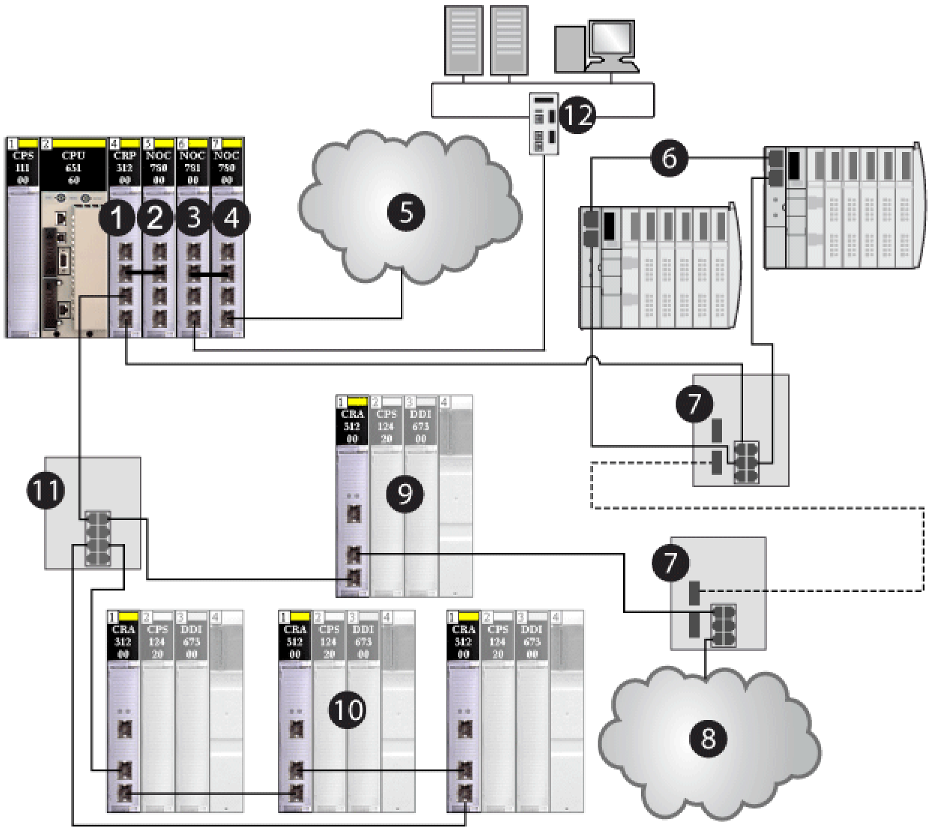
The 140NOC78000 modules are not connected any other head module on the local rack.



1 140CRP31200 remote I/O head module
2 140NOC78000 distributed I/O head modules (not interlinked with the 140CRP31200 module
3 distributed I/O clouds

Interlinking the 140NOC78000 Module in an Independent Distributed I/O Network

Interlink a 140NOC78000 distributed I/O head module with the 140NOC78100 control head module to provide Ethernet transparency between the isolated distributed I/O network [your existing distributed I/O network (5)] and the control network. You can also interlink another 140NOC78000 module with the 140CRP31200 remote I/O head module, but your existing distributed I/O network is not a physical part of the Quantum EIO device network (To configure the IP forwarding service in the 140NOC78100 module that provides transparency, refer to the *Quantum EIO Control Network Installation and Configuration Guide*.

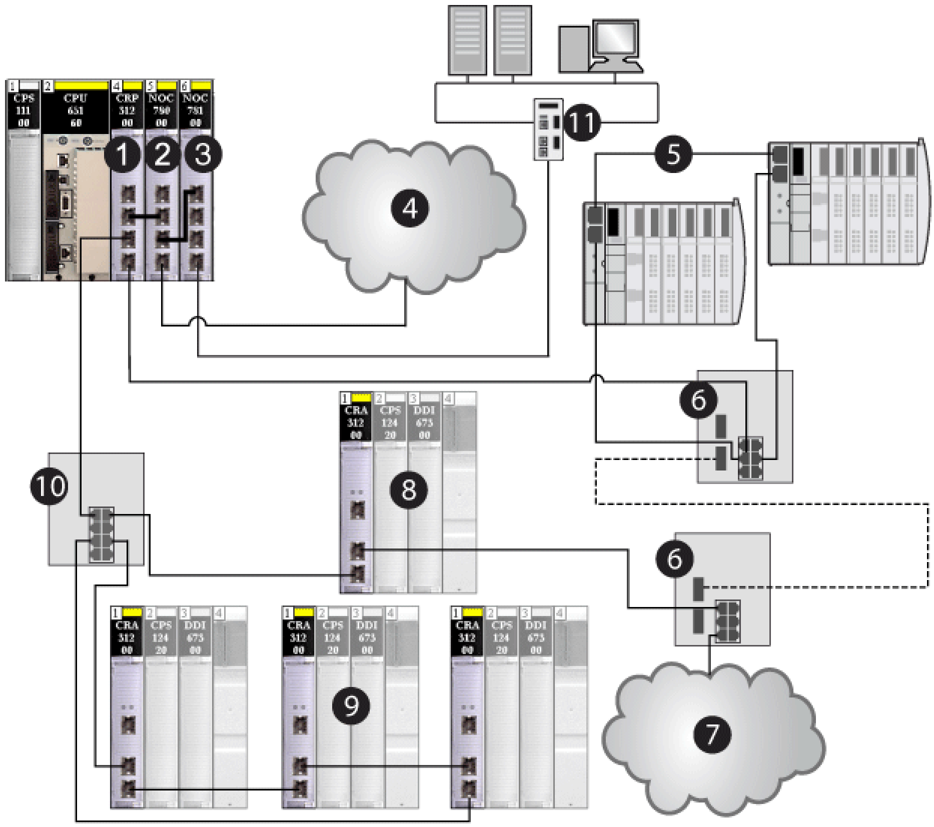


1 140CRP31200 remote I/O head module
2 140NOC78000 distributed I/O head module (interlinked with the 140CRP31200 module to support the device network)
3 140NOC78100 control head module
4 140NOC78000 module (interlinked with the140NOC78100 module to support the independent distributed I/O network)
5 independent distributed I/O network
6 distributed I/O sub-ring

7 DRSs with a predefined configuration file to support copper-to-fiber and fiber-to-copper transitions on the main ring
8 distributed I/O cloud
9 remote I/O drop on the main ring
10 remote I/O sub-ring
11 DRS connecting the remote I/O sub-ring to the main ring
12 control network (which also monitors the independent distributed I/O network)

Interlinking the 140NOC78000 Module in an Extended Distributed I/O Network

Interlink a 140NOC78000 distributed I/O head module with the *extended port* of the 140NOC78100 control head module to provide Ethernet transparency between the isolated distributed I/O network [your existing distributed I/O network (4)] and the control network. Interlink the same 140NOC78000 module with the 140CRP31200 remote I/O head module for your existing distributed I/O network to be a physical part of the Quantum EIO device network. (To configure the service port for an extended network and to configure the IP forwarding service in the 140NOC78100 module that provides transparency, refer to the *Quantum EIO Control Network Installation and Configuration Guide*.)



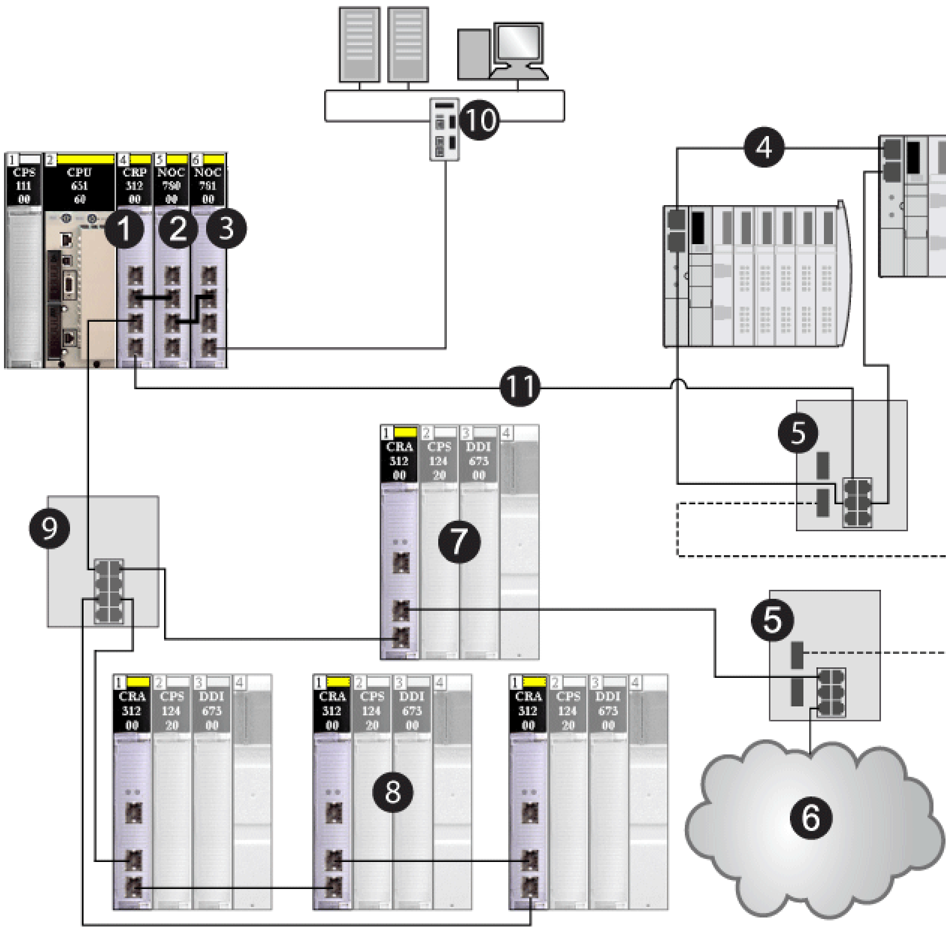
1 140CRP31200 remote I/O head module
2 140NOC78000 distributed I/O head module (interlinked with the *extend port* of the 140NOC78100 module to support the extended distributed I/O network and also interlinked with the 140CRP31200 module to support the device network)
3 140NOC78100 control head module (interlinked with the 140NOC78000 module to provide transparency between the device network and the control network)
4 extended distributed I/O network
5 distributed I/O sub-ring

6 DRSs with a predefined configuration file to support copper-to-fiber and fiber-to-copper transitions on the main ring
7 distributed I/O cloud
8 remote I/O drop on the main ring
9 remote I/O drops on the remote I/O sub-ring
10 DRS connecting the remote I/O sub-ring to the main ring
11 control network

NOTE: The types of distributed I/O topologies you can use in a Quantum EIO system are discussed at Selecting the Correct Distributed I/O Topology *(see page 55)*.

Interlinking the 140NOC78000 Module for Control Network Transparency

Interlink a 140NOC78100 control head module with a 140NOC78000 distributed I/O head module, which is interlinked with a 140CRP31200 remote I/O head module, to provide Ethernet transparency between the device network and the control network.



1 140CRP31200 remote I/O head module
2 140NOC78000 distributed I/O head module (interlinked with the 140NOC78100 module to support the device network)
3 140NOC78100 control head module (interlinked with the 140NOC78000 module to provide Ethernet transparency between the control network and the device network)
4 distributed I/O sub-ring
5 DRSs with a predefined configuration file to support copper-to-fiber and fiber-to-copper transitions on the main ring
6 distributed I/O cloud

7 remote I/O drop on the main ring
8 remote I/O drops on the remote I/O sub-ring
9 DRS connecting the remote I/O sub-ring to the main ring
10 control network
11 main ring

Connecting Distributed I/O Devices to a Quantum EIO System

Distributed I/O Devices

Distributed I/O devices can be connected to a Quantum EIO system in the following 2 ways:

- a distributed I/O cloud *(see page 40)*
- a distributed I/O sub-ring *(see page 43)*

Distributed I/O devices *(see page 38)* in a sub-ring have 2 Ethernet ports (to maintain the ring), and they support RSTP. An example of devices on a distributed I/O sub-ring would be several STB islands that use STB NIP 2311 NIMs.

Ethernet distributed I/O devices that can be put on distributed I/O clouds include 2 families of devices:

I/O Scanned Devices	Devices that Cannot be I/O Scanned
variable speed drives — Altivar ATV 32, 61, 71	Magelis HMI controllers
main protection and control functions — TeSys T	Pelco cameras
ETB (I/O modules), OTB (distributed I/O modules), and STB (modules connected on a single island)	
remote masters — Profibus master interface, ETG1000 master interface, Hart master interface	
Modbus TCP and EtherNet/IP distributed I/O devices	
third-party distributed I/O devices compatible with the Quantum EIO system	

Topology Choices

Your Ethernet remote I/O network will comprise of one of the following topologies:

- a simple daisy chain loop
- a high-capacity daisy chain loop

These 2 topologies, which are discussed later in this guide, are comprised of the devices in the following table. These devices and their Ethernet port types define how you will choose and build your topology.

To Insert in the Network...	Use...	Topology Type
distributed I/O devices with a single Ethernet port	a distributed I/O cloud (with devices in a star topology)	You can connect a distributed I/O cloud to a *high-capacity daisy chain loop*. ● A distributed I/O cloud participates in the remote I/O network only if it is connected to a DRS that resides on the main ring in a high-capacity daisy chain loop. In this case, interlink a 140CRP31200 head module with a 140NOC78000 head module on the local rack, since the 140NOC78000 module supports the distributed I/O cloud. **NOTE:** A distributed I/O cloud that is connected to a 140 NO• 780 •• module on the local rack within a *simple daisy chain loop* is *isolated* *(see page 59)*. The cloud is not physically part of the remote I/O network.
distributed I/O devices with dual Ethernet ports	● a distributed I/O cloud (with devices in a star topology) — or — ● a distributed I/O sub-ring (with devices in a daisy chain loop, if they support RSTP).	You can only connect a distributed I/O cloud or distributed I/O sub-ring via a DRS that resides on the main ring in a high-capacity daisy chain loop.

Section 3.2
Selecting the Correct Distributed I/O Topology

Introduction

This section discusses the 4 types of distributed I/O networks allowed in a Quantum EIO system.
- device network
- isolated distributed I/O network
- independent distributed I/O network
- extended distributed I/O network

What Is in This Section?

This section contains the following topics:

Planning a Device Network

Introduction

These are the advantages of a device network:
- You can integrate remote I/O and distributed I/O devices within the same system.
- You can integrate current and legacy (S908) devices from Schneider Electric, as well as third-party devices (which are connected through standard open interfaces).
- You can provide transparency to your entire system.
- Networks are simplified.
- Costs are reduced.

A device network uses a high-capacity daisy chain loop topology (Ethernet remote I/O drops and distributed I/O devices on the same physical network). Within a device network, you can add distributed I/O sub-rings and distributed I/O clouds.

DRSs connect distributed I/O devices on the device network. (You can also connect distributed I/O devices to the service port of a 140CRA31200 or BMXCRA31210 moudule.) This connection provides determinism on the Ethernet remote I/O network. Download specific predefined configuration files to these dual-ring switches, which control the data flow from distributed I/O devices so that the high priority of the remote I/O network is maintained.

Installing a Device Network

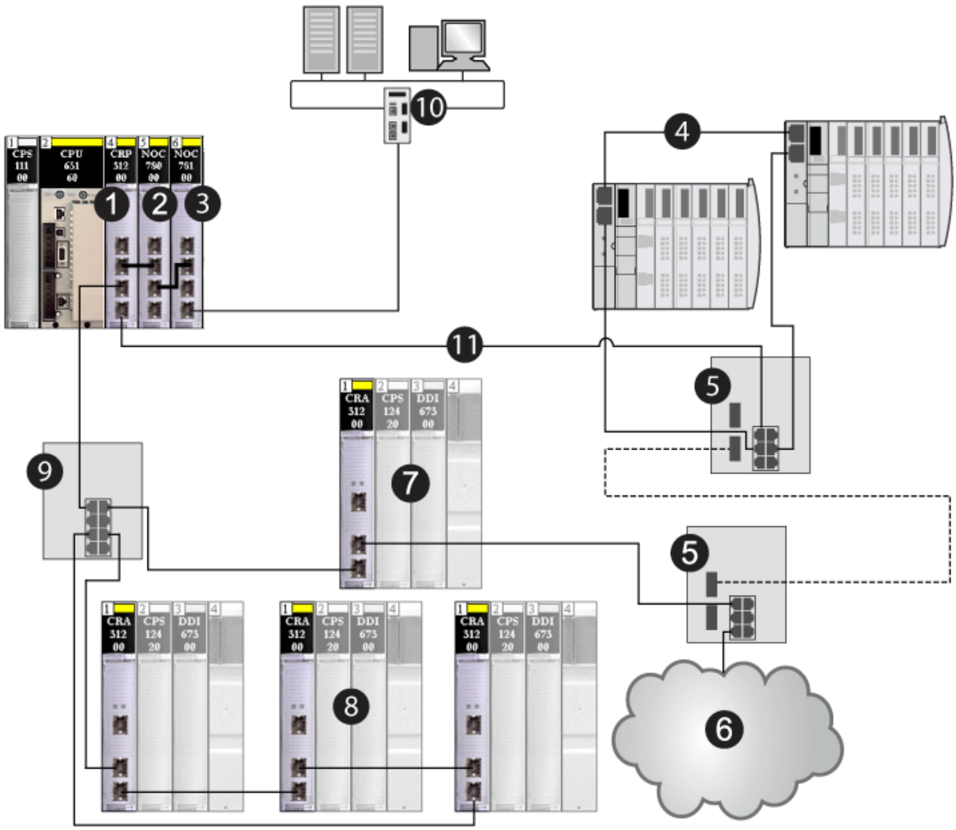
To install a device network, which allows both remote I/O and distributed I/O on the same Ethernet remote I/O network, as well as Ethernet transparency between the control network (by configuring the IP forwarding service in the 140NOC78100 control head module) and the device network, perform the following steps:

Step	Action
1	Install one 140CRP31200 remote I/O head module, one 140NOC78100 control head module, and up to six 140NOC78000 distributed I/O head modules on the local rack, only three of which can be interlinked with the 140CRP31200 module and one which can be interlinked with the 140NOC78100 module to support a device network.
2	Connect the *interlink port* (ETH 2) of the 140NOC78000 module to the *interlink port* (ETH 2) of the 140CRP31200 module.
3	Connect the *interlink port* (ETH 2) of the 140NOC78100 module to the *device network/interlink port* (ETH 3) of the 140NOC78000 module.
4	Connect the start of the main ring to the *device network port* (ETH 3 or ETH 4) of the 140CRP31200 module.
5	Connect the end of the main ring to the *device network port* (ETH 3 or ETH 4) of the 140CRP31200 module.

Step	Action
6	Connect DRSs to the main ring for distributed I/O sub-rings and/or distributed I/O clouds. Refer to the *Predefined Configurations File* topic in the *Quantum EIO System Planning Guide* for details on installing DRSs and distributed I/O devices.
7	Connect the *control network port* (ETH 3 or ETH 4) of the 140NOC78100 module to the control network.

Example

The following graphic shows a 140CRP31200 remote I/O head module and a 140NOC78000 distributed I/O head module interlinked on the local rack to support a device network.



1 140CRP31200 remote I/O head module
2 140NOC78000 distributed I/O head module interlinked with the 140CRP31200 module to manage the device network
3 140NOC78100 control head module interlinked with the 140NOC78000 module to provide transparency between the device network and the control network
4 distributed I/O sub-ring

5 DRS with a predefined configuration file to support copper-to-fiber and fiber-to-copper transitions on the main ring
6 distributed I/O cloud
7 Ethernet remote I/O drop on the main ring
8 Ethernet remote I/O sub-ring
9 DRS connecting the remote I/O sub-ring to the main ring
10 control network
11 main ring

Planning an Isolated Distributed I/O Network

Introduction

An isolated distributed I/O network is not part of the remote I/O network. It is an Ethernet-based network containing distributed I/O devices on a copper wire running from a single port connection. If you use dual-port distributed I/O devices that support RSTP, you can connect the devices in a daisy-chain loop to the 2 device network ports (ETH 3 and ETH 4) on the 140NOC78000 distributed I/O head module. There is no interlink to the remote I/O network.

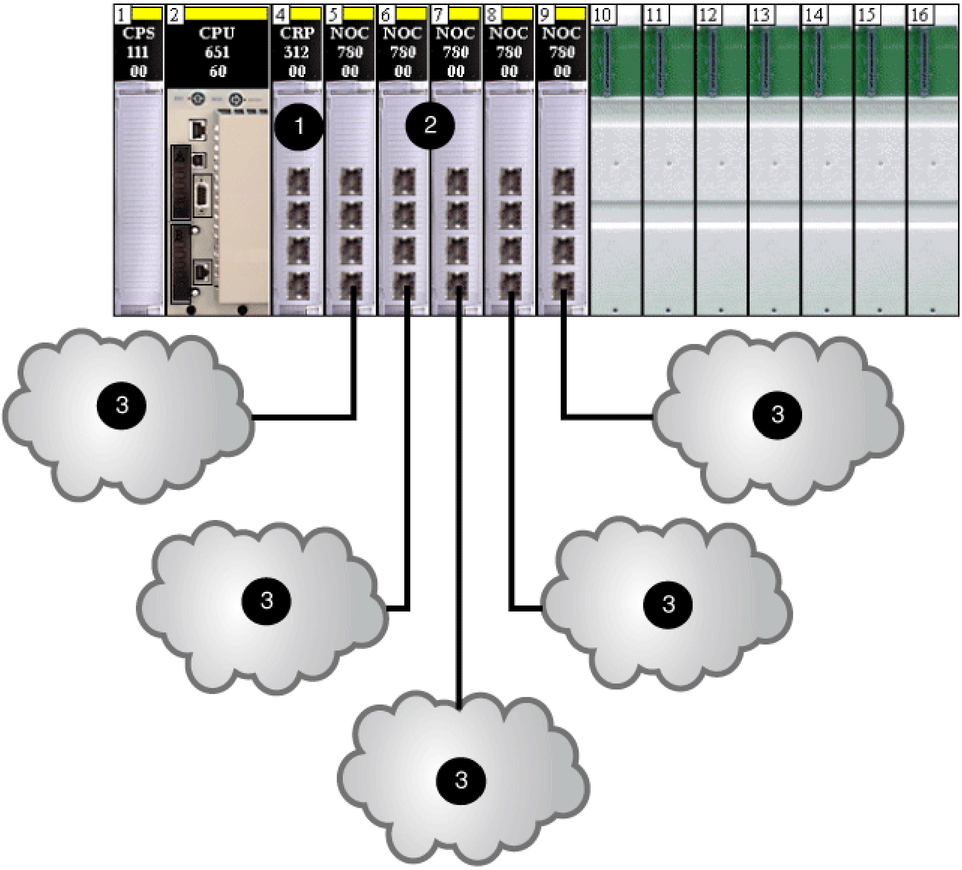
Attaching an Isolated Distributed I/O Network

To attach an isolated distributed I/O network to a Quantum EIO system:

Step	Action
1	Install up to six 140NOC78000 distributed I/O head modules on the local rack, one module for each isolated network desired.
2	Connect the *device network port* (ETH 4) of each 140NOC78000 module to each distributed I/O network.
3	If you use dual-port devices that support RSTP, then you can connect the devices in a daisy chain loop to both *device network ports* (ETH 3 and ETH 4) on the 140NOC78000 distributed I/O head module.
4	The 140NOC78000 modules are not connected to any other head module on the local rack.

Example

The following graphic shows multiple isolated distributed I/O networks. The 140NOC78000 distributed I/O head modules are not linked to the 140CRP31200 remote I/O head module on the local rack.



1 140CRP31200 remote I/O head module
2 140NOC78000 distributed I/O head modules
3 distributed I/O clouds, which do not communicate with the Quantum EIO networks

NOTE:
- A 140NOC78000 module on the local rack attaches the controller to the isolated distributed I/O network.
- A cloud can have a DRSs downloaded predefined configuration and a single connection to the 140NOC78000 module on the local rack and a ring of dual-port distributed I/O devices that support RSTP.

Adding an Independent Distributed I/O Network

Introduction

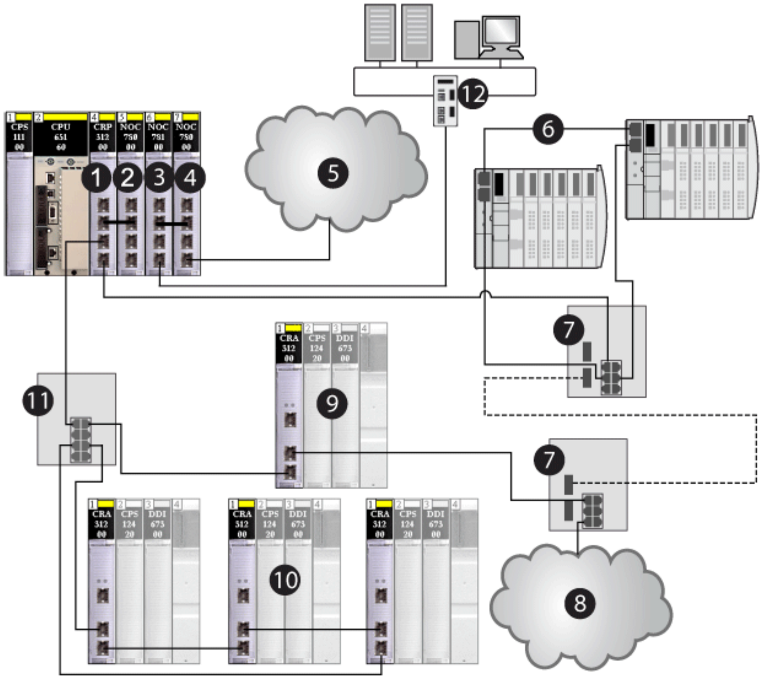
Add an independent distributed I/O network if you have existing distributed I/O devices — which you may not want to reconfigure — to the Quantum EIO system for the purpose of communicating with the control network.

An independent distributed I/O network is not part of the Ethernet remote I/O network, but it does communicate with the control network.

An independent distributed I/O network is an Ethernet-based network containing distributed I/O devices on a copper wire running from a single port connection. If you use dual-port devices that support RSTP, you can connect the devices in a daisy chain loop to the device network ports (ETH 3 and ETH 4) on the 140NOC78000 distributed I/O head module. There is no interlink to the remote I/O network. The 140NOC78000 module is interlinked with the the 140NOC78100 control head module on the local rack to support communication with the Quantum EIO control network only. The 140NOC78000 module can be interlinked with other 140NOC78000 modules. Neither the 140NOC78000 nor the 140NOC78100 modules can be interlinked with the 140CRP31200 remote I/O head module on the local rack.

Example

The following graphic shows an independent distributed I/O network (5). The 140NOC78000 distributed I/O head module is interlinked with the 140NOC78100 control head module. Neither the 140NOC78000 nor the 140NOC78100 module *in the independent distributed I/O network* is interlinked with the 140CRP31200 remote I/O head module. A second 140NOC78000 module is interlinked with the 140CRP31200 module to support distributed I/O devices on the device network:



1 140CRP31200 remote I/O head module
2 140NOC78000 distributed I/O head module interlinked with the 140CRP31200 module to support the device network
3 140NOC78100 control head module
4 second 140NOC78000 module interlinked with the 140NOC78100 module to support an independent distributed I/O network and communicate with the Quantum EIO control network
5 independent distributed I/O network, which communicates with the Quantum EIO control network
6 distributed I/O sub-ring
7 DRSs — with C4 predefined configuration files — for copper-to-fiber and fiber-to-copper transitions on the main ring
8 distributed I/O cloud
9 remote I/O drop on the main ring
10 remote I/O sub-ring
11 DRS — with a C1 predefined configuration file — on the main ring connected to the remote I/O sub-ring
12 control network

Planning an Extended Distributed I/O Network

Introduction

Plan an extended distributed I/O network if you have an existing distributed I/O network — which you may not want to reconfigure — that you want to add to the Quantum EIO system.

An extended distributed I/O network is an Ethernet-based network containing distributed I/O devices on a copper wire running from a single port connection. If you use dual-port devices that support RSTP, you can connect the devices in a daisy chain loop to the device network ports (ETH 3 and ETH 4) on the 140NOC78000 distributed I/O head module on the local rack. The network is connected to the 140CRP31200 remote I/O head module, making the extended distributed I/O network part of the Quantum EIO system. The 140NOC78000 module is connected to the service/extend port of the 140NOC78100 control head module to provide transparency between the control network and the extended distributed I/O network.

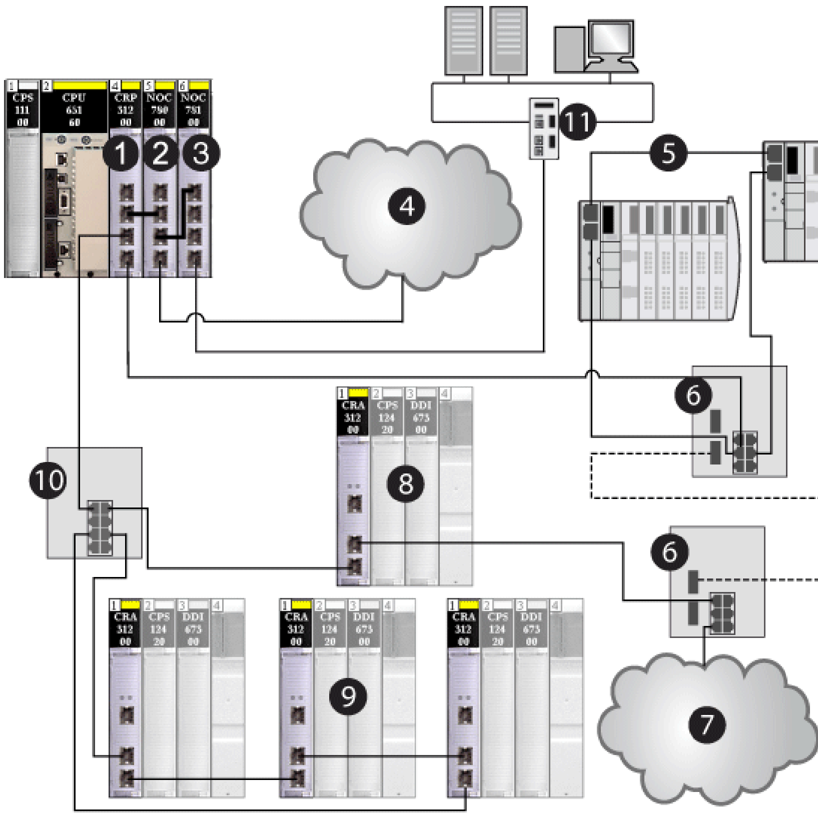
Attaching an Extended Distributed I/O Network

To attach an extended distributed I/O network to a Quantum EIO system:

Step	Action
1	Install one 140CRP31200 remote I/O head module, one 140NOC78100 control head module, and up to five 140NOC78000 distributed I/O head modules (up to 3 modules can be interlinked with the 140CRP31200 module and 1 can be interlinked with the 140NOC78100 module) on the local rack.
2	Connect the *interlink port* (ETH 2) of the 140NOC78000 module to the *interlink port* (ETH 2) of the 140CRP31200 module.
3	Connect the *device network/interlink port* (ETH 3) of the 140NOC78000 module to the *service/extend port* (ETH 1) of the 140NOC78100 module.
4	Connect the *device network* (ETH 4) of the 140NOC78000 module to your existing distributed I/O network.
5	Connect the *control network port* (ETH 3 or ETH 4) of the 140NOC78100 module to the control network.
6	Connect the start of the main ring to the *device network port* (ETH 3 or ETH 4) of the 140CRP31200 module.
7	Connect the end of the main ring to the *device network* (ETH 3 or ETH 4) of the 140CRP31200 module.
8	Connect DRSs to the main ring for distributed I/O sub-rings and/or distributed I/O clouds. Refer to the Predefined Configuration Files topic in the *Quantum EIO System Planning Guide* for details on installing DRSs.

Example

The following graphic shows an extended distributed I/O network (4). The 140NOC78000 distributed I/O head module is interlinked with the *extend port* of the 140NOC78100 control head module. The 140NOC78000 module and the 140NOC78100 module are also interlinked with the 140CRP31200 remote I/O head module. Thus, the extended distributed I/O network participates in the Quantum EIO device network.

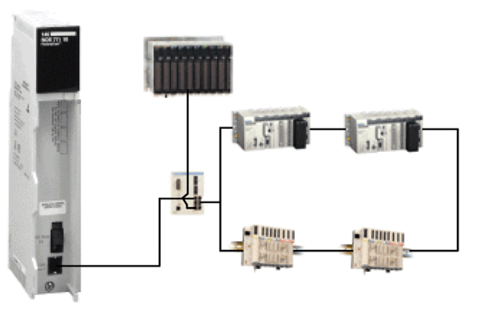


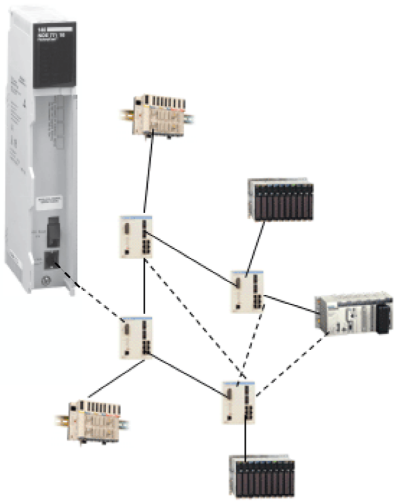
1 140CRP31200 remote I/O head module on the local rack
2 140NOC78000 distributed I/O head module (interlinked with the *extend port* of the 140NOC78100 module to support the extended distributed I/O network and interlinked with the 140CRP31200 module to support the device network)
3 140NOC78100 control head module (interlinked with the 140NOC78000 module to provide transparency between the control network and the device network)
4 extended distributed I/O network, which communicates with the Quantum EIO system
5 distributed I/O sub-ring
6 DRSs — with C4 predefined configuration files — for copper-to-fiber and fiber-to-copper transitions on the main ring

7 distributed I/O cloud
8 remote I/O drop on the main ring
9 remote I/O drops on the remote I/O sub-ring
10 DRS with a C1 predefined configuration file on the main ring connected to the remote I/O sub-ring
11 control network

Preferred Distributed I/O Network Topologies

Distributed I/O Network Topologies

Topology Type	Definition	Example
star	In a star topology, all single-port Ethernet devices are connected through an intermediate device, such as a DRS.	
ring (loop)	In a ring topology (also known as a daisy chain loop), dual-port Ethernet devices that support RSTP are connected in a ring. With a ring topology, network redundancy is achieved. **NOTE:** Single-port Ethernet devices can connect to a ring via a ConneXium extended managed switch (not required to be configured as a DRS) that is part of the ring. The graphic to the right displays this topology.	

Topology Type	Definition	Example
mesh	In a mesh topology, single-port Ethernet devices are connected to each other through intermediate devices, such as a ConneXium extended managed switch (not required to be configured as a DRS). With a mesh topology, network redundancy is possible.	

Chapter 4
Configuring the 140 NOC 780 00 Module

Introduction

This chapter shows you how to use Control Expert programming software to select and configure the 140 NOC 780 00 head module on the local rack.

NOTE: The instructions presented in this chapter include specific choices made for a sample project. Your Control Expert project may include different choices that are appropriate for your specific configuration.

What Is in This Chapter?

This chapter contains the following sections:

Section 4.1
Creating a Project in Control Expert

Overview

This section shows you how to add modules, including the 140NOC78000 distributed I/O head module, to your project, using Control Expert.

What Is in This Section?

This section contains the following topics:

Creating a Project in Control Expert

Introduction

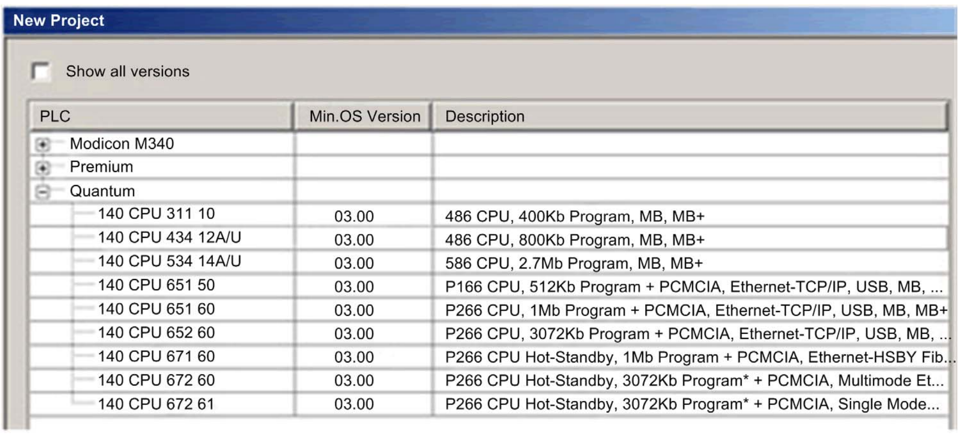
You may have already created a project in Control Expert and installed a power supply and a 140CRP31200 remote I/O head module. If so, jump to the Adding a 140NOC78000 Distributed I/O Head Module... topic *(see page 74)* If not, the following pages show you how to create a new Control Expert project and add the following components:

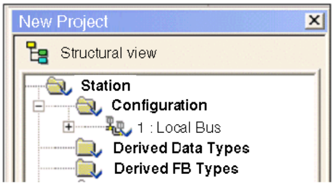
- a CPU
- a power supply
- a 140NOC78000 distributed I/O head module

NOTE: To add the power supply and a 140CRP31200 module to the local rack in Control Expert, refer to the *Quantum EIO Remote I/O Modules Installation and Configuration Guide*.

Creating and Saving a New Control Expert Project

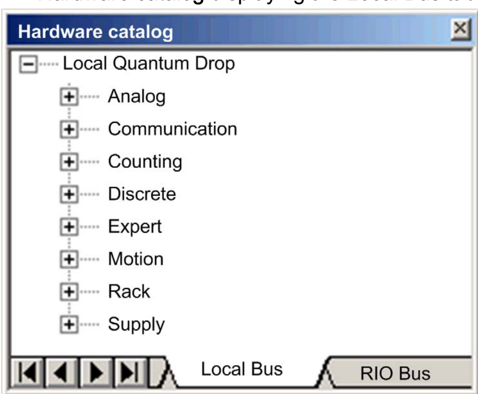
The following steps describe the creation of a project:

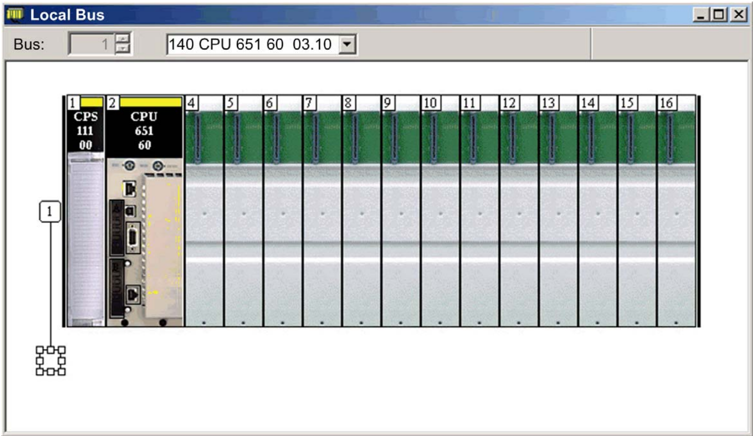
Step	Action
1	Open Control Expert.
2	In the Control Expert main menu, select **File → New...**. The **New Project** window opens displaying a list of Schneider Electric controller types.
3	In the **New Project** window, expand the Quantum node and select a CPU. In this example, select the 140 CPU 651 60 controller: 

Step	Action
4	Click **OK**. Control Expert displays the **Project Browser**, below. 
5	To save the project, select **File → Save**. The **Save As** dialog opens.
6	In the **Save As** dialog, type in a **File name** — which will be the name of your Control Expert project — then click **Save**. Control Expert saves your project to the specified path location. **NOTE:** You can change the default location Control Expert uses to store project files. Before saving your project: 1 Select **Tools → Options**. The **Options Management** window opens. 2 In the left pane, navigate to **Options → General → Paths**. 3 In the right pane, type in a new path location for the **Project path**. You can also edit the: ❍ Import/Export file path ❍ XVM path ❍ Project settings templates path 4 Click **OK** to close the window and save your path edits.

Adding a Power Supply to the New Control Expert Project

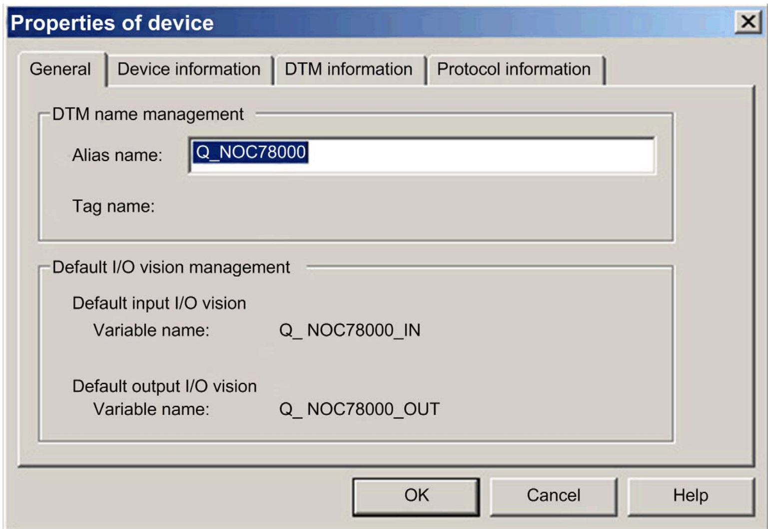
The next step is to add a power supply to your Control Expert project:

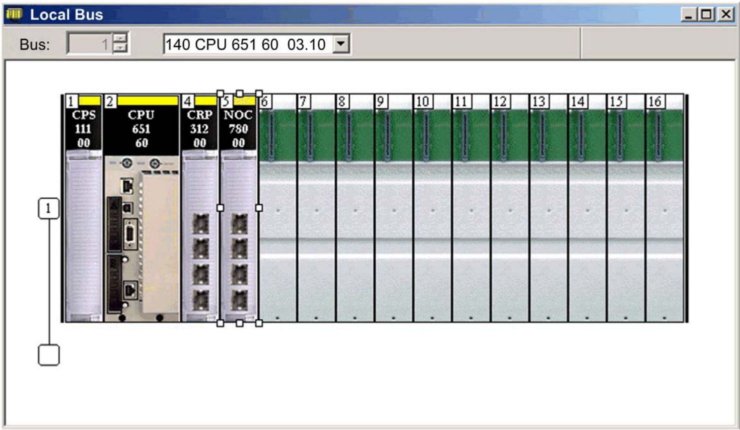
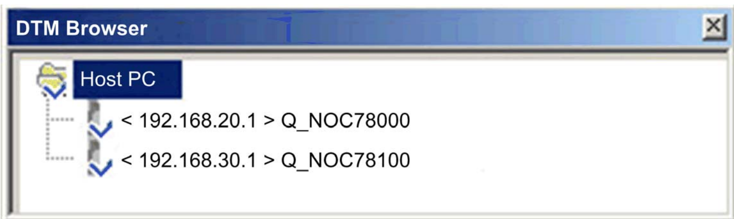
Step	Action
7	In the **Project Browser**, double click **Local Bus**. Control Expert displays both the: ● **Local Bus** window with the selected CPU in the second position, and ● **Hardware catalog** displaying the **Local Bus** tab, below: 

Step	Action
8	In the **Hardware catalog**, under the **Supply** node, use your mouse to select then drag a 140 CPS 111 00 power supply to a position in the rack — in this example, slot 1. 
9	In the **File** menu, select **Save**, to save your edits. **NOTE:** Schneider Electric recommends that you periodically save your changes as you make edits.

Adding a 140NOC78000 Distributed I/O Head Module to the New Control Expert Project

Next, add a 140NOC78000 distributed I/O head module to your project:

Step	Action
10	Returning to the **Hardware catalog**, under the **Communication** node, use your mouse to select then drag a 140NOC78000 control head module to an open slot in the rack — in this example, slot 5. When you drop the module into the rack, Control Expert opens the communication module **Properties** window.
11	In the **General** tab of the module properties window, Control Expert assigns a the device an alias name, in this example, Q_NOC78000. You can type in a name to change the default alias name.  When you change the alias name, Control Expert changes the base input and output type and variable names to match the edited alias name. **NOTE:** ● Schneider Electric recommends that you assign a unique alias name to each communication module. This practice helps you distinguish between modules of the same type. ● No other configuration needs to be, or can be, performed in the tabs of this window. All other pages are read-only.
12	In the **File** menu, select **Save** to save your edits.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>13</td><td>Click OK to close the Properties window. The Local Bus now displays the modules you have added:

</td></tr>
<tr><td>14</td><td>The next step is to configure the located memory space in the CPU for the 140NOC78000 module’s inputs and outputs (see page 76).</td></tr>
<tr><td>15</td><td>Open the DTM Browser in Control Expert to configure the DTM properties of the 140NOC78000 module by clicking Tools → DTM Browser.

</td></tr>
</table>

Configuring the Size and Location of Inputs and Outputs

Overview

Use the **Configuration** tab of the 140NOC78000 distributed I/O head module's **Properties** window to configure the:

- size and starting position of inputs
- size and starting position of outputs

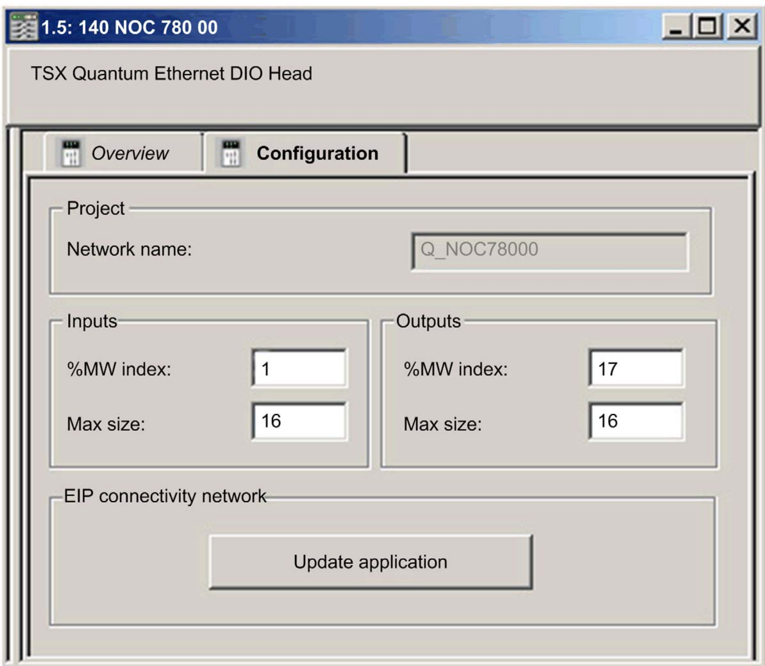
The following steps present one example of how to configure the size and location of inputs and outputs. Your own project configuration may differ.

Setting Input and Output Memory Addresses and Naming the Module

The **Properties** window opens when you double-click the left mouse button on the image of the 140NOC78000 module in either the **Local Bus** window or the **Project Browser**.

When you select the **Configuration** tab, it displays the network, or **Alias**, name. This is the name assigned to the network channel when you added the 140NOC78000 module to the project.

Use the **Configuration** page to edit the communication module inputs and outputs, as follows:



To input the above settings, perform the following steps:

Step	Action
1	In the module's **Properties** window, select the **Configuration** tab.
2	Type in the size and starting position of the inputs and outputs, as follows:
	In the **Inputs** area: • In the **%MW index** field, type in a starting address for inputs, in this example: **1**. • In the **Max size** field, type in the maximum number of 16-bit words dedicated to inputs, in this example:**16**.
	In the **Outputs** area: • In the **%MW index** field, type in a starting address for outputs, in this example: **17**. • In the **Max size** field, type in the maximum number of 16-bit words dedicated to outputs, in this example: **16**.

	Notes:
	• The inputs and outputs can be located at any available address and do not need to be located in adjacent areas. Confirm that the space allocated to inputs and outputs does not overlap.
	• Control Expert automatically reserves space for two arrays of 32 bytes, as follows:
	  ○ for connection health bits, located at the beginning of the space configured for inputs
	  ○ for connection control bits, located at the beginning of the space configured for outputs
	• The specified %MW range for both inputs and outputs are available in the CPU. For more information, refer to the **Processor Configuration Screen** topic in the Control Expert help file.

3	In Control Expert select **Edit** → **Validate** (or click the **Validate** toolbar button) to save the address and size settings for inputs and outputs. **NOTE:** After you validate module settings for the first time, you cannot edit the module name. If you subsequently decide to change the module name, delete the existing module from the configuration, then add and rename a replacement module.

Completing the Ethernet Network Configuration

After configuring settings for inputs and outputs, the next step is to configure the 140NOC78000 module settings, beginning with its Channel Properties *(see page 112)*, and then configure remote Ethernet network devices.

NOTE: After you input configuration settings for the 140NOC78000 module and remote devices, return to the **Configuration** tab of the 140NOC78000 module's **Properties** window and click the **Update application** button. This creates derived data type (DDT) variables *(see page 204)* that display the following information and commands for your Control Expert project:

- connection health bits, that display the status of each connection
- connection control bits, you can use to toggle each connection on and off
- the value of input and output items
- module and device configuration settings
- free memory space that has been reserved, but not yet allocated

Section 4.2
The Control Expert FDT/DTM Interface

Overview

The section describes the use of DTMs within Control Expert.

What Is in This Section?

This section contains the following topics:

Ethernet Configuration Tool User Interface

Overview

The Ethernet Configuration Tool presents the following two views:
- a **Device Editor** for configuring Ethernet communication modules, remote devices, and their common Ethernet connections
- a **Diagnostic window** for monitoring the real-time operation of network devices, and diagnosing their condition

Connecting and Disconnecting a Device or Module DTM

A device or module DTM can be either connected to, or disconnected from the physical device or module.

When a device and its DTM are...	You can use the Ethernet configuration tool to...
Connected	Monitor and diagnose the real-time operation of the device or module
Disconnected	Configure a communication module or remote device by editing its properties

NOTE: Be sure to distinguish between:
- connecting and disconnecting a DTM and the associated physical device using commands in the **DTM Browser**, and
- placing Control Expert in online or offline operating mode using commands in the Control Expert **PLC** menu

You can connect a DTM to, or disconnect a DTM from a device or module using the contextual pop-up menu in the **DTM Browser**. The **DTM Browser** indicates the relationship between the DTM and the remote module or device: a connected DTM is displayed in **bold** text; a disconnected DTM is displayed in normal text.

To connect a DTM to, or disconnect a DTM from its respective module or device, follow these steps:

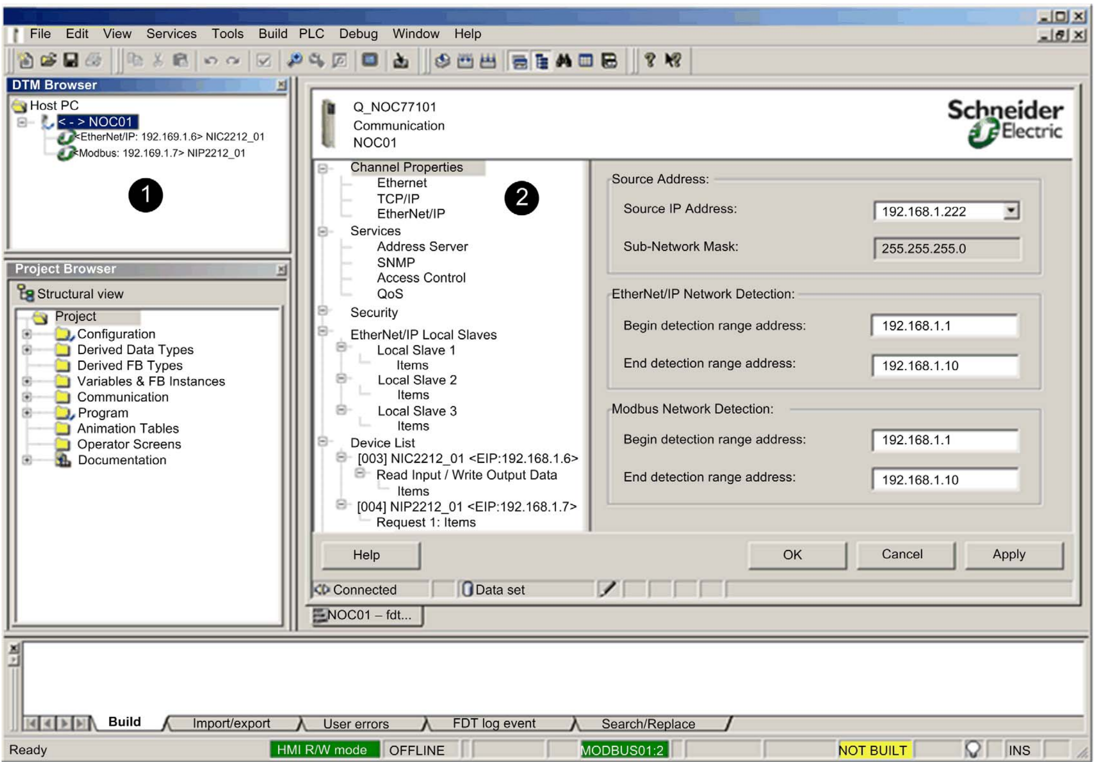
Step	Action
1	In the **DTM Browser** select the DTM that you want to connect to, or disconnect from, the physical communication module or remote device. **NOTE:** If the module or device name appears in: ● **bold** text, it is connected and only the **Disconnect** command is enabled ● normal text, it is disconnected and only the **Connect** command is enabled
2	Click the right mouse button. A pop-up menu opens.
3	Select one of the following commands: ● **Connect** ● **Disconnect** **NOTE:** The **Connect** and **Disconnect** commands are also available in the Control Expert **Edit** menu.

Device Editor

Use the **Device Editor** to display and configure device properties. The collection of properties you can view or edit depends upon the device selected in the **DTM Browser**, and whether Control Expert is operating in **Advanced Mode**.

When the communication module and its DTM are...	The Device Editor opens in this mode...
Connected	read / write
Disconnected	read-only

The **Device Editor** looks like this:



1 DTM Browser
2 Device Editor

Refer to the **Device Editor** topic in this help file for information on how to use the editor.

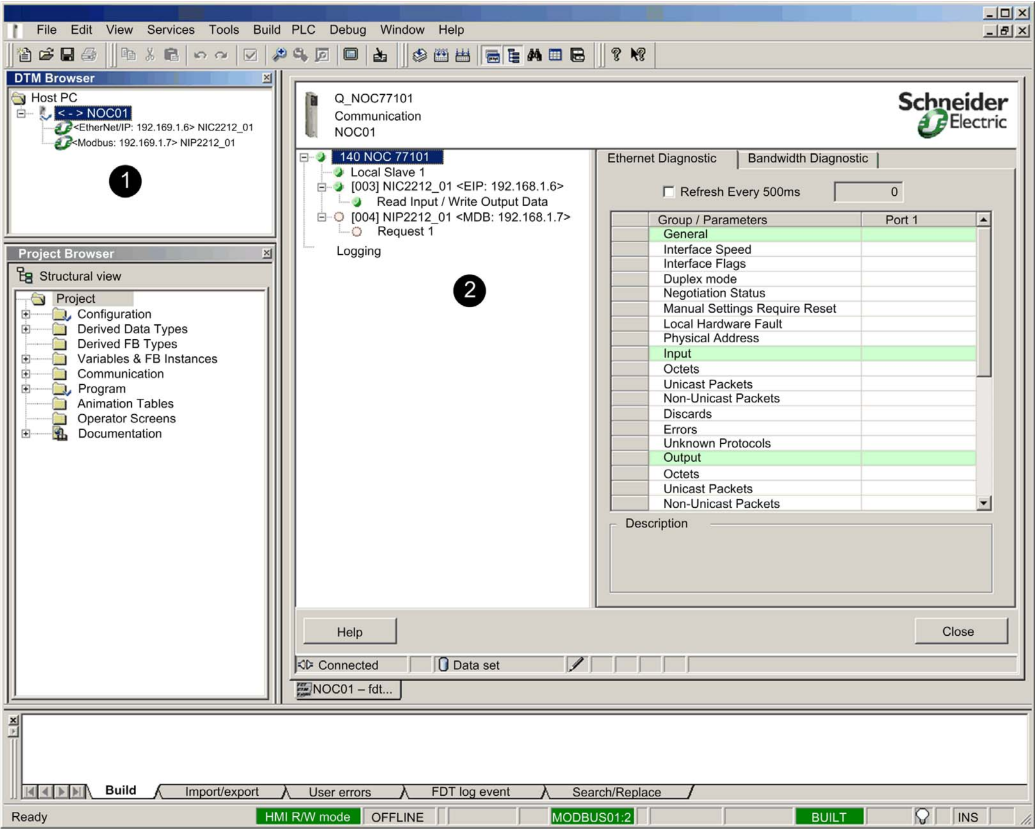
Access the **Device Editor** from the **DTM Browser**. If necessary, you may need to first disconnect the Ethernet communication module from its DTM.

Step	Action
1	(if necessary) In the **DTM Browser**, select the Ethernet communication module node and click the right mouse button. then select **Disconnect** in the pop-up menu.
2	In the **DTM Browser**, again select the Ethernet communication module node and click the right mouse button. The same pop-up menu opens.
3	Select **Device menu → Configuration** in the pop-up menu. The **Device Editor** opens.

Diagnostic Window

Use the **Diagnostic Window** to display:
- colored LED icons that indicate the operating status of the Ethernet communication module, remote devices, and their connections
- diagnostic data for the communication module, local slaves, and Ethernet connections

The **Diagnostic Window** can be displayed only when the communication module is connected to its DTM.



1 DTM Browser
2 Diagnostic Window

Refer to the **Diagnostic Window** topic in this help file for information on how to use this window.

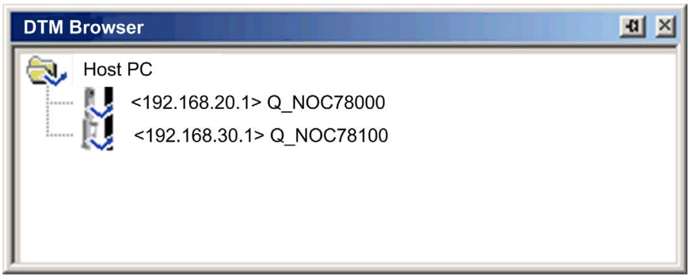
Access the **Diagnostic Window** from the **DTM Browser**. If necessary, you may need to first connect the Ethernet communication module to its DTM.

Step	Action
1	(if necessary) In the **DTM Browser**, select the Ethernet communication module node and click the right mouse button, then select **Connect** in the pop-up menu.
2	In the **DTM Browser**, again select the Ethernet communication module node and click the right mouse button. The same pop-up menu opens.
3	Select **Device menu** → **Diagnostics** in the pop-up menu. The **Diagnostic Window** opens.

DTM Browser

Overview

The **DTM Browser** displays a hierarchical list of DTMs in the form of nodes on a connectivity tree that have been added to your Control Expert project. Each DTM node represents an actual module or device in your Ethernet network.



Node Types

There are 3 types of DTM nodes:

- communication DTMs:
  - Any communication DTM can be plugged directly under the root node (host PC) and is at the 1st level.
  - A communication DTM can support gateway DTMs or device DTMs as children if their protocols are compatible.
- gateway DTMs:
  - A gateway DTM can support other gateway DTMs or device DTMs as children if their protocols are compatible.
- device DTMs:
  - A device DTM does not support any child DTMs.

Node Names

Each DTM has a default name when inserted into the browser. The default name consists of the following elements:

<channel: address> device name

Where:

Element	Description
channel	This is the name of the channel communication media, to which the device is plugged in. This name is read from the DTM and is set by the device vendor. **Example**: EtherNet/IP, Modbus
address	The bus address of the device, which can be: ● the connection point on its parent gateway network ● the slot number in the modular device parent internal bus **Example**: the device IP address
device name	The default name is determined by the vendor in the device DTM, but can be edited by the user.

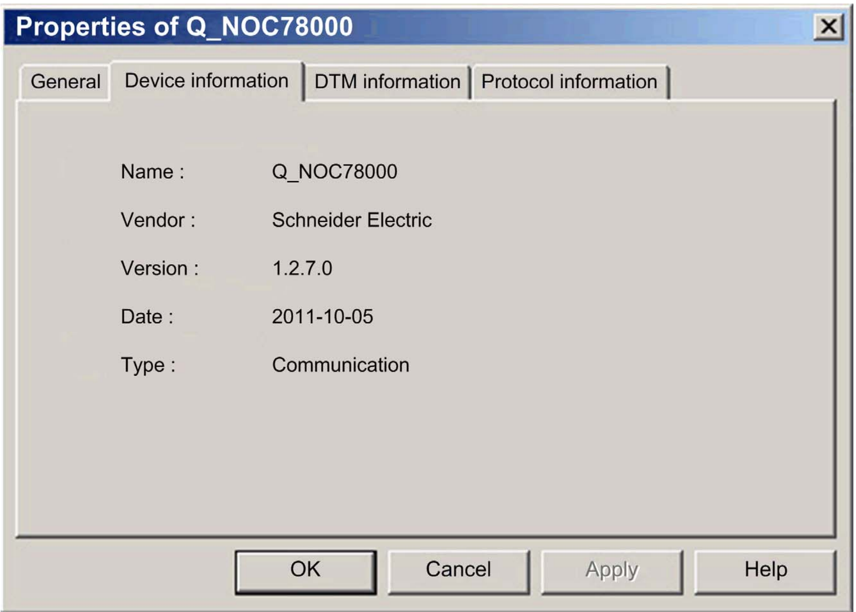
Node Status

The **DTM Browser** displays the status of each DTM node in the connectivity tree, as follows:.

Status	Description
Built / Not-built	A blue check mark ✓ superimposed on a device icon indicates that node, or one of its sub-nodes, is not built. This means that some property of the node has changed, and the information stored in the physical device is no longer consistent with the local project.
Connected / Disconnected	A connected DTM is denoted in **bold** text. An unconnected DTM appears in plain text. **NOTE:** ● Connecting a DTM to its physical device automatically connects all higher level parent nodes up to the root node. ● Disconnecting a DTM from its physical device automatically disconnects all its lower level child nodes. **NOTE:** Connecting or disconnecting a DTM to or from its device does not also connect or disconnect Control Expert to or from the PLC. DTMs can be connected/disconnected while Control Expert is either offline or online.
Installed / Not-installed	A red X superimposed on a device icon indicates the DTM for that device is not installed on the PC.

Handling Invalid Nodes

As indicated above, a red X superimposed on a node indicates the DTM for that node is not installed on the PC. To resolve this situation, right-click the node to open a pop-up menu with the following 2 commands:

Command	Description
Delete	Removes the selected node (and its sub-nodes) from the **DTM Browser**.
Properties	Opens the following dialog, which you can use to identify the name of the missing DTM: 

NOTE: After you install the DTM, reopen the Control Expert application.

DTM Browser Menu Commands

Overview

The **DTM Browser** includes a pop-up, contextual (right-click) menu that displays commands for the currently selected DTM. The list of available commands consists of:

- universal commands, as determined by the selected node level:
  - host PC node (level 1)
  - communication module node (level 2)
  - remote device node (level 3)
- device-specific commands, as determined by the device DTM

Host PC Node Commands

The **Host PC** node contextual menu includes the following commands:

Name	Description
Add[1]	Opens the **Add** dialog — containing a subset of the **Hardware Catalog**, allowing the selection of a communication module DTM.
Check DTM devices[1]	Checks the current project for invalid DTMs or DTMs that are not installed in the PC. If the results of the check include invalid or not-installed DTMs, they are displayed in the **User errors** tab in the information window and a red X is superimposed over their icons in the **DTM Browser**.
DTM services	Displays the communication DTMs selection, as well as the device topology, their respective IP addresses, and connection state. In this dialog, for each device you can connect, disconnect, load from devices, or store to devices. You can also choose to stop communication or continue activity when detected errors occur.
DTM hardware catalog	Displays the **DTM catalog** tab of the **Hardware Catalog** dialog.
Expand all[2]	Displays every DTM in the project.
Collapse all[2]	Displays only the communication DTMs in the project.
1. This command also appears in the Control Expert **Edit** menu. 2. This command also appears in the Control Expert **View** menu.	

Communication Module and Remote Device Node Commands

The **DTM Browser**'s contextual menu has the following items:

Name	Description
Open[1]	This opens the **Device Editor** for the selected communication module. **NOTE:** Double-clicking the left mouse button on the DTM in the **DTM Browser** also opens this window.
Add [1]	This opens the **Add** dialog, displaying a subset of the **Hardware Catalog**, allowing the selection of a DTM. **NOTE:** Control Expert filters the content of the **Add** dialog, so that it displays only DTMs that are compatible with the selected DTM selected.
Delete[1]	If the selected DTM allows this function, this deletes the selected DTM and its sub-node DTMs from the DTM connectivity tree. Deletion from the DTM connectivity tree does not affect the DTM's link to the I/O scanning table.
Field Bus Discovery	This scans the connected physical devices to create the corresponding field bus topology. Refer to the Field Bus Discovery Service topic.
Connect[1]	This connects the DTM *(see page 90)* to its physical device on the network. This connection does not depend on the PLC online/offline status of the Control Expert project application. **NOTE:** Connecting a gateway or device DTM implicitly connects its parent DTM.
Disconnect[1]	This disconnects the DTM *(see page 90)* from its physical device. This disconnection depends on the PLC online/offline status of the Control Expert project application. **NOTE:** Disconnecting a gateway or device DTM implicitly disconnects its parent DTM.
Load data from device[1]	This loads data from the physical device on the network to the DTM.
Store data to device[1]	This loads data from the DTM to the physical device on the network.
Copy	This command is disabled.
Paste	This command is disabled.
Device menu	This command opens a sub-menu that contains device-specific commands, as determined by the device vendor. For details, refer to the Communication Module Commands topic *(see page 88)*.
Device menu 2	This command opens a sub-menu that contains device-specific commands, as determined by the device vendor. For details, refer to the Communication Module Commands topic *(see page 88)*.
Properties[1]	Opens the Ethernet communication module **Properties** window.
1. This command also appears in the Control Expert **Edit** menu. 2. This command also appears in the Control Expert **View** menu.	

Name	Description
Print device[1]	If this optional function is supported by a DTM, this function displays the device documentation — including configuration settings — in the PC's default Internet browser, which can then be printed. **NOTE:** Device information can be printed: ● for only one device DTM at a time, when that DTM is not open for editing in the **Device Editor**. ● only when the DTM is disconnected from the physical device.
Zoom out[2]	This returns to the display of the entire DTM connectivity tree.
Expand all[2]	This displays DTMs below the selected DTM.
Collapse all[2]	This displays only the selected DTM.
1. This command also appears in the Control Expert **Edit** menu. 2. This command also appears in the Control Expert **View** menu.	

Communication Module Commands

When you select **Device menu** in the main contextual menu for the communication module, a sub-menu with the following commands is displayed:

Name	Description
Offline Parameter	This command is disabled.
Online Parameter	This command is disabled.
Compare	This compares 2 devices, either online or offline.
Configuration	This opens the **Device Editor** for the selected communication module, when the module and its DTM are disconnected.
Observe	This command is disabled.
Diagnosis	This opens the **Diagnosis Window** for the selected communication module, when the module and its DTM are connected.

Name		Description
Additional functions	Add EDS to library	Opens the **EDS File Wizard**, which you can use to add a device EDS file to the Control Expert EDS device library. Control Expert displays the contents of EDS files as DTMs for use in the **DTM Browser** and **Device Editor**.
	Remove EDS from library	Opens the **EDS Deletion from Device Library** window, which you can use to delete an EDS file from the device library.
	Online Action	Opens the **Online Action** window. Depending upon the protocol(s) a remote device supports, you can use the **Online Action** window to: ● Ping a remote EtherNet/IP or Modbus TCP device ● view and write to EtherNet/IP properties in a remote EtherNet/IP device ● view and write to port configuration properties in a remote EtherNet/IP device
	EtherNet/IP Explicit Message	Opens the EtherNet/IP Explicit Message *(see page 241)* window, which you can use to send explicit messages to EtherNet/IP remote devices.
	Modbus TCP Explicit Message	Opens the Modbus TCP Explicit Message *(see page 244)* window, which you can use to send explicit messages to Modbus TCP remote devices.
	About	
	Advanced Mode	Displays or hides expert-level properties that help define Ethernet connections. See the Enabling Advanced Mode topic *(see page 91)* for instruction on how to use this feature.

When you select **Device menu 2** in the main contextual menu for the communication module, a sub-menu with the following commands is displayed:

Name	Description
Configuration	This opens the **Device Editor** for the selected communication module, when the module and its DTM are disconnected.
Diagnosis	This opens the **Diagnosis Window** for the selected communication module, when the module and its DTM are connected.
Add EDS to library	Opens the **EDS File Wizard**, which you can use to add a device EDS file to the Control Expert EDS device library. Control Expert displays the contents of EDS files as DTMs for use in the **DTM Browser** and **Device Editor**.
Remove EDS from library	Opens the **EDS Deletion from Device Library** window, which you can use to delete an EDS file from the device library.
Online Action	Opens the **Online Action** window. Depending upon the protocol(s) a remote device supports, you can use the **Online Action** window to: ● Ping a remote EtherNet/IP or Modbus TCP device ● view and write to EtherNet/IP properties in a remote EtherNet/IP device ● view and write to port configuration properties in a remote EtherNet/IP device

Name	Description
EtherNet/IP Explicit Message	Opens the EtherNet/IP Explicit Message *(see page 241)* window, which you can use to send explicit messages to EtherNet/IP remote devices.
Modbus TCP Explicit Message	Opens the Modbus TCP Explicit Message *(see page 244)* window, which you can use to send explicit messages to Modbus TCP remote devices.
Advanced Mode	Displays or hides expert-level properties that help define Ethernet connections. See the Enabling Advanced Mode topic *(see page 91)* for instruction on how to use this feature.

Connecting and Disconnecting a Device or Module DTM

A device or module DTM can be either connected to, or disconnected from, the physical device or module.

When a device and its DTM are...	You can use the Ethernet configuration tool to...
Connected	Monitor and diagnose the real-time operation of the device or module
Disconnected	Configure a communication module or remote device by editing its properties

NOTE: Distinguish between:
- connecting and disconnecting a DTM and the associated physical device using commands in the **DTM Browser**
  — and —
- placing Control Expert in online or offline operating mode using commands in the Control Expert **PLC** menu

You can connect a DTM to, or disconnect a DTM from a device or module using the contextual pop-up menu in the **DTM Browser**. The **DTM Browser** indicates the relationship between the DTM and the remote module or device: a connected DTM is displayed in **bold** text; a disconnected DTM is displayed in normal text.

To connect a DTM to, or disconnect a DTM from its respective module or device, follow these steps:

Step	Action
1	In the **DTM Browser** select the DTM that you want to connect to, or disconnect from, the physical communication module or remote device. **NOTE:** If the module or device name appears in: ● **bold** text, it is connected and only the **Disconnect** command is enabled. ● normal text, it is disconnected and only the **Connect** command is enabled.
2	Click the right-mouse button. **Result**: A pop-up menu opens.
3	Select one of the following commands: ● **Connect** ● **Disconnect** **NOTE:** The **Connect** and **Disconnect** commands are also available in the Control Expert **Edit** menu.

Enabling Advanced Mode

Use the contextual menu in the **DTM Browser** to toggle Control Expert in or out of **Advanced Mode**, thereby displaying or hiding expert-level properties that help define Ethernet connections. These properties are identified by the icon.

NOTE: To maintain system performance, confirm that **Advanced Mode** properties are configured only by persons with a solid understanding of communication protocols.

To toggle **Advanced Mode** on and off:

Step	Action
1	Close both the **Diagnosis Window** and every instance of the **Device Editor** before attempting to toggle **Advanced Mode** on or off. **NOTE:** If the **Device Editor** or the **Diagnosis Window** is open, the **Advanced Mode** status — on or off — cannot be changed.
2	In the **DTM Browser**, right-click the communication module. **Result**: A pop-up menu opens.
3	To toggle ON advanced mode, select **Device Menu → Advanced Mode**.
4	To toggle OFF advanced mode, repeat steps 1 through 3, above.

Field Bus Discovery Service

Introduction

Use the field bus discovery service to detect, and add to your Control Expert application, network devices that are situated on a local channel. The field bus discovery service is available only when the Ethernet communication module DTM is connected to its physical device.

Only the first level devices below the communication DTM are detected.

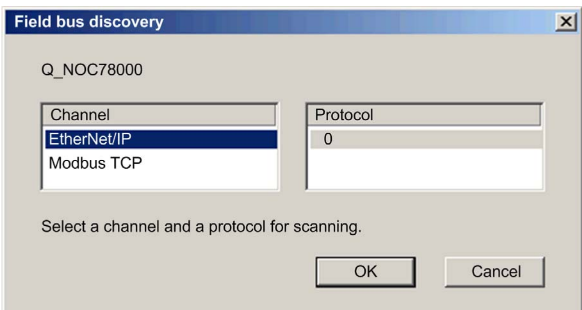
Performing Field Bus Discovery

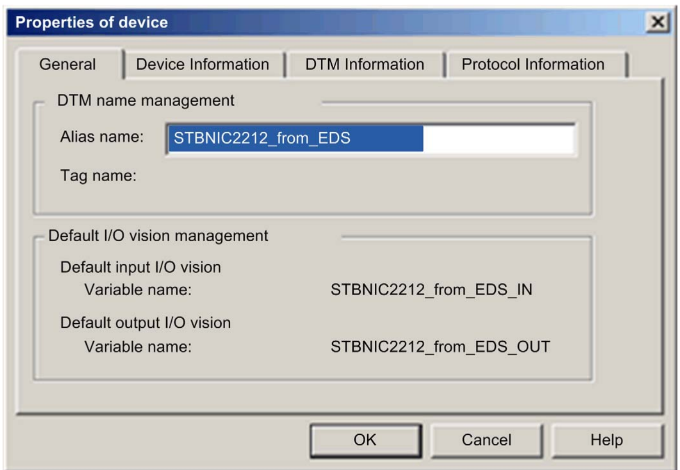
The results of the scanning process is compared to the registered DTMs in the DTM catalog of the computer. If a match is found in the DTM catalog for a scanned device, the results are accompanied with a matching type that gives the accuracy of the match.

The three available matching types are:

- exact match:
  All identification attributes are matching. The correct device type was found.
- generic match:
  At least the **Vendor** and device **Type ID** attributes match. The support level of DTM is `Generic Support`.
- uncertain match:
  At least the **Vendor** and device **Type ID** attributes match. The support level of DTM is **not** `Generic Support`.

The following procedure explains how to use the field bus discovery service:

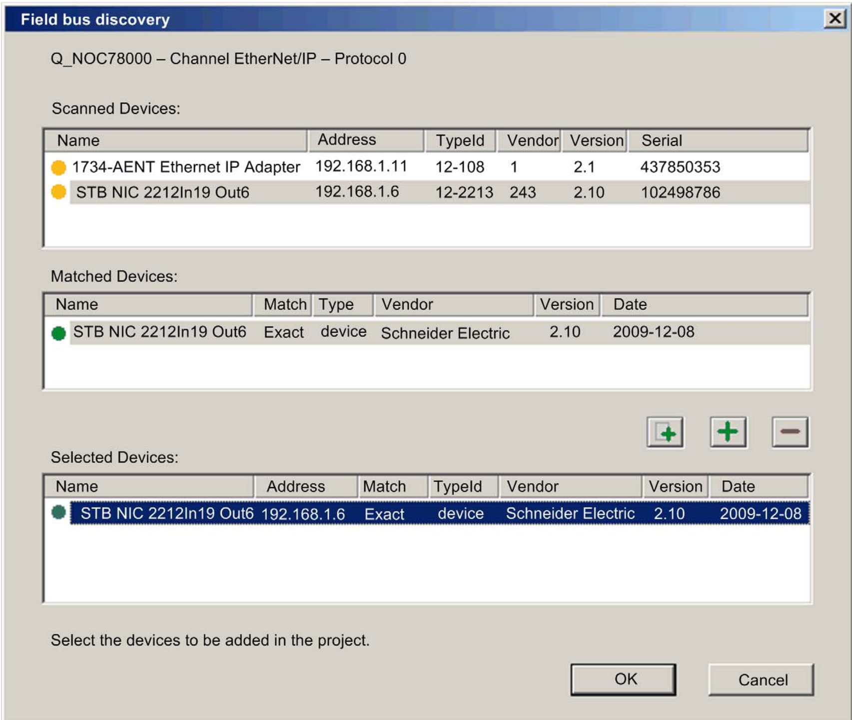
Step	Action
1	In the **DTM Browser**, select an appropriate DTM.
2	Right click, and in the pop-up menu and select **Field bus discovery**. **Result**: The **Field bus discovery** dialog opens. 
3	If necessary, select a channel and a protocol: • if the DTM has more than one channel • if the channel supports more than one protocol

Step	Action
4	Click **OK**. **Result**:The service starts to detect devices on the selected channel. **NOTE:** The field bus discovery service limits its search to only the range of IP addresses that is pre-configured for the selected channel in the **Channel Properties** page *(see page 113)*.
5	If at least one matched device has been found, the **Field bus discovery** dialog displays a list of **Scanned Devices**.
6	Use the controls of the **Field bus discovery** dialog to select the devices to add to your Control Expert application.
7	After you have selected all the devices you want to add in the **Field bus discovery** dialog, click **OK**.
8	If the field bus discovery process has found one or more devices with an IP address that is already in use in the project, you will be asked if you want to continue and replace the existing project device(s). ● Click **Yes** and proceed to step 9. — or — ● Click **No** to cancel automatic field bus discovery.
9	The device properties dialog opens, displaying the default name for the first discovered device to be added:  In the **General** page of the device properties dialog, type in the **Alias name** for the device to be added, then click **OK**. **Result**: The dialog closes, then re-opens if there is another device to be added to the application.
10	Repeat step 9 for each additional discovered device.

Step	Action
11	After all devices have been added to the application, configure each device for operation as part of the application. To do this: ● Disconnect the Ethernet communication module from its DTM. In the **DTM Browser**, select the Ethernet communication module, then select **Edit** → **Disconnect**. ● Configure the new device properties in the DTMs for both the Ethernet communication module, and the newly added remote device.

Field Bus Discovery Dialog

If at least one matched device has been found, the **Field bus discovery** dialog box is displayed listing the scanned and matched devices. Select the matched devices to be created in the Control Expert project (which then shows up in the **Selected Devices** list:



This dialog presents 3 lists:

This list...	Displays...
Scanned Devices	All the devices (matched and unmatched) found during the scan.
Matched Devices	The matched DTMs found in the workstation DTM catalogue for the device that you selected in the **Scanned Devices** list. Each time a scanned device is selected in the **Scanned Devices** list, the contents of the **Matched Devices** list is updated to display the matched device DTMs found for the selected scanned device. The matching process can yield one or more matched devices for a given scanned device. In this case, only one DTM was discovered for the selected scanned device.
Selected Devices	This list displays the device DTMs that have been selected in the **Matched Devices** list, which will be added to the Control Expert project.

The lists use the following colored icons:

<table>
<tr><th>This color...</th><th>Indicates...</th></tr>
<tr><td>green</td><td>The device has been selected.</td></tr>
<tr><td>yellow</td><td>The device has been matched.</td></tr>
<tr><td>red</td><td>The device has not been matched.</td></tr>
<tr><td>black</td><td>Information about the address of the scanned device:
In the Scanned Devices list, the device has an address identical to one of the DTMs in the Control Expert project.
In the Matched Devices list, the device will be assigned an address identical to one of the DTMs in the Control Expert project.</td></tr>
<tr><td colspan="2">NOTE: An icon can consist of 2 colors. For example, a search can discover a device that:
has a matching DTM
— and —
has an IP address identical to a device already added to the Control Expert application
In this case, the icon next to the discovered device would be:
half yellow and half black, before it is selected
— and —
half green and half black, after it is selected</td></tr>
</table>

This dialog has 5 buttons:

Button	Use this button to...
Add All 	Automatically add the best matched (according to the matching types listed above) device DTM for each found device in the **Matched Devices** list to the **Selected Devices** list.
Add One 	Add the matched device DTM selected in the **Matched Devices** list.
Remove 	Remove one or more devices from the **Selected Devices** list.
OK	Insert the device DTMs in the **Selected Devices** list into the Control Expert project. If there are one or more devices in the **Selected Devices** list that have the same address in the Control Expert project, a message box opens asking if you want to continue. If you click **OK**, all devices in the Control Expert project that have identical addresses as the selected devices are **deleted** and **replaced** by the DTMs selected in the **Selected Devices** list.
Cancel	Cancel the field bus discovery scan and do nothing. All information in the three lists is discarded.

Device Editor

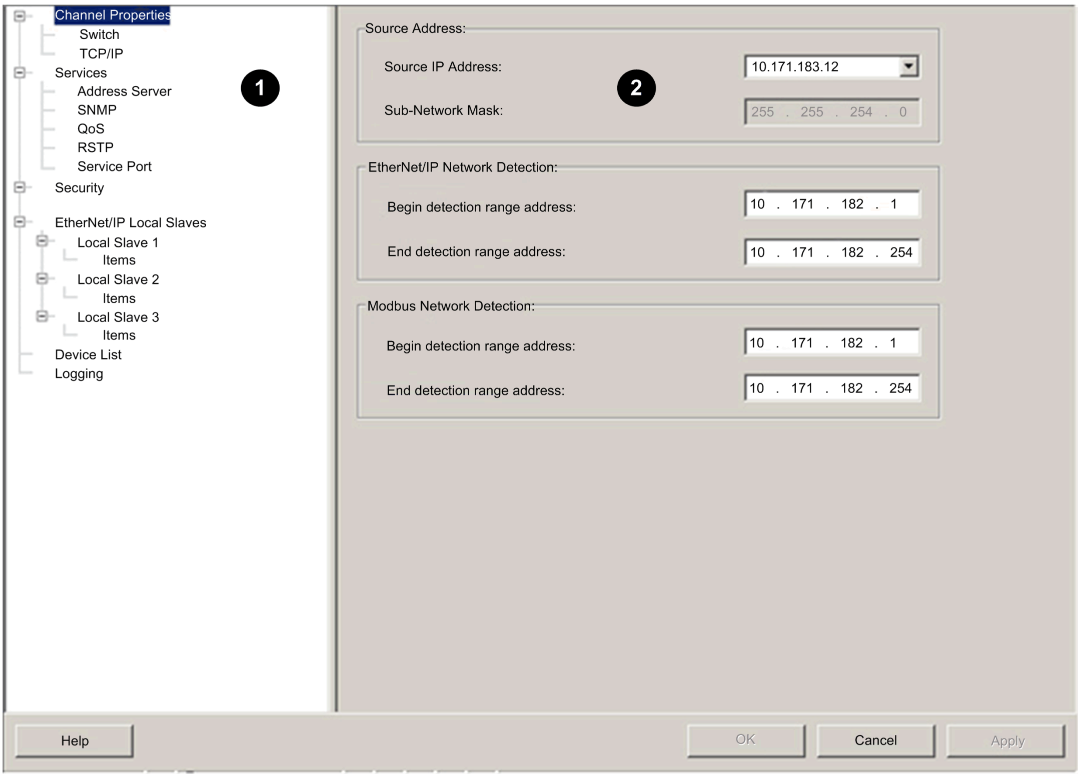
Description

Use the **Device Editor** to view and configure Ethernet communication modules and remote devices. The collection of properties you can view or configure depends on:

- the node type selected in the **DTM Browser**:
  - communication module
  - remote device
- whether Control Expert is operating in **Advanced Mode**

Displaying Properties of the Ethernet Communication Module

After you open the Ethernet communication module in the **DTM Browser**, the left pane (1) of the **Device Editor** displays a tree control containing configurable property groups for the communication module. Click a node in the tree control to display one or more pages of module properties for the selected group in the right pane (2).



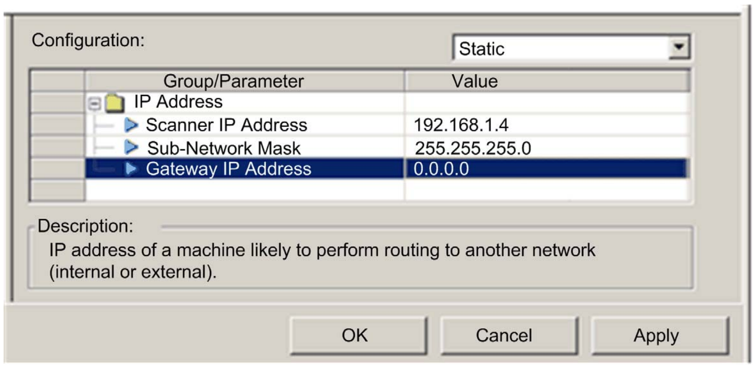
Property Types

The **Device Editor** displays an icon next to many device properties. The following 3 icons are displayed:

This icon...	Indicates the property is...
	Read-only The property value cannot be edited in this page.
	Read-write The property value can be edited in this page.
	An expert-level communication protocol property that is displayed only when **Advanced Mode** is enabled.

Displaying Property Definitions

Many property configuration pages provide an on-screen definition of the property you are editing. To display a property definition in the **Description** section of the page, select that property in the property list. The following screen displays a description of the **Gateway IP Address** property.



NOTE: The preceding displayed page can be accessed by opening an Ethernet communication module in the **Device Editor**, and then selecting **Channel Properties** → **TCP/IP** in the navigation tree.

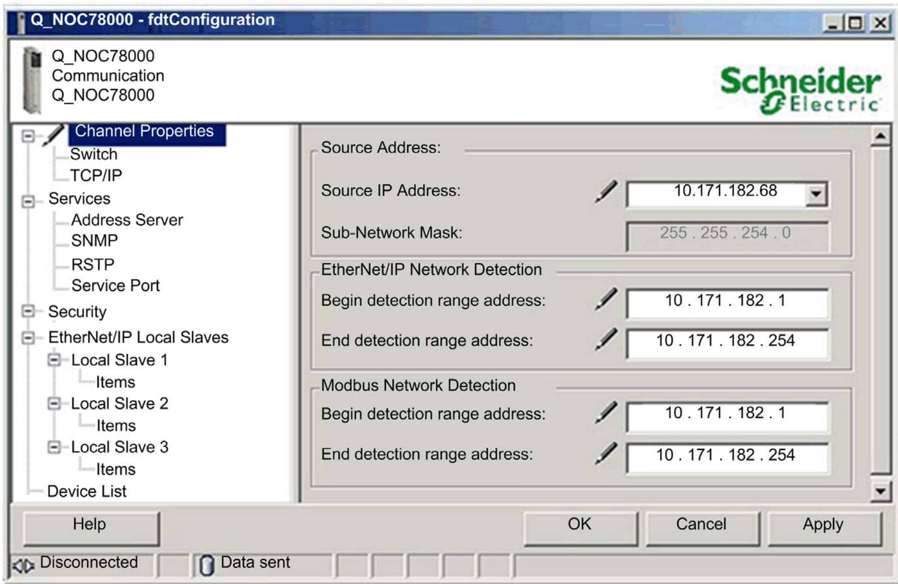
Configuring Properties in the Device Editor

Configuring Properties

The **Device Editor** can be opened from the **DTM Browser**.

To open the **DTM Browser** select **Tools** → **DTM Browser** in the Control Expert main menu.

To use the **Device Editor**:

Step	Description
1	Confirm that the DTM you want to use is not connected to the actual communication module or device. If necessary, disconnect the DTM from the module or device *(see page 90)*.
2	In the **DTM Browser**, select the Ethernet network node you want to configure, which can be either: ● an Ethernet communication module — or — ● a remote device
3	With a node selected in the **DTM Browser**, do one of the following: ● In the Control Expert main menu, select **Edit** → **Open**. — or — ● In the **DTM Browser** click the right mouse button and, in the pop-up menu, select **Open**. The **Device Editor** appears. It displays the configurable properties for the selected module or device: 
4	Expand the navigation tree and select a node in the left window pane to display its properties in the right pane. The list of configurable properties varies, depending on the node type, communication module or remote device, selected in the **DTM Browser**.

<table>
<tr><th>Step</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4">5</td><td colspan="2">While you edit a parameter, Control Expert displays an icon next to the field you are editing and in the navigation tree indicating the parameter value is being edited. Control Expert displays one of the following icons:</td></tr>
<tr><td>This icon...</td><td>Indicates the importance of the parameter being edited is...</td></tr>
<tr><td></td><td>High: Editing this parameter may limit or deny access to the module or device.</td></tr>
<tr><td></td><td>Low: Editing this parameter will not limit or deny access to the module or device.</td></tr>
<tr><td>6</td><td colspan="2">After you finish editing a page, click:
● Apply to save your edits and keep the page open.
— or —
● OK to save your edits and close the page.

NOTE: Your edits will not take effect until they are successfully downloaded from your PC to the CPU and from the CPU to the communication modules and network devices.</td></tr>
</table>

Uploading and Downloading DTM-Based Applications

Introduction

You can use Control Expert to download an application file from your PC to the PLC, and to upload an application file from the PLC to your PC.

To successfully complete an upload, the application file needs to include specific upload-related information as part of the application.

Downloading DTM-Based Applications

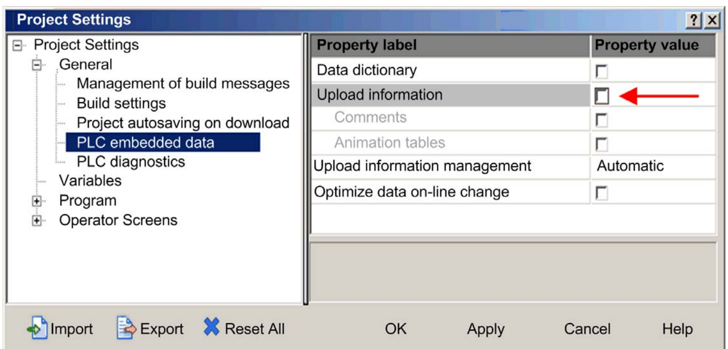
Control Expert applications that include DTM files require more memory than traditional Control Expert applications. The following products employ DTMs for network configuration:

- 140 NOC 771 01 Ethernet communication module for Quantum
- TSX ETC 101 Ethernet communication module for Premium
- BMX NOC 0401 Ethernet communication module for M340
- 140NOC78•00 Ethernet communication module for Quantum

In some cases, the configurations created for these modules and the data associated with them require more memory than is available in the CPU.

If the amount of memory required by an application exceeds the amount of memory that is available in the CPU, Control Expert provides notice of this condition during the build process, before the application is downloaded to the PLC.

When this situation occurs, exclude the additional upload-related information from the application to complete the build and enable the application download. To do this, make the following configuration change in Control Expert:

Step	Action
1	In the main menu, select **Tools → Project Settings...** The **Project Settings** window opens.
2	In the left pane of the **Project Settings** window, select **General → PLC embedded data**.
3	In the right pane, de-select **Upload information**: 
4	Click **OK** to save your changes and close the **Project Settings** window.

After the **Upload information** setting is disabled, you can build the application and download it to the PLC.

NOTE: An application in which the **Upload information** setting has been disabled cannot later be uploaded from the PLC to the PC.

Uploading DTM-Based Applications

DTM-based applications that were successfully downloaded to Control Expert with the project's **Upload information** setting enabled can later be uploaded from the PLC to the PC if the following pre-conditions exist:

DTM-based applications that were successfully downloaded to Control Expert with the project's **Upload information** setting enabled can later be uploaded from the PLC to the PC if the target PC has the following files installed on it:

- a version of Control Expert that is equal to or later than the version used to create the application
- the master DTMs for the modules included in the configuration
  NOTE: The Ethernet Configuration Tool installation CD contains the Master DTMs for all the Ethernet communication modules, referenced above.
- the device DTMs for all DTM-based devices attached to the network (the DTMs are of the same or higher revision as each device DTM used in the configuration)
- the device EDS files for any EtherNet/IP device used in the configuration (the EDS files are of the same or higher revision as each device EDS file used in the configuration)

After all the above components have been installed on the target PC, you can upload a DTM-based Control Expert application from a PLC.

NOTE: All of the above DTM components need to be installed on the target PC *before* attempting the upload.

Section 4.3
Hardware Catalog

Overview

Control Expert includes a collection of modules and devices called the **Hardware Catalog** that you can add to a Control Expert project. EtherNet/IP and Modbus TCP devices are located in the hardware catalog’s **DTM Catalog** page. Each device in the catalog is represented by a DTM that defines the parameters of the module or device.

Not all devices in the market today offer device-specific DTMs. Some devices are instead defined by a device-specific EDS file. Control Expert displays each EDS file in the form of a DTM. In this way, you can use Control Expert to configure these Ethernet/IP devices defined by an EDS file in the same way you would configure a DTM-defined device.

Other devices lack both a DTM and an EDS file. You can configure these devices by using a generic DTM that is included in the **DTM Catalog** page.

This section address the topics:
- how to add a DTM to the catalog
- how to add an EDS file to the catalog
- how to update the catalog
- how to remove an EDS file from the catalog

What Is in This Section?

This section contains the following topics:

Adding a DTM to the Control Expert Hardware Catalog

A Manufacturer Defined Process

Before a DTM can be used by the Control Expert **Hardware Catalog**, install the DTM on the host PC, the same PC that is running Control Expert, by means of an installation process defined by the device manufacturer.

Consult your device documentation, provided by the device manufacturer, for information describing how to install a device DTM on your PC.

NOTE: After a device DTM is successfully installed on your PC, update the Control Expert Hardware Catalog *(see page 108)* so the new DTM is visible in the catalog and available to be added to a Control Expert project.

Add an EDS File to the Control Expert Hardware Catalog

Overview

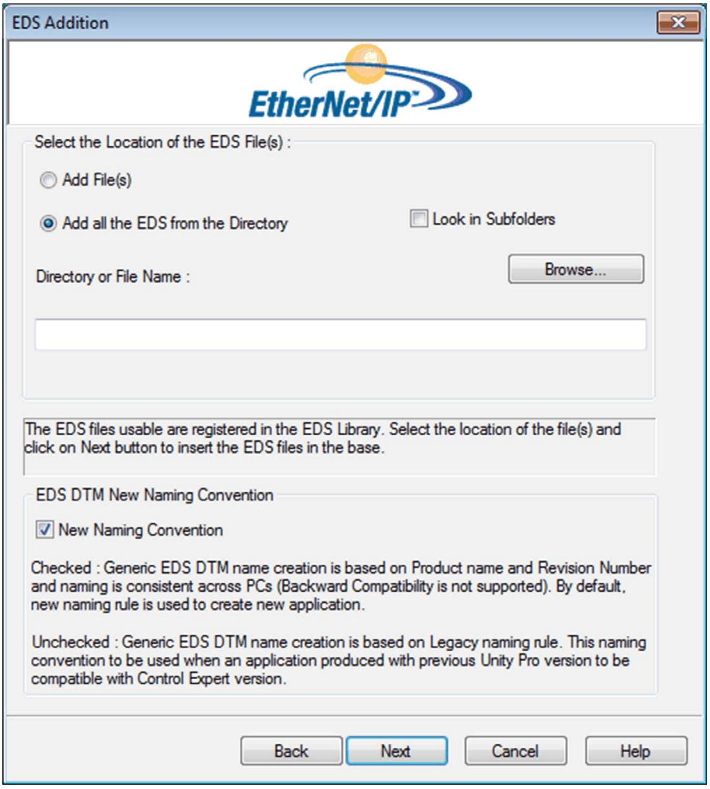
Control Expert includes a wizard you can use to add one or more EDS files to the Control Expert **Hardware Catalog**. The wizard presents a series of instruction screens that:
- simplify the process of adding EDS files to the catalog
- provide a redundancy check in case you attempt to add duplicate EDS files to the catalog

NOTE: The Control Expert **Hardware Catalog** displays a partial collection of DTMs and EDS files registered with the ODVA. This library includes DTMs and EDS files for products not manufactured or sold by Schneider Electric. The non-Schneider Electric EDS files are identified by vendor in the catalog. Please contact the identified device's manufacturer for inquiries regarding the corresponding non-Schneider Electric EDS files.

Adding EDS Files

To add one or more EDS files to the library:

Step	Action
1	If the **DTM Browser** is not already open, in the Control Expert main menu select **Tools → DTM Browser**.
2	In the **DTM Browser**, select a communication module, then click the right mouse button. **Result**: A pop-up menu opens.
3	In the pop-up menu, select **Device menu → Add EDS to library**. **Result**: The introductory page of the wizard opens.

Step	Action
4	Click **Next**. **Result**: Page 2 of the wizard opens: 
5	In the **Select the Location of the EDS File(s)** section, select one of the following: ● **Add File(s)**, to add one or more EDS files you will individually select ● **Add all the EDS from the Directory**, to add all files from a folder you will select. ❍ Select **Look in Subfolders** to also add EDS files in subfolders beneath the folder you selected.
6	Click the **Browse** button. **Result**: The **Open** dialog opens.
7	Use the **Open** dialog to navigate to and select one of the following: ● one or more EDS files ● a folder containing EDS files
8	After you have made your selections, click **Open**. **Result**: The dialog closes, and your selection appears in the **Directory or File Name** field.

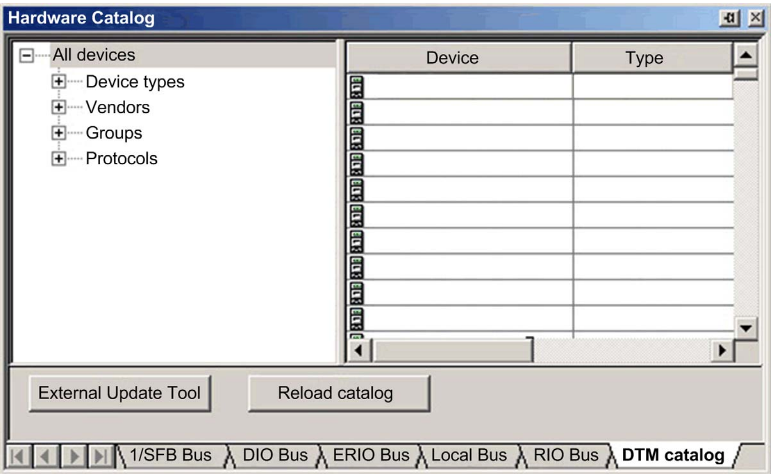
Step	Action
9	Choose the naming convention rule for the EDS DTM name creation. The new naming convention is based on Model Name / Product Name and Revision. A random character is automatically suffixed when Model Name / Product Name and Revision of an EDS file of the library are identical. The new naming convention is irrespective of the order in which EDS files are added to device library. By default, the **New Naming Convention** check box is selected and the new naming rule applies. **NOTE:** To keep backward compatibility with Unity Pro/Control Expert versions, unchecked the **New Naming Convention** check box and the naming rule is based on Model Name / Product Name. **NOTE:** Unity Pro is the former name of Control Expert for version 13.1 or earlier.
10	Click **Next**. **Result**: The wizard compares the selected EDS files against existing files in the library.
11	(Conditional) If one or more selected EDS files is a duplicate, a **File Already Exists** message opens. **Close** the message.
12	Page 3 of the wizard opens indicating the status of each device you attempted to add: ● A green check mark indicates the EDS file can be added. ● A blue informational icon indicates a redundant file. ● A red exclamation point indicates an invalid EDS file. (Optional) Select a file in the list, then click **View Selected File** to open it.
13	Click **Next** to add the non-duplicate files. **Result**: Page 4 of the wizard opens, indicating the action is complete.
14	Click **Finish** to close the wizard.
15	The next step is to update the Control Expert **Hardware Catalog** *(see page 108)*, so that the newly added device is available for inclusion in a Control Expert project.

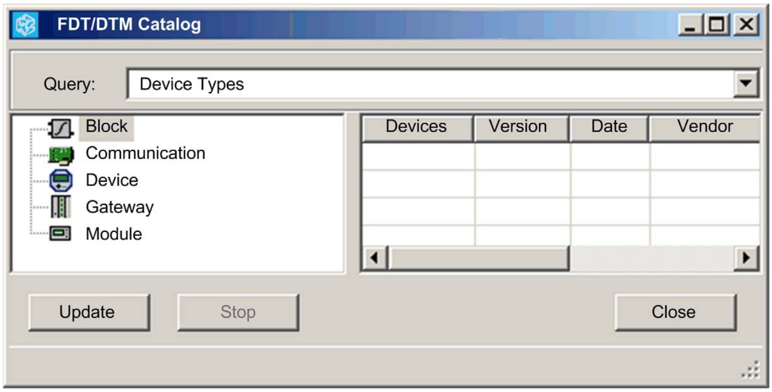
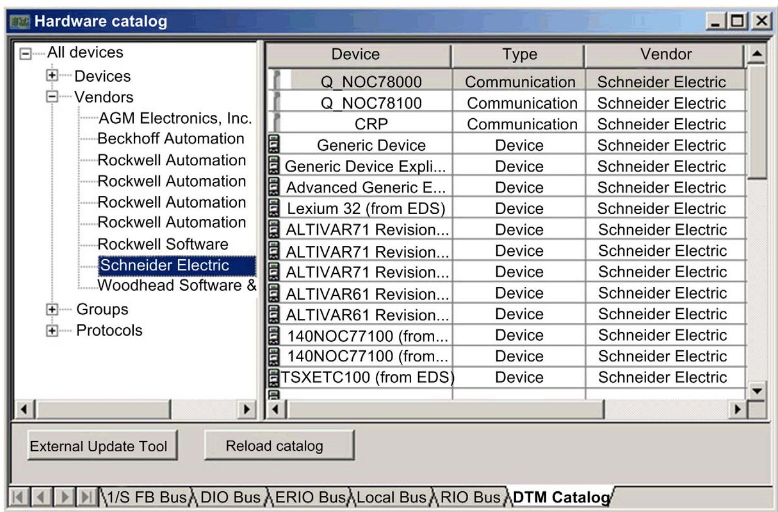
Updating the Control Expert Hardware Catalog

Updating Hardware Catalog

After you have followed the manufacturer's instructions and installed a module or device DTM on your PC, the next step is to update the Control Expert **Hardware Catalog**. Updating the **Hardware Catalog** makes the new Ethernet module or device available for addition to your Control Expert application.

To update the **Hardware Catalog**:

Step	Action
1	In the Control Expert main menu, select **Tools** → **Hardware Catalog**. **Result**: The **Hardware Catalog** window opens.
2	In the **Hardware Catalog** window, select the **DTM Catalog** tab to display a module and device DTM list. At the time of initial software installation, the catalog displays no devices. 

Step	Action
3	Click the **External Update Tool** button. **Result**: The **FDT/DTM Catalog** window opens. 
4	In the **FDT/DTM Catalog** window, click **Update**. The window refreshes itself, as indicated by the progress bar in the lower right corner of the window.
5	After the update has finished, click **Close**. **Result**: The **FDT/DTM Catalog** window closes, and the **Hardware Catalog** displays.
6	In the **Hardware Catalog** window, click **Reload catalog** to refresh the DTM list. 

Remove an EDS File from the Control Expert Hardware Catalog

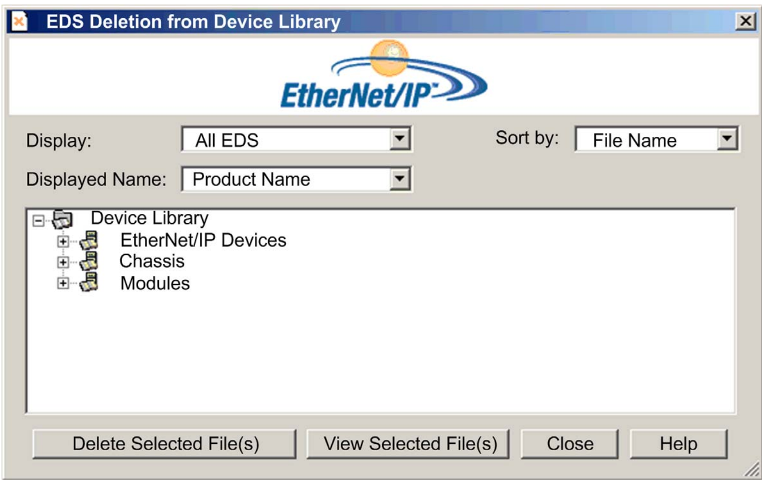
Overview

You can remove a module or device from the list of available devices in the Control Expert **Hardware Catalog** by removing its **EDS** file. When you remove an EDS file from the library, the device or module is no longer displayed by Control Expert in the **DTM Catalog** page of the **Hardware Catalog** window.

However, removing an EDS file from the library does not delete the file. Instead, the EDS file remains in its stored location and can again be added to the catalog *(see page 105)* at a future time.

Removing an EDS File from the Catalog

To remove an EDS file from the catalog:

Step	Action
1	If the **DTM Browser** is not already open, select **Tools** → **DTM Browser** in the Control Expert main menu.
2	In the **DTM Browser**, select a 140NOC78•00 head module, then click the right mouse button. **Result**: A pop-up menu opens.
3	In the pop-up menu, select **Device menu** → **Additional functions** → **Remove EDS from library**. **Result**: The following window opens: 

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="4">4</td><td colspan="2">Use the selection lists in the heading of this window to specify how EDS files are displayed:</td></tr>
<tr><td>Display</td><td>Filters the list of displayed EDS files; select:
● All EDS (no filtering)
● Only Devices
● Only Chassis
● Only Modules</td></tr>
<tr><td>Sort by</td><td>Sorts the list of displayed EDS files; select:
● File Name
● Manufacturer
● Category
● Device Name</td></tr>
<tr><td>Displayed Name</td><td>The description displayed for each device; select:
● Catalog Name
● Product Name</td></tr>
<tr><td>5</td><td colspan="2">In the Device Library tree control, navigate to and select the EDS file you want to remove.</td></tr>
<tr><td>6</td><td colspan="2">(Optional) Click the View Selected File button to display the read-only contents of the selected EDS file.</td></tr>
<tr><td>7</td><td colspan="2">Click the Delete Selected File button.
Result: A message box opens.</td></tr>
<tr><td>8</td><td colspan="2">Click Yes to remove the selected EDS file from the list.</td></tr>
<tr><td>9</td><td colspan="2">When you have finished removing EDS files, click Close.</td></tr>
<tr><td>10</td><td colspan="2">The next step is to update the Hardware Catalog (see page 108).</td></tr>
</table>

Section 4.4
Channel Properties

Overview

This section describes how to configure channel properties for the Ethernet network.

What Is in This Section?

This section contains the following topics:

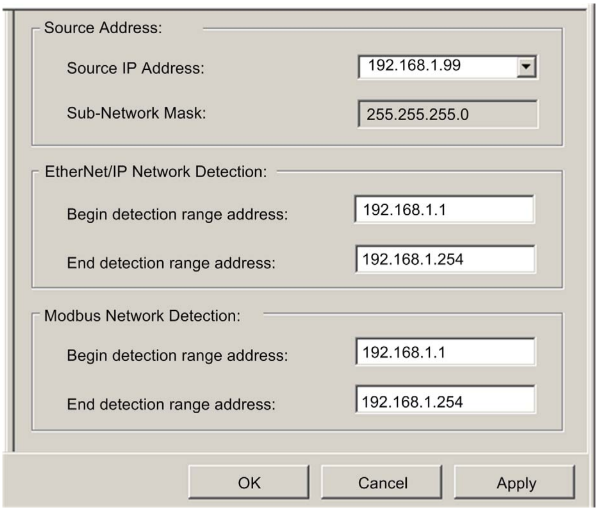
Channel Properties Page

Description

Use the **Channel Properties** page to:

- select the IP address to use for:
  - connecting module or device DTMs to physical devices
  - sending explicit messages to Modbus TCP and EtherNet/IP devices
- view your PC's IP address settings

The **Channel Properties** page looks like this:



To display this page, select the **Channel Properties** node in the navigation tree located on the left side of the **Device Editor**.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)* for instructions on how to edit properties.

Properties

This page presents the following properties:

Name	Description
Source Address area:	
Source IP Address (PC):	A list of IP addresses assigned to network interface cards installed on your PC.
Sub-Network Mask:	The subnet mask associated with the selected Source IP Address.
EtherNet/IP Network Detection area:	
Begin detection range address	The starting IP address of the address range for automatic field bus discovery of EtherNet/IP devices.
End detection range address	The ending IP address of the address range for automatic field bus discovery of EtherNet/IP devices.
Modbus TCP Network Detection area:	
Begin detection range address	The starting IP address of the address range for automatic field bus discovery of Modbus TCP devices.
End detection range address	The ending IP address of the address range for automatic field bus discovery of Modbus TCP devices.

Managing Source IP Address for Multiple PCs

When you connect a PC to a DTM-based Control Expert application, Control Expert requires that you define the IP address of the PC connected to the PLC, which is referred to as the *source IP address (PC)*. Rather than having to perform a **Build** in Control Expert each time you connect a PC to the PLC, the source IP address (PC) is selected automatically when you import the Control Expert application. During application import, the DTM retrieves all available configured NIC addresses of a connected PC and matches the subnet mask of the master with the available NIC list.

- If a match between the subnet mask of the master and the NIC list exists, Control Expert automatically selects the matched IP address as the *source IP address (PC)* in the **Channel Properties** page.
- If multiple matches exist, Control Expert automatically selects the IP address nearest to the subnet mask.
- If no match exists, Control Expert automatically selects the IP address to the nearest available subnet mask.

Channel Properties - Ethernet Page

Description

The **Ethernet** page presents communication settings for Quantum 140 NOC 780 00 and 140 NOC 781 00 Ethernet communication modules. Use this page to:

- view and edit the **Connection Speed**, which includes both the:
  - transmission speed, and
  - duplex mode
- view the **Frame Format**

To display this page, select the **Channel Properties** → **Ethernet** node in the navigation tree located on the left side of the **Device Editor**.

NOTE: Refer to the topic Configuring Properties in the Device Editor for instructions on how to edit properties.

Properties

The **Ethernet** page presents the following properties:

Name	Description
Connection Speed	The transmission speed and duplex mode for the network. Values include: ● Auto 10/100 Mb (default) ● 100 Mb Half ● 100 Mb Full ● 10 Mb Half ● 10 Mb Full **NOTE:** Schneider Electric recommends the default setting—Auto 10/100 Mb. This setting causes the connected devices to perform auto-negotiation and thereby determine the fastest common transmission rate and duplex mode.
Frame Format	Ethernet II is the only available value (read-only).

Channel Properties - Switch Page

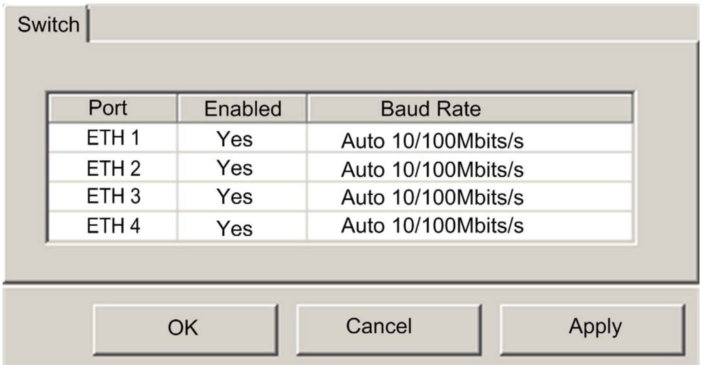
Description

Use the **Switch** tab of the **Switch** page to:

- enable or disable each of the 4 Ethernet ports on the 140NOC78•00 head module
- view and edit the **Baud Rate** for each port, which includes both the:
  - transmission speed
    — and —
  - duplex mode

NOTE: The Ethernet communication module supports only the **Ethernet II** frame type.

The **Switch** page looks like this:



To display this page, select the **Channel Properties → Switch** node in the navigation tree located on the left side of the **Device Editor**. Then click the **Switch** tab.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)* for instructions on how to edit properties.

Properties

This page presents the following properties:

Name	Description
Port	(read-only) The Ethernet port number: 1...4.
Enabled	The active status of the port: ● **Yes** = enabled ● **No** = disabled
Baud Rate	The transmission speed and duplex mode for the network. Values include: ● **Auto 10/100 Mbits/sec** (default) ● **100 Mbits/sec Half duplex** ● **100 Mbits/sec Full duplex** ● **10 Mbits/sec Half duplex** ● **10 Mbits/sec Full duplex** **NOTE:** Schneider Electric recommends the default setting — **Auto 10/100 Mbits/sec**. This setting causes the connected devices to perform auto-negotiation and thereby determine the fastest common transmission rate and duplex mode.

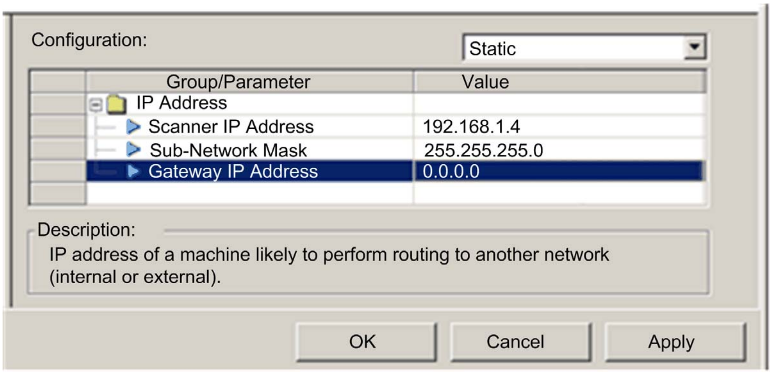
Channel Properties - TCP/IP Page

Description

Use the **TCP/IP** page to:

- select a **Configuration** mode, which specifies how the communication module obtains its IP addressing settings
  — and —
- edit the IP addressing settings that will be used if the **Configuration** mode is set to **Static**

The **TCP/IP** page looks like this:



To display this page, select the **Channel Properties** → **TCP/IP** node in the navigation tree located on the left side of the **Device Editor**.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)* for instructions on how to edit properties.

Selecting a Configuration Mode

Use the **Configuration** list to specify a configuration mode. The configuration mode setting determines how the communication module obtains its IP address at startup. Choices are:

Configuration Mode	Description
Static	The module uses the scanner IP address, gateway IP address, and sub-network mask configured in this page.

Setting the Module Addresses in Static Mode

Configure 3 IP addressing properties for the Ethernet communication module in **Static** configuration mode:

Property	Description
Scanner IP Address	The 32-bit identifier — consisting of both a network address and a host address — assigned to a device connected to a TCP/IP Internet network using the Internet Protocol (IP).
Sub-Network Mask	The 32-bit value used to hide (or mask) the network portion of the IP address and thereby reveal the host address of a device on a network using the IP protocol.
Gateway Address	The address of a device, if any, that serves as a gateway to the communication module.

Default Address Configurations

The communication module uses a default address configuration when it is not configured or when a duplicate IP address is detected. The default address is based on the MAC address of the module and makes it possible for several Schneider Electric devices to use their default network configuration on the same network.

The module uses the following default address configurations:

- No router configured:
  Service port (ETH 1) default IP = 10.10.MAC5.MAC6
  Interlink port (ETH 2) default IP = 10.10.MAC5.MAC6
  Control ports (ETH 3, 4 default IP = 10.10.MAC5.MAC6
- Router configured, fieldbus network configured, service port configured for access mode:
  Service port (ETH 1) default IP = 169.254.10.MAC6
  Interlink port (ETH 2) default IP = 169.254.20.MAC6
  Control ports (ETH 3, 4) default IP = 169.254.10.MAC6
- Router configured, fieldbus network configured, service port configured for extended network:
  Service port (ETH 1) default IP = 169.254.30.MAC6
  Interlink port (ETH 2) default IP = 169.254.20.MAC6
  Control ports (ETH 3, 4) default IP = 169.254.10.MAC6

Duplicate Address Checking

Before going online, the module sends out at least 4 ARP messages with a proposed IP address:

- if an answer is returned:
  - another network device is already using the proposed IP address.
  - the module will not use the proposed IP address, but will instead use the default IP address.
- if an answer is not returned:
  - the module is assigned the proposed IP address (along with the associated network parameters).

NOTE: When powering up an entire network, some switches may be slow to complete the power up process. This can cause some ARP messages to be dropped. To help avoid this situation, Schneider Electric recommends that, when powering up an entire network, confirm that all network switches complete their power up cycle before powering up the PLCs.

Channel Properties - EtherNet/IP Page

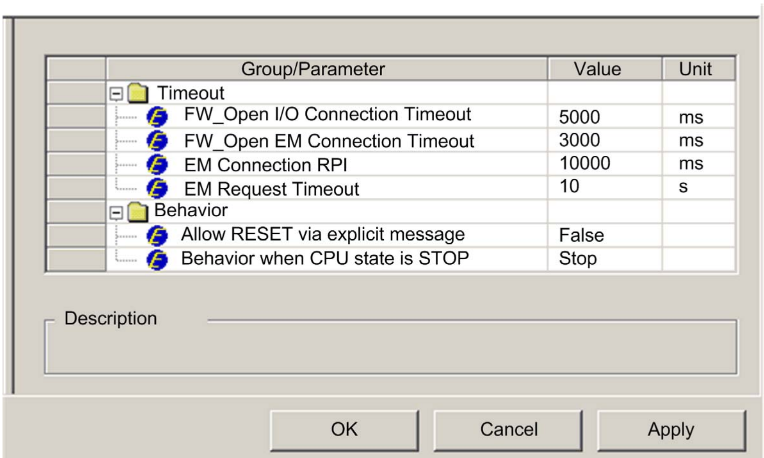
Description

The **EtherNet/IP** page is displayed only when Control Expert is operating in Advanced Mode *(see page 91)*. Advanced mode properties are identified by the icon.

Use the **EtherNet/IP** page to configure the following communication module properties:

- properties that determine how the communication module, as a scanner, opens connections for both implicit and explicit messages
- the frequency for transmitting produced data over implicit messaging connections
- the timeout period for explicit messaging connections
- the behavior of the communication module—as a scanner—when:
  - the application is stopped, or
  - the communication module receives a reset service request

The **EtherNet/IP** page looks like this:



To display this page, select the **Channel Properties → EtherNet/IP** node in the navigation tree located on the left side of the **Device Editor**.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)* for instructions on how to edit properties.

Properties

Note: Users experienced in the configuration of EtherNet/IP networks can edit the following read-write properties.

Name	Description
Timeout	
FW_Open IO Connection Timing	The amount of time the communication module waits for the Forward_Open IO messaging transaction to open an implicit messaging connection. Default = 5000 ms
FW_Open EM Connection Timing	The amount of time the communication module waits for the Forward_Open IO messaging transaction to open an explicit messaging connection. Default = 3000 ms
EM Connected RPI	The value used to set the T->O (target to originator) and O->T (originator to target) requested packet interval (RPI) for explicit message connections. This value is used to calculate the lifetime of a connection. Default = 10000 ms.
EM Request Timeout	The amount of time the communication module waits between a request and reply of an explicit message. Default =10 s.
Output	
Allow reset explicit message	The behavior of the communication module—as scanner—when it receives a reset service request: ● TRUE indicates the module will accept the request and reset itself. ● FALSE indicates the module ignores the reset service request and continues uninterrupted operations. Default = FALSE
Behavior when CPU state is STOP	The state of the communication module when the CPU application goes into a STOP state: ● TRUE indicates that the module enters STOP state (implicit connections are closed). ● FALSE indicates that the module enters IDLE state (implicit connections are not closed). Default = FALSE

Section 4.5
Ethernet Services

Overview

This section describes how to enable and configure Ethernet services provided by the 140NOC78000 head modules.

What Is in This Section?

This section contains the following topics:

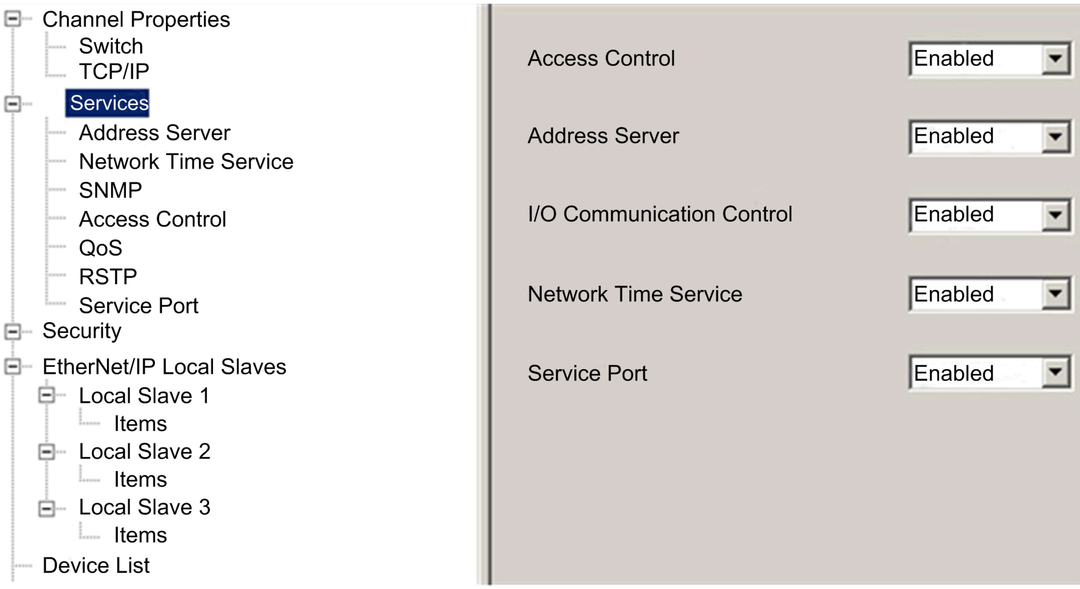
Enabling Ethernet Services

Introduction

Use the **Services** page to enable and disable Ethernet services provided by the 140NOC78000 head module.

NOTE: After you enable a service, you can configure its settings. If you do not configure a service, Control Expert applies its default settings.

The **Services** page looks like this, with all settings configured as **Enabled**:



NOTE: By default, the **Access Control** field is set to **Disabled**. All other fields are set to **Enabled**.

Enabling/Disabling Ethernet Services

To enable or disable Ethernet services on the **Services** page:

Step	Action
1	Double-click the 140NOC78000 head module in the DTM browser or right-click the module and click **Open**.
2	Click **Services** in the navigation tree located in the left panel of the **Device Editor**.
3	For each feature, change the setting as desired. **Access Control** and **Network Time Service** are **Disabled** by default. All other features are **Enabled** by default.
4	Click **Apply** to save changes, or click **OK** to save changes and close the window.

When you **Enable** a service, Control Expert displays a node for that service in the navigation tree in the left panel of the **Device Editor**, beneath the **Services** parent node. Click a service node to access its settings.

When you **Disable** a service, Control Expert hides the node for that service.

NOTE: Refer to Configuring Properties in the Device Editor *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)* for instructions on how to edit properties.

Ethernet Services Descriptions

The 140NOC78000 head module can be configured to provide the following Ethernet services:

If this service is enabled...	The module can...
Access Control	deny access to the 140NOC78000 head module by unauthorized network devices.
Address Server	provide both IP addressing parameters and operating parameters to other Ethernet devices.
I/O Communication Control	allow the Control Expert application to control the enabling and disabling of individual connections between the head module and remote I/O devices. **NOTE:** ● The application can open and close individual connections using the control bits located at the beginning of the output area. ● If this service is disabled, the user — via the application program — cannot toggle on and off connection control bits
Network Time Service	synchronize computer clocks over the Internet for the purposes of event recording (sequence events), event synchronization (trigger simultaneous events), or alarm and I/O synchronization (time stamp alarms).

If this service is enabled...	The module can...
Service Port	support 2 functions: • **Access Port**: You can connect an Ethernet device (ex: an HMI or a PC with Control Expert software or a PC with ConneXium Network Manager software) to this port, and communicate with the CPU/PLC, the 140NOC78000 head module itself, or access other devices connected to the network. • **Port Mirroring**: You can connect to this port via a PC and sniff (using Ethereal, WireShark, etc.) the traffic that is travelling through the other ports including the Ethernet port — the internal port — that is connected to the CPU.
SNMP	• serve as an SNMP v1 agent. • provide trap information to up to two devices configured as SNMP managers. **NOTE:** The SNMP service is always enabled.
QoS Tagging	add *Differentiated Services Code Point* (DSCP) tags to Ethernet packets so that network switches can prioritize the transmission and forwarding of Ethernet packets. **NOTE:** The DSCP service is always enabled.
RSTP	create a loop-free logical network path for Ethernet devices that are part of a topology that includes redundant physical paths. The RSTP-enabled module also automatically restores network communication – by activating redundant links – in the event the network experiences a loss of service. **NOTE:** The DSCP service is always enabled.

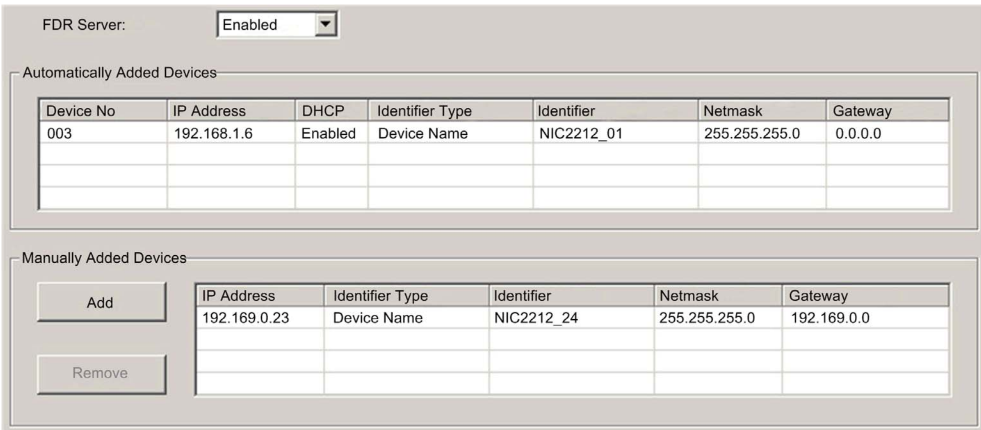
Configuring the DHCP and FDR Servers

Introduction

The 140 NOC 78• 00 head module includes both a DHCP and a fast device replacement (FDR) server. The DHCP server provides IP address settings to networked Ethernet devices. The FDR server provides operating parameter settings to replacement Ethernet devices that are equipped with FDR client functionality.

Use the **Address Server** page to:
- enable and disable the 140 NOC 78• 00 head module's FDR service
- view an automatically generated list of all devices included in the 140 NOC 78• 00 head module's Ethernet configuration, displaying for each device:
  - IP addressing parameters
  - whether the device's IP addressing parameters are provided by the 140 NOC 78• 00 head module's embedded DHCP server
- manually add remote devices — that are not part of the 140 NOC 78• 00 head module's Ethernet configuration — to the 140 NOC 78• 00 module's DHCP client list
  NOTE: Remote devices added in this way are equipped with DHCP client software, and are configured to subscribe to the 140 NOC 78• 00 head module's IP addressing service.

The **Address Server** page looks like this:



Configuring the Address Server

To configure the **Address Server**:

Step	Action
1	Open the 140 NOC 78• 00 head module in the DTM browser.
2	In the **Services** page, set the **Address Server** field to **Enabled**.
3	Click the **Address Server** node in the **Services** navigation tree in the left panel.
4	● Enable or disable the **FDR Server** field. ● View an automatically generated list of all devices included in the 140 NOC 78• 00 head module's Ethernet configuration, displaying for each device: ❍ IP addressing parameters ❍ whether the device's IP addressing parameters are provided by the 140 NOC 78• 00 head module's embedded DHCP server ● Manually add remote devices to the DHCP service if necessary.
5	Click **Apply** to save changes, or click **OK** to save changes and close the window.

Enabling the FDR Service

Before enabling the FDR service, enable the FTP/TFTP services *(see page 154)*.

To enable the 140 NOC 78• 00 head module's FDR service, set the **FDR Server** field to **Enabled**. To disable the service, toggle the same field to **Disabled**.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)* for instructions on how to apply edited properties to networked devices.

Any networked Ethernet device equipped with FDR client functionality can subscribe to the 140 NOC 78• 00 head module's FDR service. The module can store up to 1 MB of FDR client operating parameter files. When this file storage capacity is reached, the module can not store any additional client FDR files.

The 140 NOC 78• 00 head module can store FDR client files for up to 128 devices, depending on the size of each stored file. For example, if the size of each FDR client file is small — not more than 8 Kb — the module could store up to the maximum of 128 parameter files.

Manually Adding Remote Devices to the DHCP Service

Remote devices that are part of the 140 NOC 78• 00 head module's Ethernet configuration — and which have subscribed to the 140 NOC 78• 00 head module's IP addressing service — automatically appear in the **Automatically Added Devices** list.

Other remote devices — that are not part of the 140 NOC 78• 00 head module's configuration — can be manually added to the module's DHCP IP addressing service.

To manually add networked Ethernet devices — which are not part of the 140 NOC 78• 00 head module's Ethernet configuration — to the module's IP addressing service:

<table>
<tr><th>Step</th><th colspan="2">Description</th></tr>
<tr><td>1</td><td colspan="2">In the Address Server page, click the Add button in the Manually Added Devices field. Control Expert adds an empty row to the list of Manually Added Devices.</td></tr>
<tr><td rowspan="6">2</td><td colspan="2">In the new row, configure the following parameters for the client device:</td></tr>
<tr><td>IP Address</td><td>Type in the IP address of the client device.</td></tr>
<tr><td>Identifier Type</td><td>Select the type of value the client device will use to identify itself to the FDR server:
● MAC address
● device Name</td></tr>
<tr><td>Identifier</td><td>Depending upon the identifier type, type in the client device setting for the MAC address or name.</td></tr>
<tr><td>Netmask</td><td>Type in the client device subnet mask.</td></tr>
<tr><td>Gateway</td><td>Type in the gateway address that remote devices can use to communicate with devices located on other networks. Use 0.0.0.0 if remote devices will not communicate with devices located on other networks.</td></tr>
<tr><td>3</td><td colspan="2">Refer to the topic Configuring Properties in the Device Editor (see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual) for instructions on how to apply edited properties to networked devices.</td></tr>
</table>

Viewing the Auto-Generated DHCP Client List

The list of **Automatically Added Devices** includes a row for each remote device that is:

- part of the 140 NOC 78• 00 head module's Ethernet configuration
- configured to subscribe to the 140 NOC 78• 00 head module's DHCP addressing service

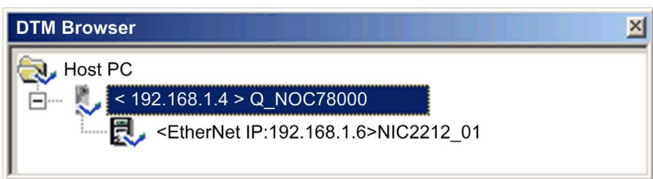
NOTE: You cannot add devices to this list in this page. Instead, use the configuration pages for the remote device to subscribe to this service.

The list of **Automatically Added Devices** contains the following information for each networked device:

Property	Description
Device No	The number assigned to the device in the Control Expert configuration.
IP Address	The client device IP address.
DHCP	TRUE indicates that the device subscribes to the DHCP service.
Identifier Type	Indicates the mechanism used by the server to recognize the client (MAC address or DHCP device name).
Identifier	The actual MAC address or DHCP device name.
Netmask	The client device subnet mask.
Gateway	The IP address a DHCP client device will use to access other devices that are not located on the local subnet. A value of 0.0.0.0 constrains the DHCP client device by allowing it to communicate only with devices on the local subnet.

Subscribing to the DHCP Service for a Device that is Part of the Configuration

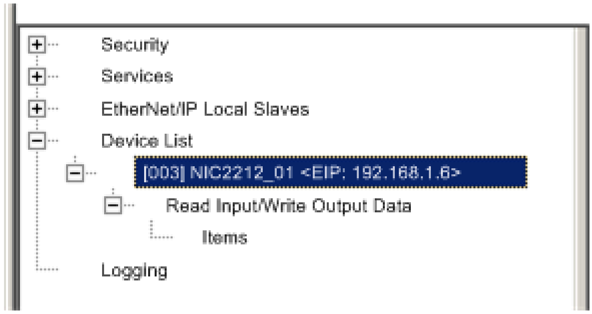
An Ethernet device — that is part of the 140 NOC 78• 00 head module's Ethernet configuration — can subscribe to the module's IP addressing service. To subscribe to this service, follow these steps:

Step	Action
1	In the **DTM Browser**, select the 140 NOC 780 00 head module that is connected to the remote device that you want to add to the DHCP service. In the following example, the module with the alias name of Q_NOC78000 is selected:  **NOTE:** The selected module is connected to the STB NIC 2212 network interface device bearing the alias name NIC2212_01, which is the module you want to add to the DHCP service.

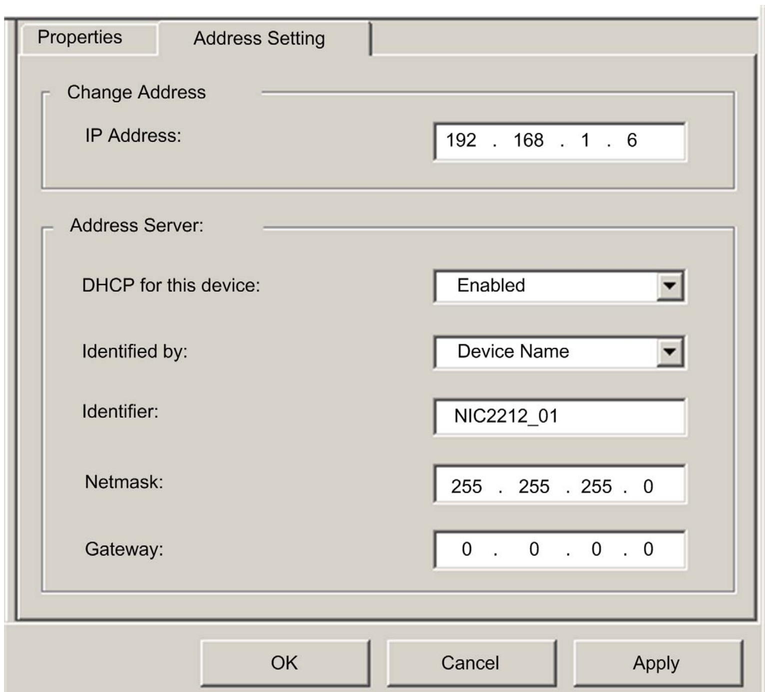
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2</td><td>With Q_NOC78000 selected in the DTM Browser, right-click and select Open in the pop-up menu.:


The Device Editor opens.</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>3</td><td>In the navigation tree on the left side of the Device Editor, expand the Device List node and select the device for which you want to enable the DHCP service. In this example, select NIC2212_01:



Control Expert displays the properties for the selected remote device in the right pane of the window.</td></tr>
<tr><td>4</td><td>In the right pane of the window, select the Address Setting tab to display the following page:


</td></tr>
</table>

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="6">5</td><td colspan="2">In the Address Server area of this page, configure the following properties:</td></tr>
<tr><td>DHCP for this device</td><td>Select Enabled</td></tr>
<tr><td>Identified by</td><td>The choices are:
● MAC Address, or
● Device Name

Select Device Name.</td></tr>
<tr><td>Identifier</td><td>Control Expert has automatically added the device name Q_NOC78000. For the purpose of this example, accept this default value.</td></tr>
<tr><td>Netmask</td><td>Control Expert has automatically applied the same netmask used for the 140 NOC 780 00 head module. For the purpose of this example, accept the default value of 255.255.255.0.</td></tr>
<tr><td>Gateway</td><td>For the purpose of this example, accept the default value of 0.0.0.0.</td></tr>
<tr><td>6</td><td colspan="2">Click Apply to save changes, or click OK to save changes and close the window.

NOTE: Refer to the topic Configuring Properties in the Device Editor (see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual) for more information on editing and saving property settings in this window.</td></tr>
</table>

Configuring the SNMP Agent

Description

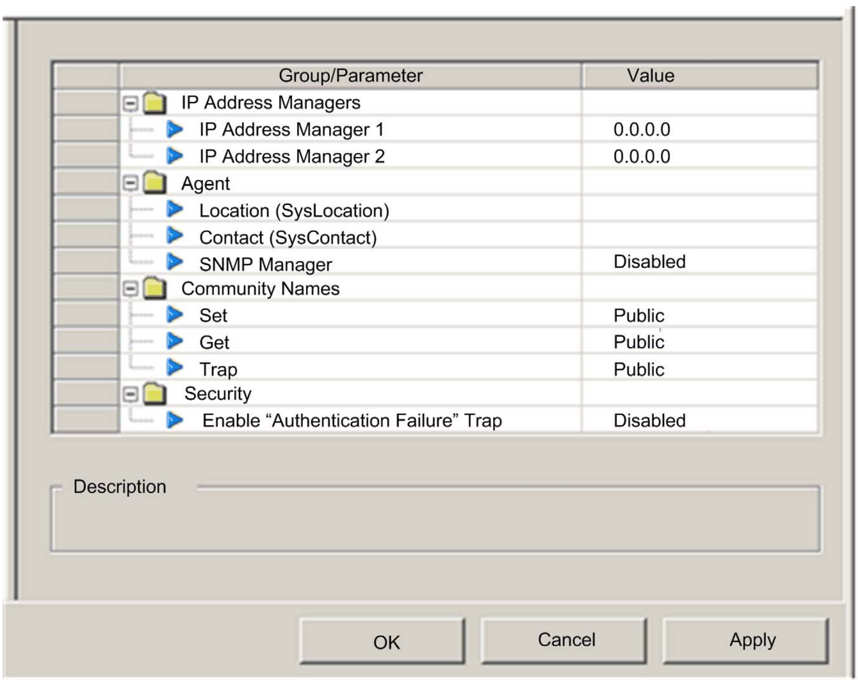
The 140 NOC 78• 00 head module includes an SNMP v1 agent. An SNMP agent is a software component running on the communication module that allows access to the module's diagnostic and management information via the SNMP service.

SNMP browsers, network management software, and other tools typically use SNMP to access this data. In addition, the SNMP agent can be configured with the IP address of up to 2 devices, typically PCs running network management software, to be the target of event driven trap messages. These trap messages inform the management device of events such as cold start and unauthorized access.

Use the **SNMP** page to configure the SNMP agent in the 140 NOC 78• 00 head module. The SNMP agent can connect to and communicate with up to 2 SNMP managers as part of an SNMP service. The SNMP service includes:

- authentication checking, by the 140 NOC 78• 00 head module, of any SNMP manager that sends SNMP requests
- management of event, or trap, reporting by the module

The **SNMP** page looks like this:



To display this page:

Step	Description
1	Click **Services** in the navigation tree in the left panel of the **Device Editor**. **Result**:The **Services** page opens.
2	In the **Services** page, set the **SNMP** field to **Enabled**. Then click **Apply**. **Result**:**SNMP** appears in the navigation tree.
3	Select the **SNMP** in the navigation tree.
4	Click **Apply** to save changes and leave the window open, or click **OK** to save changes and close the window.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)* for instructions on how to edit properties.

Viewing and Configuring SNMP Properties

NOTE: The sysName SNMP parameter is neither editable nor visible in the Control Expert Ethernet Configuration Tool software. By default, the sysName is set to the 140 NOC 78• 00 head module part number.
When DHCP is enabled and **Device Name** is selected as the DHCP identifier for the module, the SNMP sysName parameter is not set to the module part number, but is instead the device name.

The following properties can be viewed and edited in the **SNMP** page:

Property	Description
IP Address Managers:	
IP Address Manager 1	The IP address of the first SNMP manager to which the SNMP agent sends notices of traps.
IP Address Manager 2	The IP address of the second SNMP manager to which the SNMP agent sends notices of traps.
Agent:	
Location	The device location (32 characters maximum)
Contact	Information describing the person to contact for device maintenance (32 characters maximum)
SNMP Manager	Select either: ● **TRUE**: the Location and Contact information are editable in this page ● **FALSE**: Location and Contact settings are not editable in this page
Community Names:	
Get	Password required by the SNMP agent before executing read commands from an SNMP manager. Default = **Public**.
Set	Password required by the SNMP agent before executing write commands from an SNMP manager. Default = **Public**

Property	Description
Trap	Password an SNMP manager requires from the SNMP agent before the manager will accept trap notices from the agent. Default = **Public**
Security:	
Enable Authentication Failure Trap	**TRUE** causes the SNMP agent to send a trap notice to the SNMP manager if an unauthorized manager sends a Get or Set command to the agent. Default = **FALSE**.

Configuring Access Control

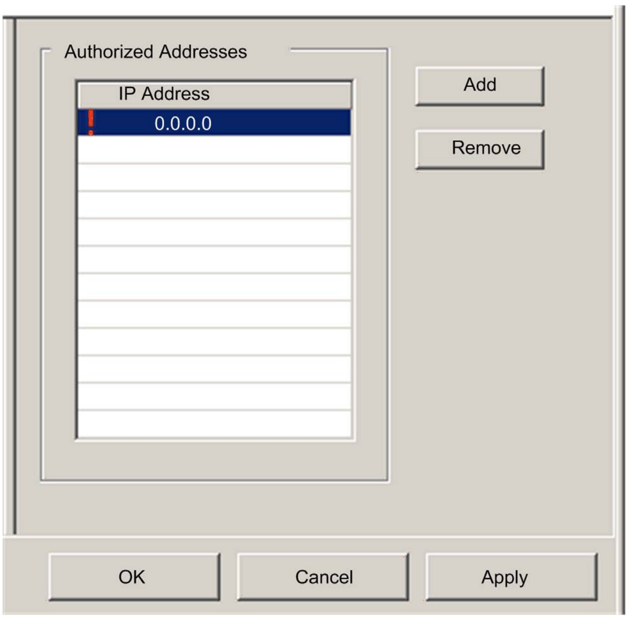
Description

Use the **Access Control** page to restrict access to the 140NOC78•00 head module in its role as either a Modbus TCP or EtherNet/IP server. When access control is enabled in the **Services** page, add the IP addresses of the following devices to the list of **Authorized Addresses** to permit communication with that device:

- the 140NOC78•00 head module itself, so that the module can use EtherNet/IP explicit messaging for any of the following purposes:
  - obtaining diagnostic data
  - resetting the module
  - changing the IP address
- any client device that may send a request to the 140NOC78•00 head module, in its role as either Modbus TCP or EtherNet/IP server
- your own maintenance PC, so that you can communicate with the PLC via Control Expert to configure and diagnose your application, and to view the module’s web pages
- any target device to which the 140NOC78•00 head module may send a Modbus TCP explicit message

  NOTE: You do not need to list the IP address of a target device to which the communication module may send an EtherNet/IP explicit message.

The following graphic depicts the **Access Control** page immediately after a new row has been added to the list of **Authorized Addresses**, but before the new item has been configured:



To display this page:

Step	Description
1	Select the **Services** node in the navigation tree located in the left panel of the **Device Editor**. **Result**:The **Services** page opens.
2	In the **Services** page, set the **Access Control** field to **Enabled** and click either **OK** or **Apply**. **Result**:The **Access Control** node appears in the navigation tree.
3	Select the **Access Control** node in the navigation tree.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)* for instructions on how to edit properties.

Adding and Removing Devices in the Authorized Address List

To add a device to the **Authorized Addresses** list:

Step	Description
1	In the **Access Control** page, click **Add**. A new row appears in the **Authorized Addresses** list, displaying: ● a red exclamation point, indicating editing has begun ● a placeholder IP address of 0.0.0.0
2	Double-click the placeholder IP address. **Result**:The IP address field expands and becomes editable.
3	In the new IP address field, type the IP address of the device that will be able to access the 140NOC78•00 head module, then press **Enter**.
4	Repeat steps 1 through 3, above, for each additional device for which you want to grant access to the 140NOC78•00 head module.
5	Refer to the topic Configuring Properties in the Device Editor *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)* for instructions on how to save your configuration edits.

To remove a device from the **Authorized Addresses** list, select its IP address in the list, then click **Remove**. The selected IP address is removed.

Configuring QoS Ethernet Packet Tagging

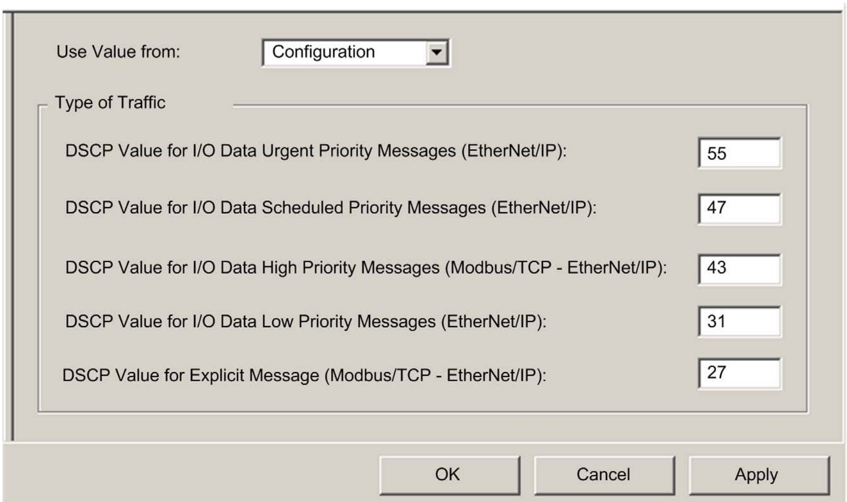
Description

The Ethernet communication module can be configured to perform Ethernet packet tagging. The module supports the OSI layer 3 Quality of Service (QoS) standard defined in RFC-2475. When you enable QoS, the module adds a *differentiated services code point* (DSCP) tag to each Ethernet packet it transmits, thereby indicating the priority of that packet.

Use the **QoS** page to:

- specify the source of QoS packet priority settings, and
- view or edit the five QoS DSCP prioritization values

The **QoS** page looks like this:



To display this page:

Step	Description
1	Select the **Services** node in the navigation tree located on the left side of the **Device Editor**. The **Services** page opens.
2	In the **Services** page, set the **QoS Tagging** field to **Enabled**, then click **OK** or **Apply**. The **QoS** node appears in the navigation tree.
3	Select the **QoS** node in the navigation tree.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)* for instructions on how to edit properties.

Specifying the Source of QoS Settings

The five QoS prioritization values can be set either from the communication module's flash memory, or in this page. To specify the QoS configuration source, set the **Use value from** field to either:

Setting	Description
Configuration[1]	The communication module uses the settings input in the **Type of Traffic** section of this page.
Flash[1]	The communication module uses the settings saved in the module's flash memory. The fields in the **Type of Traffic** section are read-only.
1. Schneider Electric recommends that QoS values be set in the configuration, and not by saving settings to flash memory. Settings saved to flash memory will be lost if the module is replaced.	

NOTE: You can also edit QoS configuration settings by using explicit messages to set the attributes of the QoS CIP object *(see page 274)*.

Type of Traffic Settings

QoS tagging lets you prioritize Ethernet packet streams based on the type of traffic in that stream. The communication module recognizes the traffic types described below. When the **Use value from** field is set to **Configuration**, you can edit the prioritization values in this page. Each traffic type can have a prioritization value from 0... 63.

Traffic Type	Default
DSCP Value for IO Data Urgent Priority Messages (EtherNet/IP)	55
DSCP Value for IO Data Scheduled Priority Messages (EtherNet/IP)	47
DSCP Value for IO Data High Priority Messages (Modbus TCP & EtherNet/IP)	43
DSCP Value for IO Data Low Priority Messages (EtherNet/IP)	31
DSCP Value for Explicit Message (Modbus TCP & EtherNet/IP)	27

To effectively implement QoS settings in your Ethernet network:
- Use network switches that support QoS.
- Consistently apply DSCP values to network devices and switches that support DSCP.
- Confirm that switches apply a consistent set of rules for sorting DSCP tags, when transmitting and receiving Ethernet packets.

NOTE: The QoS settings for Scheduled, High, and Low priority messages also apply to input and output priority messages for a remote device. You can configure these settings for a remote device *(see Premium using EcoStruxure™ Control Expert, TSX ETC 101 Ethernet Communication Module, User Manual)* in the **Device Editor** by selecting a device connection node, then opening the connection's **General** page.

Configuring the Rapid Spanning Tree Protocol

Description

Ethernet ports 3 and 4, located on the front of the 140 NOC 78• 00 head module, support the *Rapid Spanning Tree Protocol* (RSTP). RSTP is an OSI layer 2 protocol defined by IEEE 802.1D 2004. RSTP performs 2 services:

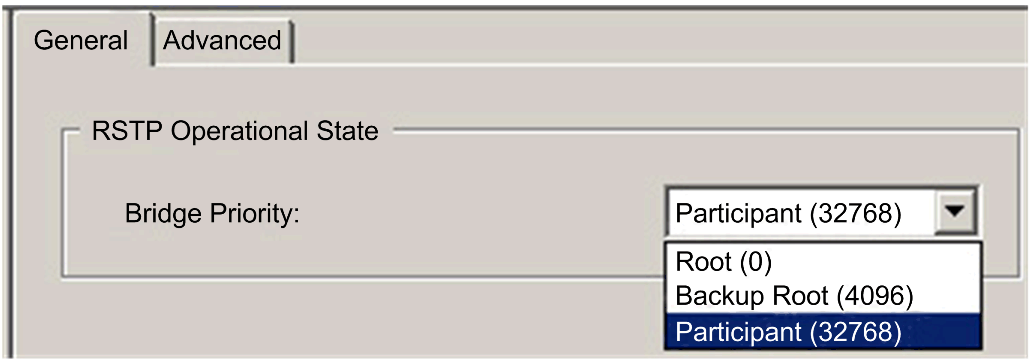
- It creates a loop-free logical network path for Ethernet devices that are part of a topology that includes redundant physical paths.
- It automatically restores network communication — by activating redundant links — in the event the network experiences a loss of service.

RSTP software, operating simultaneously in all network switches, obtains information from each switch, which enables the software to create a hierarchical logical network topology. RSTP is a flexible protocol that can be implemented on many physical topologies, including ring, mesh, or a combination of ring and mesh.

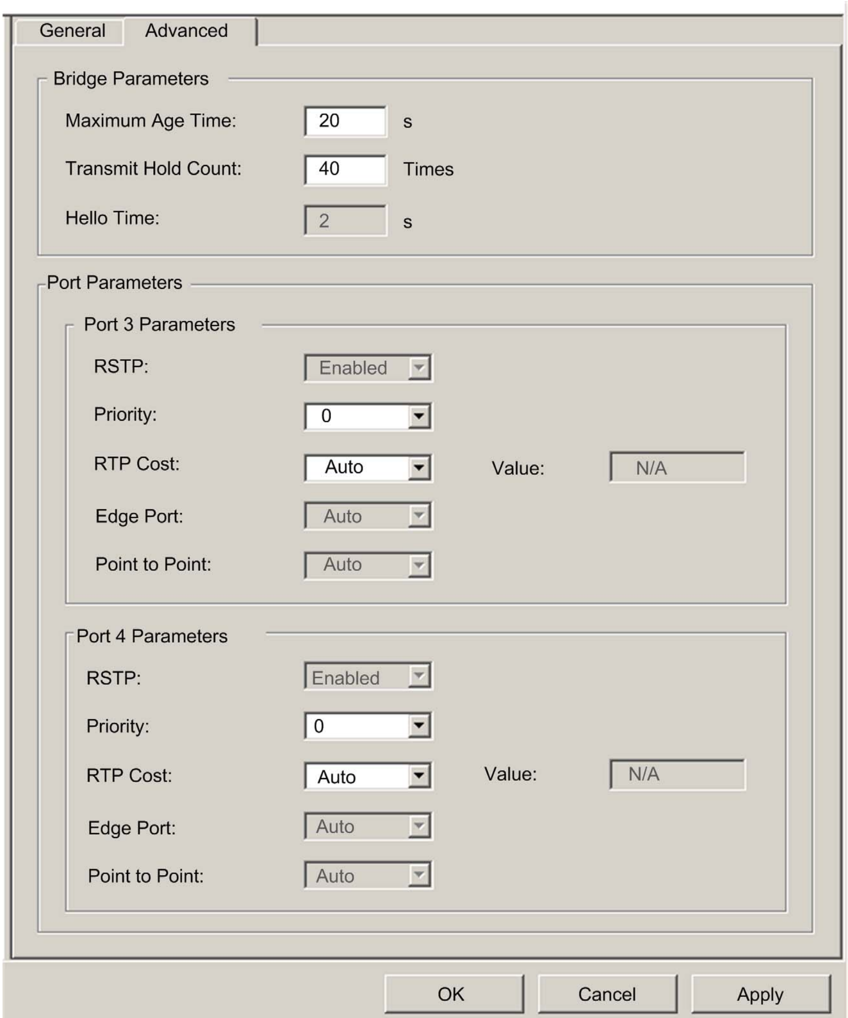
Use the **RSTP → General** and the **RSTP → Advanced** pages to configure RSTP for the embedded Ethernet switch in the 140 NOC 78• 00 head module.

NOTE: RSTP can be implemented only when all network switches are configured to support RSTP.

The **RSTP → General** page looks like this:



The **RSTP → Advanced** page looks like this:



NOTE: The **Advanced** page is available only when you enable advanced mode.

To display these pages:

Step	Description
1	Select the **Services** node in the navigation tree located on the left side of the **Device Editor**. **Result**:The **Services** page opens.
2	In the **Services** page, set the **RSTP** field to **Enabled**, then click **OK** or **Apply**. **Result**: The **RSTP** node appears in the navigation tree.
3	Select the **RSTP** node in the navigation tree, then click on either the **General** or **Advanced** tab to display that page.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)* for instructions on how to edit properties.

Configuring RSTP Properties

The following properties can be viewed and edited in the **RSTP** → **General** page:

Property	Description
RSTP Operational State: Bridge Priority	Select one of the following values in the drop-down list: ● Root (0) ● Backup Root (4096) ● Participant (32768) **NOTE:** Network switches running RSTP software periodically exchange information about themselves using special packets called Bridge Protocol Data Units (BPDUs), which act as a heartbeat. The Bridge Priority value is contained in the BPDU and establishes the relative position of the switch in the RSTP hierarchy.

The following properties can be viewed and edited in the **RSTP → Advanced** page:

Property	Description
Bridge Parameters:	
Maximum Age Time	This value is set to The length of time, from 6 to 40 seconds, that the switch waits for receipt of the next hello message, before initiating a change to the RSTP topology. Default = 40 s.
Transmit Hold Count	The maximum number of BPDUs, from 1 to 40, that the switch can transmit per second. Default = 40.
Hello Time	(read-only) The frequency—set at 2 seconds—that the embedded switch sends heartbeat BPDUs.
Port Parameters (These properties can be separately configured for ports 3 and 4):	
RSTP	(read-only) This property is set to **Enabled** in the **Services** page.
Priority	The priority assigned to the switch port, an integer from 0 to 240 in increments of 16. Default = 0. This value is used by the RSTP process if it needs to break a tie between two ports on the same switch when identifying a: ● root port: the port on a non-root switch that is closest to the root bridge in terms of path cost, or ● designated port: the port at one end of a network segment through which traffic passes on its way to the root bridge
Path Cost	The method used to determine the path cost through the embedded switch. Values include: ● **Auto**: The RSTP protocol automatically assigns a value to the switch by operation of the RSTP algorithm. ● **Manual**: Input the RSTP cost — an integer from 1 to 200000000 — in the **Value** field.
Edge Port	(read-only) Set to a fixed value of **Auto**. The RSTP process automatically determines if the port is an RSTP edge port.
Point to Point	(read-only) Set to a fixed value of **Auto**. The RSTP process automatically determines if the port is an RSTP point-to-point port.

Configuring the Service Port

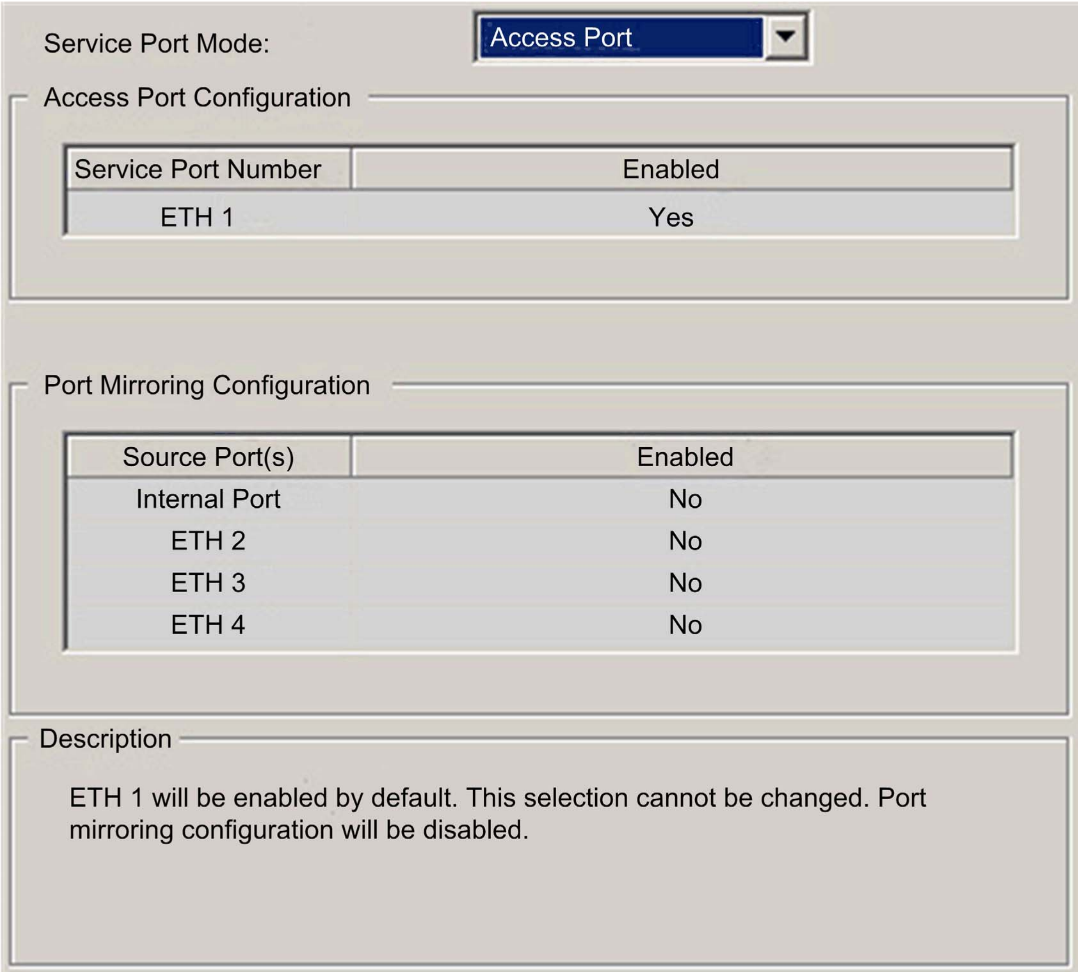
Introduction

The 140NOC78000 head module includes a service port that can be configured to support 2 functions:

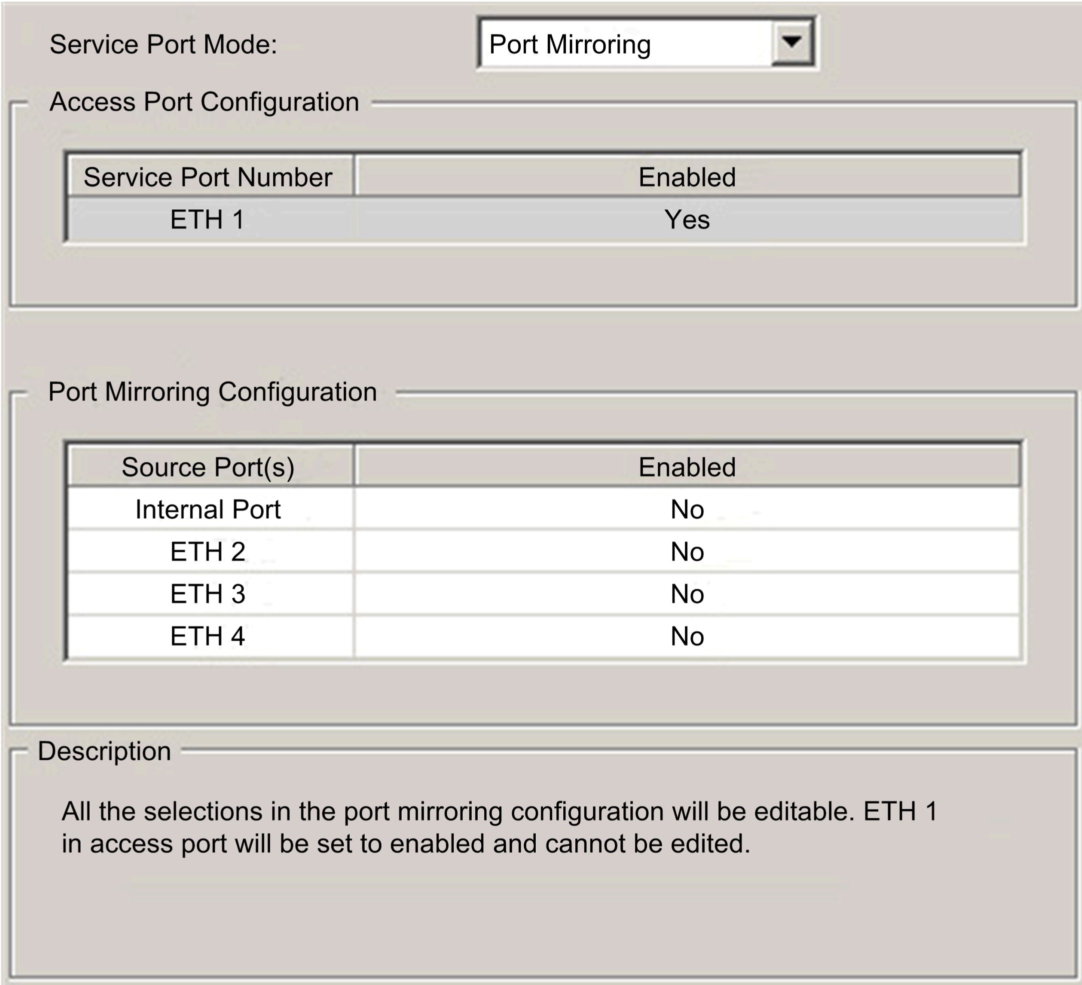
- **Access Port**: If you configure the service port as access port, you can connect an Ethernet device (example: an HMI or a PC with Control Expert software or a PC with ConneXium Network Manager software) to this port, and communicate with the CPU/PLC, the 140NOC78000 head module itself, or access other devices connected to the network.
- **Port Mirroring**: If you configure the service port for port mirroring, you can connect to this port via a PC and sniff (using Ethereal, WireShark, etc.) the traffic that is travelling through the other ports including the Ethernet port, the internal port, that is connected to the CPU.

NOTE: You can also configure the service port online via Control Expert using CIP explicit messaging *(see page 240)*, but this configuration may be lost when the 140NOC78000 head module is reset.

The **Service Port** page, when configured as an **Access Port**, looks like this:



The **Service Port** page, when configured for **Port Mirroring**, looks like this:



Displaying the Service Port Page

To display the **Service Port** page:

Step	Action
1	Click **Services** in the navigation tree in the left panel of the **Device Editor**. **Result**: The **Services** page opens.
2	In the **Services** page, set the **Service Port** field to **Enabled**. Then click **Apply** or **OK**. **Result**: The **Service Port** page appears in the navigation tree.
3	Click **Service Port** in the navigation tree, and set the **Service Port Mode** to one of the following: ● **Access Port** ● **Port Mirroring**
4	Click **Apply** to save changes or click **OK** to save changes and close the window.

Service Port Page Properties

The following properties can be viewed and edited in the **Service Port → Access Port** page:

Property	Description
Access Port Configuration:	
ETH 1	This port is always set to **Enabled** and cannot be edited.
Port Mirroring Configuration:	
Internal Port	These ports are disabled.
ETH 2	
ETH 3	
ETH 4	

The following properties can be viewed and edited in the **Service Port → Port Mirroring** page:

Property	Description
Access Port Configuration:	
ETH 1	This port is always set to **Enabled** and cannot be edited.
Port Mirroring Configuration:	
Internal Port	To enable these ports, click **Yes** in the **Enabled** field. To disable these ports, click **No** in the **Enabled** field.
ETH 2	
ETH 3	
ETH 4	

Configuring Time Synchronization

Introduction

The network time service (SNTP) synchronizes the clock in the 140NOC78000 distributed I/O head module to that of the time server. The synchronized value is used to update the clock in the PLC. Typical time service configurations utilize redundant servers and diverse network paths to achieve high accuracy and reliability.

Use the time service for:
- event recording (sequence events)
- event synchronization (trigger simultaneous events)
- alarm and I/O synchronization (time stamp alarms)

Time Synchronization Service Features

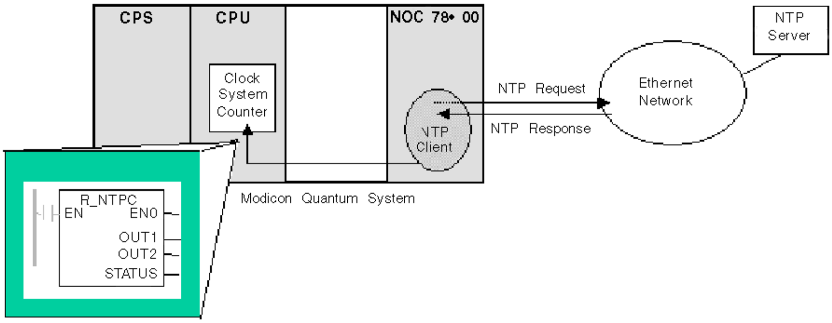
Some features of the time synchronization service are:
- periodic time correction obtained from the reference-standard time server
- automatic switch over to a backup time server if a detected problem occurs with the normal time server system
- controller projects use a function block to read the accurate clock, a feature that allows project events or variables to be time stamped
- estimated time stamping accuracy of:
  - 5 msec for 140 CPU 651 •0s and later
  - 10 msec for other CPUs
- local time zone is configurable, including daylight savings time

Time Synchronization Process

The 140NOC78•00 control head module contains an SNTP client, which provides time synchronization.

Action	Result
An SNTP client requests a time synchronization signal from an SNTP server. (The request is sent over an Ethernet network.)	The SNTP server responds with a signal.
The SNTP client stores the time.	
The SNTP client sends a message to the controller's clock system counter.	The controller updates its internal clock.
Use the R_NTPC function block in either MAST, FAST, or Interrupt sections to read the clock from the PLC application.	

On an Ethernet network, all controllers should be synchronized with the same SNTP server.



Power Up

To establish the accurate Ethernet system network time, the system performs the following at power up:

- requires the 140NOC78000 head module to boot
- uses the 140NOC78000 head module to obtain the time from the SNTP server
- requires a predefined interval until time is accurate; your configuration determines how long before time is accurate
- may require several updates to achieve peak accuracy

Once an accurate time is received, the service sets the status in the associated time service register.

The time service clock value starts at 0 until fully updated from the 140NOC78100 head module.

Model	Starting Date
Modicon Quantum with Control Expert	January 1st 1980 00:00:00.00

Stop or Run PLC

- Stop and run have no effect on the accuracy of the clock.
- Stop and run have no effect on the update of the clock.
- A transition from one mode to the other has no effect on the accuracy of the Ethernet system network time.

Download Application

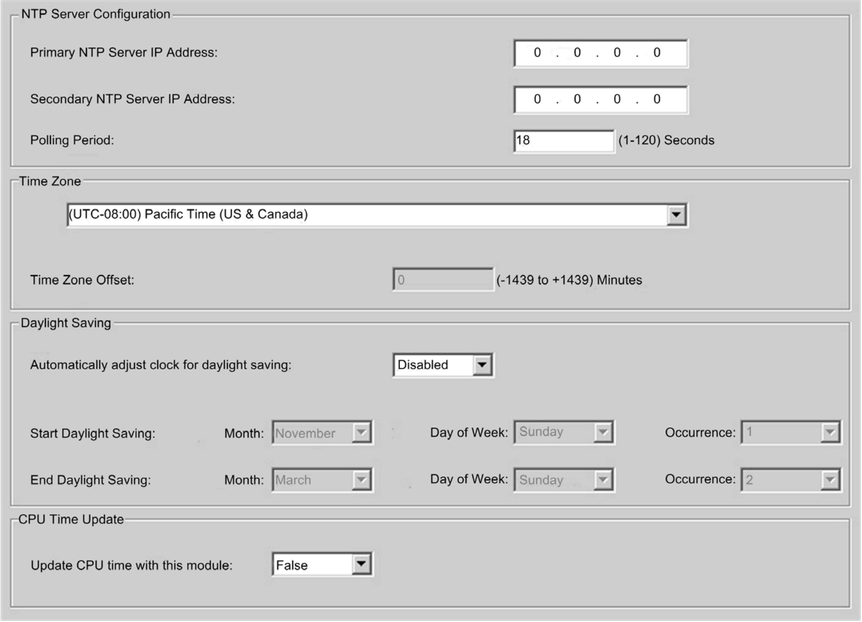
The status clock value associated with the time service register in the CPU is reinitialized after an application is downloaded or after an SNTP server swap.

There will be 2 polling periods before the time is accurate.

Configuring the Time Synchronization Service

To configure the time synchronization service:

Step	Action
1	Open the 140NOC78000 head module in the DTM browser.
2	In the **Services** page, set the **Network Time Service** field to **Enabled**.
3	Click the **Network Time Service** node in the **Services** navigation tree in the left panel.
4	Enter changes in the appropriate fields on the **Network Time Service** configuration page. The table below describes the configuration page parameters.
5	Click **Apply** to save changes, or click **OK** to save changes and close the window.



Time Synchronization Configuration Parameters

Field	Parameter	Action
NTP Server Configuration		
	Primary NTP Server IP Address	Enter a valid IP address.
	Secondary NTP Server IP Address	Enter a valid IP address.
	Polling Period	The polling period is the time (in seconds) between updates from the SNTP server. To obtain optimal accuracy (and if your network allows), reduce the polling period to a small value. The default is 5 seconds. Enter a value: ● min = 1 sec ● max = 120 sec
Time Zone		
	Time Zone	Select the desired time zone from the drop-down list. The default value is your current system's time zone (as found in Windows). You can also select **Custom Time Zone**.
	Time Zone Offset	If you selected **Custom Time Zone**, enter a value in the range of (24 hours * 60 minutes - 1) [1-minute step].
Daylight Saving		
	Automatically adjust clock for daylight saving change	**Disabled**: In the **Start Daylight Saving** and **End Daylight Saving** fields, enter the month, day of week, and occurrence range from the respective drop-down lists. **Disabled** is the default. **Enabled**: The 140 NOC 78• 00 head module automatically corrects the local time to account for daylight saving time. The **Start Daylight Saving** and **End Daylight Saving** fields are disabled because their times are automatically changed in the spring and fall every year.
	Start Daylight Saving	**Month**: January to December **Day of Week**: Sunday to Saturday **Occurrence**: 1 to 5
	End Daylight Saving	**Month**: January to December **Day of Week**: Sunday to Saturday **Occurrence**: 1 to 5
CPU Time Update		
	Update CPU time with this module	Select **True** or **False** from the drop-down list. **False** is the default.

Section 4.6
Security

Security Features

Security and HTTP, FTP, and TFTP Services

The module uses HTTP services to provide access to its embedded Web pages. The module uses FTP and TFTP services to support various features including firmware upgrades, FDR services, and Ethernet remote IO.

The module's HTTP, FTP, and TFTP services can be disabled or enabled using the **DTM Browser Security** screen.

HTTP, FTP, and TFTP services are disabled by default in DTM instances created using PlantStruxure EIO Version 1.5 and Unity Pro 8.0 or later. They are enabled by default in instances created using earlier versions of Unity Pro.

NOTE: Unity Pro is the former name of Control Expert for version 13.1 or earlier.

You can use Control Expert to enable or disable HTTP, FTP, and TFTP services as described in the following procedure.

If the HTTP, FTP, or TFTP services have been enabled with Control Expert, they can also be enabled or disabled at run time using an MBP_MSTR block with operation code FFF0 (hex) *(see page 238)*.

Using Control Expert to Enable and Disable Firmware Upgrade & FDR and Web Access Services

Perform the following steps to enable or disable FTP/TFTP or HTTP services on the module.

Step	Action
1	In the Control Expert main menu, select **Tools → DTM Browser** to open the **DTM Browser**.
2	Confirm that the DTM you want to use is not connected to the actual communication module or device. If necessary, disconnect the DTM from the module or device *(see page 90)*.
3	In the **DTM Browser**, select the module. Right-click and select **Open** to open the **Device Editor**.
4	Click the **Security** node in the navigation tree in the left panel to open the **Security** screen.
5	On the **Security** screen, choose the appropriate setting: (**Enabled** or **Disabled**) for the service or services.
6	Click: ● **Apply** to save the changes and keep the window - or - ● **OK** to save the changes and close the window

The edits will not take effect until they are successfully downloaded from your PC to the CPU and from the CPU to the communication modules and network devices.

Section 4.7
Configuring the 140 NOC 78• 00 Head Module as an EtherNet/IP Adapter

Introduction

This section describes how to configure the 140 NOC 78• 00 head module to act as an EtherNet/IP adapter, using a functionality called *local slave*. The head module supports up to 3 instances of local slaves.

In its role as a EtherNet/IP adapter, the module initiates no messages. Instead, it responds to:
- implicit messaging requests from a scanner device in the network
- explicit messaging requests—directed to the head module's assembly object *(see page 268)* — from other devices on the network

  NOTE: If no local slave instance is enabled, the head module can respond to explicit messaging requests directed at its CIP objects other than the assembly object.

What Is in This Section?

This section contains the following topics:

Introducing the Local Slave

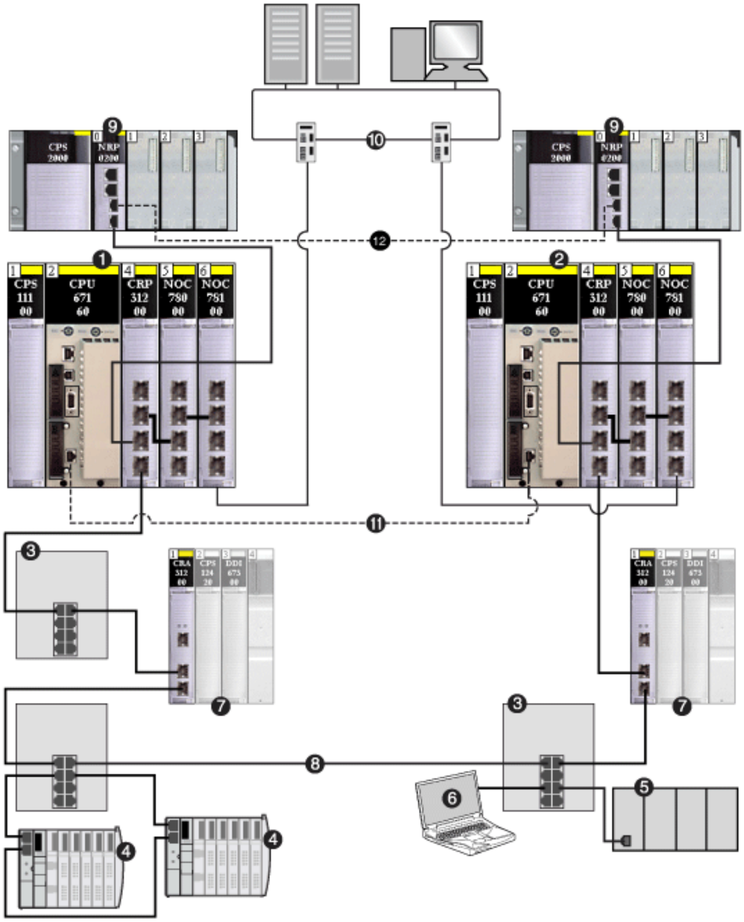
Local Slave Networking Example

The Ethernet communication module supports up to 3 instances of the local slave functionality. The local slave functionality allows other scanners on the network to read from, and write to, the Ethernet communication module using implicit messaging. Each local slave instance can accept one exclusive owner connection and one listen only connection. Through a local slave, a remote scanner can access the communication module's CIP assembly object *(see page 268)*. The local slave function is especially beneficial for peer-to-peer data exchanges at a repetitive rate.

NOTE:

- The Ethernet communication module can provide 3 local slave adapter instances, while simultaneously performing as a scanner. These roles are not mutually exclusive.
- The local slave is exclusively an EtherNet/IP functionality.

In the following example, the local slave instance is part of the following topology:



1 primary PLC (containing remote I/O, distributed I/O, and control head modules)
2 standby PLC
3 DRS connecting remote I/O and distributed I/O devices to the main ring
4 distributed I/O device (Advantys STB island)
5 third-party PLC
6 PC
7 remote I/O devices
8 main ring
9 Modicon X80 racks — connected to the PLCs via copper cable — containing BMX NRP 020• fiber converter modules to extend the distance between the 2 PLCs beyond 100 m
10 control network (connected to the primary and standby PLCs)
11 CPU-sync link (fiber cable)
12 fiber cable connecting the two BMX NRP 020• modules to extend the distance between the 2 PLCs beyond 100 m

This sample configuration includes the following devices:

- A primary PLC (1) incorporates the 140NOC78000 head module with one local slave instance enabled. The PLC scans I/O data from remote devices (4).
- A standby PLC (2) listens to the scan of the primary PLC’s local slave by the third-party PLC (5).
- DRS (3)
- An Advantys STB island (4) includes an STB NIC 2212 EtherNet/IP network interface module plus 8 I/O modules.
- A third-party scanner (5) lacks adapter capability and therefore cannot itself be scanned by the primary PLC:
  - The scanner collects data from sources that are not part of this network.
  - The scanner writes data to inputs of the primary PLC’s local slave.
  - The scanner scans the primary PLC’s local slave’s output data through an exclusive owner connection.
- A PC (6) runs this software:
  - Control Expert
  - the Control Expert configuration tool
  - Advantys configuration software

NOTE:

- Because the third-party scanner (5) and the standby scanner (2) both receive the same data from the local slave, confirm that the requested packet interval (RPI) settings of the third-party scanner’s exclusive owner connection and the standby scanner’s listen-only connection are the same.
- By enabling a local slave on the primary PLC (1):
  - The PLC (1) allows the third-party PLC (5) to write to it at a repetitive rate, even if the PLC is not capable of acting as an adapter.
  - The standby PLC (2) is able to scan the primary PLC (1) at a repetitive rate, rather than through application-intensive explicit messaging.

The following topics show you how to use Control Expert software installed in the PC (6) to configure a local slave, and to create input and output items in support of the peer-to-peer data transfers between and among scanners.

Configuring a Local Slave

Description

The Ethernet communication module presents 3 identical **Local Slave** configuration pages. Use each page to configure a separate local slave instance. Create a local slave instance by:
- enabling and naming the local slave
- specifying the size of local slave input and output assemblies
- configuring local slave variable names

To display this page, select one of the 3 **Local Slave** nodes in the navigation tree located on the left side of the **Device Editor**.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see page 97)* for instructions on how to edit properties.

The following steps describe a sample configuration for **Local Slave 1**. Your configuration may be different.

Configuration Example: Local Slave 1

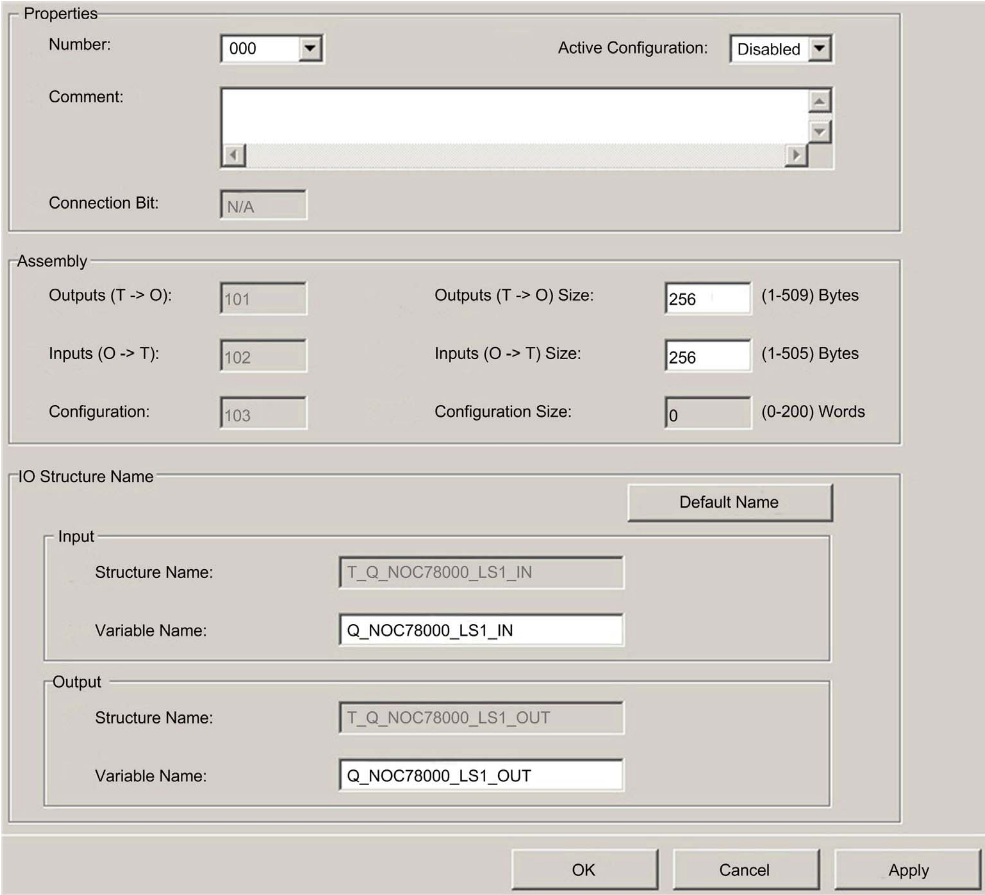
In the sample network configuration, the application in the third-party PLC produces data, which is available in the PLC's Ethernet communication module as inputs. In this example, the third-party device produces the following information:
- production totals for manufacturing line A
- production totals for manufacturing line B
- the number of production interruption events for line A
- the number of production interruption events for line B

Any information that needs to be passed to the third-party device, for example, confirmation that data from the third-party device has been received by the PLC, is accessible in the third-party device as input data. In this example, the third-party device is programmed to scan **Local Slave 1** for this confirmation.

When configuring inputs and outputs in both the local slave and the third-party PLC, associate inputs and outputs as follows:

Associate these local slave items:	with these third-party PLC items:
outputs (T -> O) — assembly instance 111	inputs — assembly instance 111
inputs (O -> T) — assembly instance 112	outputs — assembly instance 112

The configured **Local Slave** page looks like this:



Enabling and Naming the Local Slave

Use the **Properties** section of the **Local Slave** page to enable (or disable) and identify the local slave.

Setting	Description
Number	The unique number — or identifier — assigned to the device. By default, Control Expert assigns: • 000 = local slave 1 • 001 = local slave 2 • 002 = local slave 3 In this example, accept the default **000**.
Active Configuration	• **Enabled** activates the local slave. • **Disabled** de-activates the local slave, but saves the current local slave settings. In this example, select **Enabled**.
Comment	An optional free text comment field up to 80 characters maximum. In this example, leave blank.
Connection bit	Auto-generated integer (0...127) indicating the offset of the connection's: • health bit, located in the module's input area • control bit, located in the module's output area **Note:** This setting is auto-generated after the local slave settings are input and the network configuration is saved.

Configuring the Size of Local Slave Input and Output Assemblies

Use the **Assemblies** section of the **Local Slave** page to configure the size of the local slave inputs and outputs. The assembly numbers are non-editable, and are assigned by Control Expert as follows:

Assembly number	Local slave number	Used for connection
101	1	T->O [1]
102	1	O->T Exclusive Owner
103	1	Configuration
199	1	O->T Listen Only
111	2	T->O
112	2	O->T Exclusive Owner
113	2	Configuration
200	2	O->T Listen Only
1. In this table: • O indicates the originator — or scanner —device. • T indicates the target — or adapter —device.		

Assembly number	Local slave number	Used for connection
121	3	T->O
122	3	O->T Exclusive Owner
123	3	Configuration
201	3	O->T Listen Only

1. In this table:
- O indicates the originator — or scanner —device.
- T indicates the target — or adapter —device.

The **Local Slave** assembly settings include:.

Setting	Description
Outputs (T->O)	A read-only value (see table, above). In this example, **101**.
Outputs (T->O) Size	The maximum size — in bytes — reserved for local slave outputs — an integer from 1 to 509. In this example, only 2 output bytes are used: type in **2**.
Inputs (O->T)	A read-only value (see table, above). In this example, **102**.
Inputs (O->T) Size	The maximum size — in bytes — reserved for local slave inputs — an integer from 0 to 505. In this example, only 8 input bytes are used: type in **8**.
Configuration	A read-only value (see table, above). In this example, **103**.
Configuration Size	A read-only value set to **0**.

NOTE: When using explicit messaging to read the Ethernet communication module's assembly object, allocate sufficient room for the response, because the size of the response will equal the sum of:

the assembly size + Reply service (1 byte) + General Status (1 byte)

Configuring Local Slave I/O Variable Names

Each input and output that Control Expert creates for your application has both a non-editable structure name (used by Control Expert to internally identify input and output items) and an editable variable name.

Use the **I/O Structure Name** section of the **Local Slave** page to:
- view and edit local slave input and output variable names
- view non-editable local slave structure names

The following property settings have been made in this example:

Setting	Description
Input:	
Structure Name	The read-only name for input structures. By default, it is the concatenation of: • the prefix T_ • the alias device name — in this case Q_NOC78000 • the device number —in this case 01 • the suffix _IN In this case, the default would be T_Q_NOC78000_01_IN.
Variable Name	The editable base name for input variables. By default, it is the concatenation of: • the alias device name — in this case Q_NOC78000 • the device reference — in this case 01 • the suffix _IN In this case, the default would be Q_NOC78000_01_IN. For this example, accept the default variable name.
Output:	
Structure Name	The read-only name for output structures. By default, it is the concatenation of: • the prefix T_ • the alias device name — in this case Q_NOC78000_01 • the suffix _OUT In this case, the default would be T_Q_NOC78000_01_OUT.
Variable Name	The editable base name for output variables. By default, it is the concatenation of: • the alias device name — in this case Q_NOC78000 • the device number — in this case 01 • the suffix _OUT In this case, the default would be Q_NOC78000_01_OUT. For this example, accept the default variable name.

If you have edited one or more variable names, you can restore the default variable names by clicking the **Default Name** button.

Local Slave Inputs and Outputs

Introduction

The Ethernet communication module serves as an adapter when the **Active Configuration** field is set to **Enabled** in the configuration window for one (or more) of the module's local slave nodes.

When a local slave instance of an Ethernet communication module is enabled, the designated memory location allocated to that instance is exposed to, and can be accessed by, other devices.

The I/O data exchange, between the remote device and the local slave, is configured as part of the remote device's configuration settings.

Configuring the I/O Items

You can configure input and output items in groups of 1 or more single bits, 8-bit bytes, 16-bit words, 32-bit dwords, or 32-bit IEEE floating values. The number of items you create depends upon the data type and size of each item.

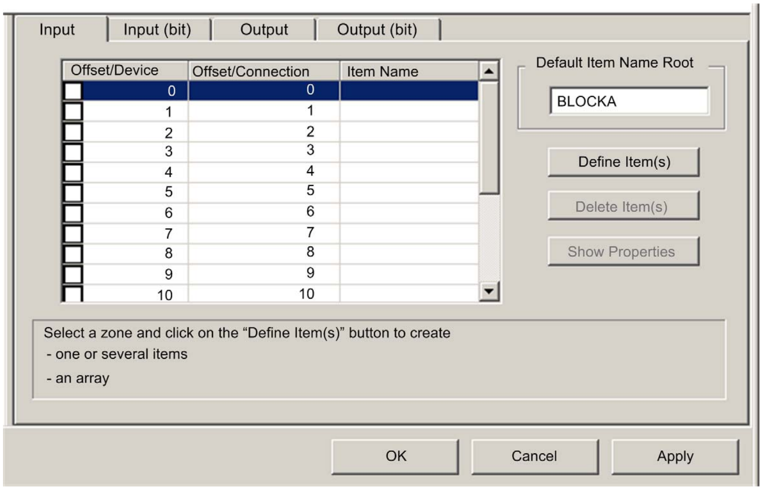
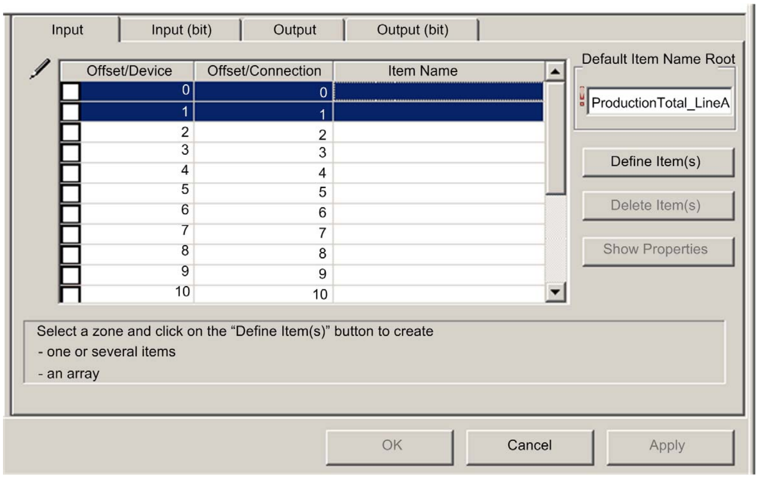
The process for creating and defining I/O items for the local slave is the same as for any adapter class device, and depends upon the type of items you wish to create.

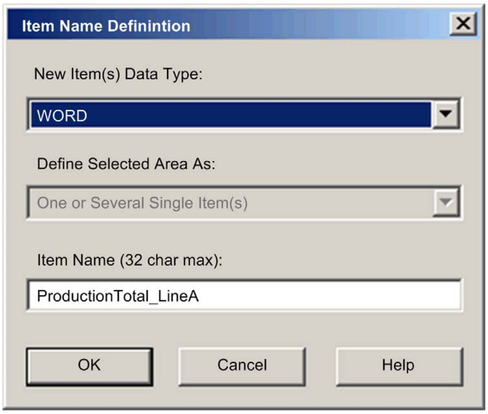
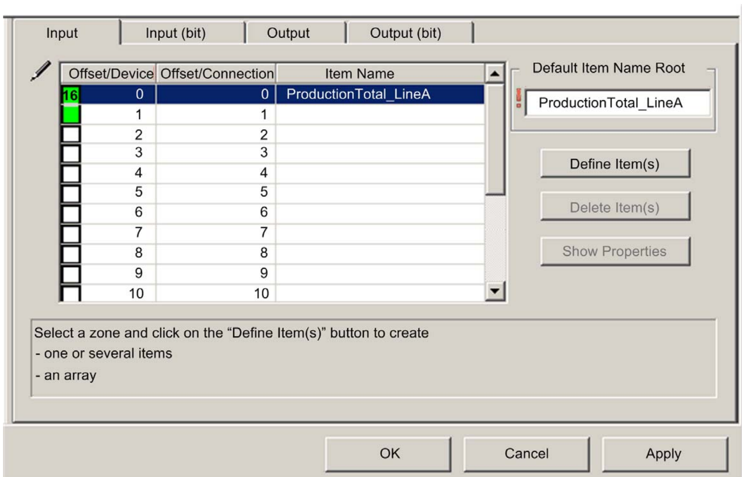
In support of the ongoing configuration example, the following items are required:
- 4 input word items
- 1 output word item

NOTE: The items created, below, are designed to hold data received from, or sent to, the third-party scanner. In addition to these items, it is necessary to include logic in the application programs in which the Ethernet communication module and the third-party scanner, respectively, are included. Writing this code is beyond the scope of this example.

Creating Input Word Items

To create input items for local slave 01:

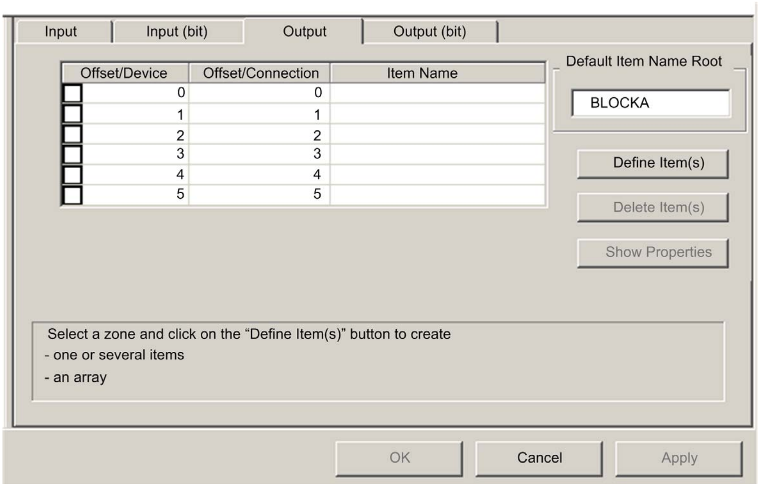
Step	Action
1	Select the **Input** tab to open that page:  **NOTE:** In this example, each row represents a byte. Because the items you create will be a 16-bit words, each item consists of 2 rows.
2	In the **Default Item Name Root** input box type: **ProductionTotal_LineA**.
3	Starting at the beginning of the table, select the first two rows: 0 and 1: 

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>Click the Define Item(s) button.
Result: The Item Name Definition dialog opens:

</td></tr>
<tr><td>5</td><td>Select WORD as the New Item(s) Data Type, then click OK.
Result: A new item is created:

</td></tr>
<tr><td>6</td><td>Click Apply to save the new items, and leave the page open.</td></tr>
<tr><td>7</td><td>Repeat steps 2 - 6 for each new word item you need to create. In this example, that includes the following items:
Rows 2-3, Default Items Name Root: ProductionTotal_LineB
Rows 4-5: Default Items Name Root: Events_LineA
Rows 6-7: Default Items Name Root: Events_LineB</td></tr>
<tr><td>8</td><td>Create output words.</td></tr>
</table>

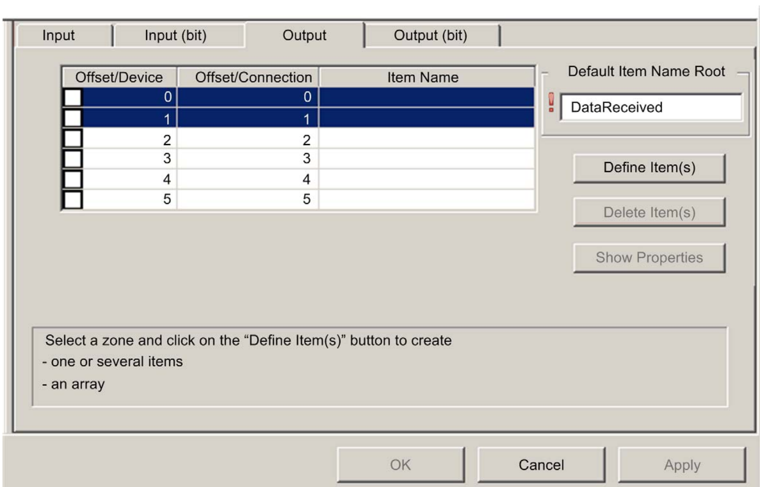
Creating Output Word Items

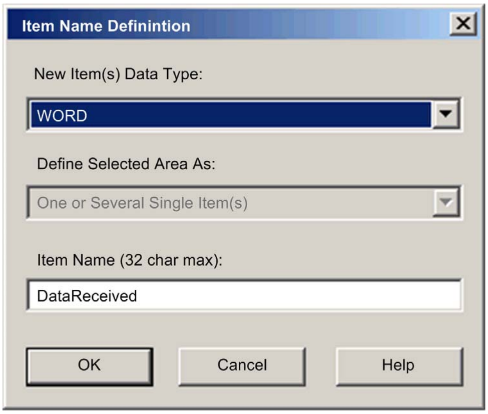
To create output items for local slave 01:

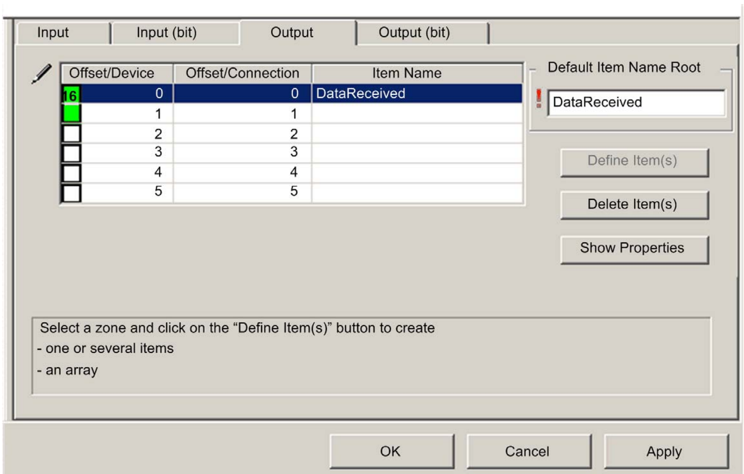
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Click the Output tab to open the following page:



NOTE: In this example, each row represents a byte. Because the only item you will create is a 16-bit word, you will select 2 rows.</td></tr>
<tr><td>2</td><td>In the Default Item Name Root input box type: DataReceived.</td></tr>
<tr><td>3</td><td>Starting at the beginning of the table, select the first 2 rows, 0 and 1:


</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>Click the Define Item(s) button.
Result: The Item Name Definition dialog opens:


</td></tr>
<tr><td>5</td><td>Select WORD as the New Item(s) Data Type, then click OK.
Result: A new item is created:


</td></tr>
<tr><td>6</td><td>Click OK to close the Items window.</td></tr>
<tr><td>7</td><td>Select File → Save to save your edits.</td></tr>
</table>

Using Local Slave Inputs and Outputs

The inputs and outputs created, above, are used as follows:

- The third-party device updates values of the following variables:
  - ProductionTotal_LineA
  - ProductionTotal_LineB
  - Events_LineA
  - Events_LineB
- The Ethernet communication module updates value of the DataReceived variable in the third-party device at the configured RPI.

Device List Configuration and Connection Summary

Introduction

The **Device List** page displays read-only properties that summarize the:

- configuration data, including:
  - input data image
  - output data image
  - maximum and actual numbers for devices, connections and packets
  - recommended PLC scan times
- Modbus requests and EtherNet/IP connection data

To display this page, first select a communication module in the **DTM Browser** then, in the left pane of the **Device Editor**, select the **Device List** node.

Configuration Data

The **Device List** page displays the following configuration data:

Name	Description	Value set by...
Input		
Input Offset	The starting address for inputs (%MW index)	**Configuration** page in Control Expert
Input Reserved Size	The total number of words configured for inputs (Max size)	**Configuration** page in Control Expert
Input Current Size	The cumulative number of inputs (Input size) actually used in the application	**General** page in the **Device Editor** for a selected remote device and connection
Output		
Output Offset	The starting address for outputs (%MW index)	**Configuration** page in Control Expert
Output Reserved Size	The total number of words configured for outputs (Max size)	**Configuration** page in Control Expert
Output Current Size	The cumulative number of outputs (Output size) actually used in the application	**General** page in the **Device Editor** for a selected remote device and connection
Note: When configuring an offset and a reserved size for both inputs and outputs, be sure that inputs and outputs do not overlap.		
Configuration Size		
Maximum Number of Devices	The maximum number of devices that can be added to the configuration.	predefined
Current Number of Devices	The number of devices currently in the configuration.	network design in the Control Expert **Device Editor**

Name	Description	Value set by...
Maximum Number of Connections	The maximum number of connections that can be managed by the module.	predefined
Current Number of Connections	The number of connections in the configuration.	network design in the Control Expert **Device Editor**
Maximum Number of Packets	The maximum number of packets per second the module is able to manage.	predefined
Current Number of Packets	The number of packet/s that will be generated by the current configuration.	network design in the Control Expert **Device Editor**
PLC Scan Time (Quantum PLCs only)		
Minimum PLC Scan Time	The estimated cycle time to process inputs and outputs, equal to the sum of estimates for communication over both the backplane and the network.	predefined
Module Exchange Time	The estimated additional time contributed by the EtherNet/IP module to perform the I/O management. This value is included in the "minimum PLC scan time" value.	predefined

Request / Connection Data

The **Device List** page displays the following request and connection data:

Name	Description
Connection Bit	The offset for both the connection’s health bit and control bit.
Device	The device **Number** as set in the **Properties** configuration page for the local slave or remote device.
Type	The target device type: ● Ethernet/IP ● Local Slave ● Modbus TCP
Address	The target device IP Address. **NOTE:** Only for remote devices. Not applicable for local slaves.
Rate	The RPI (for EtherNet/IP) or the Repetitive Rate (for Modbus TCP), in ms.
Packets / s	The number of Ethernet packets per second generated by this remote device.
Offset In	The starting %MW address for inputs to this device.
Size In	The number of input words configured for this remote device.
Offset Out	The starting %MW address for outputs from this device.
Size Out	The number of output words configured for this remote device.

Configuring Device Properties

Overview

Use the **Properties** configuration page to view and configure settings for a remote device. These settings will:

- assign a numeric address to the device
- include or exclude device inputs and outputs in the Control Expert project
- specify variable and structure names for device inputs and outputs
- determine how I/O items will be managed

To display this page, select the device name, which is found under the **Device List** node in the left pane of the **Device Editor**, then click the **Properties** tab.

NOTE: Refer to the topic Configuring Properties in the Device Editor for instructions on how to edit properties.

Device Properties

The **Properties** page includes the following settings:

Setting	Description
In the **Properties** section:	
Number	The relative position of the device in the list, from 0 to 127. By default, this number is assigned sequentially to devices in the project, beginning with the number **000** that is assigned to the first local slave.
Active Configuration	• **Enable:** adds this device to the Control Expert project configuration • **Disable:** removes this device from the Control Expert project configuration **NOTE:** • Changing this setting also changes the addresses of items in project memory. Selecting **Enable** adds the device's inputs and outputs to project memory; selecting **Disable** removes these inputs and outputs from memory. • Alternatively, if you enable the **I/O Communication Control** service, you can turn ON and OFF the connection between a communication module and a remote device by toggling the output CONTROL_BIT for that connection. This leaves unchanged the size of the project's input and output data images.
In the **IO Structure Name** section:	

Setting	Description
Input area:	
Instance Variable Name	The editable base name for input variables. By default, it is the concatenation of: ● the string DEVICE_ ● the device number ● the suffix _IN
Structure Name	The read-only name for input structures. By default, it is the concatenation of: ● The prefix T_ ● the string DEVICE_ ● the device number ● the suffix _IN
Output area:	
Instance Variable Name	The editable base name for output variables. By default, it is the concatenation of: ● the string DEVICE_ ● the device number ● the suffix _OUT
Structure Name	The read-only name for input structures. By default, it is the concatenation of: ● The prefix T_ ● the string DEVICE_ ● the device number ● the suffix _OUT
Default Name button	Restores the default variable and structure names.
In the Items Management section of the page, edit the following:	
Import mode	● **Automatic:** I/O items are taken from the device DTM and updated if the items list in the device DTM changes. Items cannot be edited in the **Device Editor**. ● **Manual:** I/O items are added when the device DTM is first added to Control Expert. Thereafter, all I/O item edits are made manually in the **Device Editor**. Changes to the device DTM do not impact the I/O items list.
Reimport Items	Imports the I/O items list from the device DTM, overwriting any manual I/O item edits. Enabled only when **Import mode** is set to **Manual**.

Configuring Device IP Address Settings

Overview

Use the pages of the **Device Editor** to view and edit IP address settings for a remote device.

To display this page, select a remote device name in the **Device List** node in the left pane of the **Device Editor**, then click on the **Address Setting** tab.

NOTE: Refer to the topic Configuring Properties in the Device Editor for instructions on how to edit properties.

Configuring the Address Setting Page

The **Address Setting** page includes the following settings:

Setting	Description
In the **Address Settings** page, edit the following:	
IP Address	By default: ● the first three octet values equal the first three octet values of the Ethernet communication module, ● the fourth octet value equals this device Number setting
DHCP for this Device	● **Enabled** activates the DHCP client in this device. The device obtains its IP address from the DHCP service provided by the Ethernet communication module and appears on the auto-generated DHCP client list. ● **Disabled** (the default) de-activates the DHCP client in this device.
Identified by	If **DHCP for this Device** is **Enabled**, this indicates the device identifier type: ● **MAC Address**, or ● **Name**
Identifier	If **DHCP for this Device** is **Enabled**, the specific device MAC Address or Name value. **NOTE:** If you use a device name, confirm that you have typed the same device name in the DTM device. Otherwise, the device will not take its IP address.
Mask	The device subnet mask. The default = 255.255.255.0.
Gateway	The gateway address used to reach this device. The default of 0.0.0.0 indicates this device is located on the same subnet as the Ethernet communication module.

Configuring Modbus TCP Request Settings

Overview

Use the **Request Setting** page to configure scanner connection information for a remote Modbus TCP device.

To display this page, select a remote Modbus TCP device in the **Device List** node in the left pane of the **Device Editor**, then click on the **Request Settings** tab.

NOTE: Refer to the topic Configuring Properties in the Device Editor for instructions on how to edit properties.

Configuring the Request Settings Page

The **Request Settings** page includes the following settings:

Setting	Description
Connection Bit	The offset for both this connection's health bit and control bit.
Unit ID	The number of the device, or module, that is the target of the connection. A value of: ● 255 (the default) used to access the Ethernet communication module itself ● 0...254 identifies the device number of the target device, behind a Modbus TCP to Modbus gateway **NOTE:** When accessing data in the Ethernet communication module itself, use 255. When accessing data in the application running in the PLC, use a value from 0 to 254 (a value of 1 is recommended).
Health Timeout	The maximum allowed period, in milliseconds, between device responses, from 0 to 120000 ms, in intervals of 5 ms. When this setting is exceeded, the health timeout bit is set to 1. Default = 1500 ms
Repetitive Rate	The rate at which data will be scanned, from 0 to 60000 ms, in intervals of 5 ms. Default = 60 ms
RD Address	Address—from 0 to 65535—in the remote device of the first word from which the communication module reads data.
RD Length	The number of words in the remote device, from 0 to 125, that the communication module will read.
Last Value	The behavior of inputs in the application in the event communication is lost: ● Hold Value (the default) ● Set To Zero

Setting	Description
WR Address	Address—from 0 to 65535—in the remote device of the first word to which the communication module writes data.
WR Length	The number of words in the remote device, from 0 to 120, that the communication module will write.

NOTE: For **RD Address** and **WR Address**, take into account the remote device address system and check you read or you write the right information, for instance Modicon Quantum addresses start from 1 whereas Modicon Premium addresses start from 0.

Configuring Communication Module Connection Settings

Overview

Control Expert automatically creates a connection between the communication module and a remote device when the remote device is added to the Control Expert project. Properties defining each connection must be configured in the DTMs for both the connection module and the remote device *(see page 185)*.

Use the **Connection Settings** page to view and edit connection properties from the perspective of the communication module.

To open this page, in the left pane of the **Device Editor** expand the navigation tree and, under the **Device List** node, select **<remote device>** → **<connection>**, where:

- **<remote device>** represents the name of the selected remote device appearing in the **Device List**, and
- **<connection>** represents the name of the selected connection, which depends upon the types of connections supported by the remote device and the particular connection type selected in the connection configuration settings for the remote device DTM.

NOTE: The name of the connection displayed in the **Device List** depends upon the types of connections supported by the remote device and the particular connection type selected in the connection configuration settings for the remote device DTM.

Refer to the topic Configuring Properties in the Device Editor for instructions on how to edit properties

Communication Module Connection Properties

The following connection settings for this sample configuration can be viewed or configured in the DTM for the communication module:

Setting	Description
Connection Bit	(Read-only) The system generated number, from 0 to 256, for the health bit for this connection. **NOTE:** The first numbered Connection Bit is 0, which maps to: ● HEALTH_BITS_IN[0].0: the first bit in the first byte of the zero-based HEALTH_BITS_IN byte array, and ● CONTROL_BITS_OUT[0].0: the first bit in the first byte of the zero-based CONTROL_BITS_OUT byte array,
Request Packet Interval (RPI)	The refresh period, from 2 to 65535 ms, for this connection. Default = 12 ms **NOTE:** This parameter can be set in the DTM for the communication module, or in the DTM for the remote device.
Time-out Multiplier	This setting, multiplied against the RPI, produces a value that triggers an inactivity timeout. Setting selections include: x4, x8, x16, x32, x64, x128, x256 and x512. Default = x4 **NOTE:** To view the **Time-out Multiplier** parameter, Control Expert must be operating in **Advanced Mode**.

EtherNet/IP Connection Information

Overview

Use this read-only page to view connection properties for the remote device. An EtherNet/IP connection provides a communication link between two or more devices. Properties for a single connection are configured in the DTMs for each of the connected devices—typically a communication module and a remote device. The read-only properties viewable in this page can be configured in the General page of the connection node configuration page for the DTM of the remote device *(see page 185)*.

To open this page, in the left pane of the **Device Editor** expand the navigation tree and, under the **Device List** node, select **<remote device>** → **<connection>**, where:

- **<remote device>** represents the name of the selected remote device appearing in the **Device List**, and
- **<connection>** represents the name of the selected connection, which depends upon the types of connections supported by the remote device and the particular connection type selected in the connection configuration settings for the remote device DTM.

Remote Device Connection Properties

A connection to a remote Schneider Electric device can present the following properties:

Setting	Description
RPI	The refresh period for this connection, in milliseconds.
Input size	The number of bytes reserved for input data, from 0 to 505.
Input mode	The transmission type: ● Multicast ● Point to Point
Input type	Ethernet packet type—fixed or variable length—to be transmitted. **NOTE:** The Ethernet communication module supports only **Fixed** length packets.
Input priority	The transmission priority. The value depends upon the device DTM. Values can include: ● Low ● High ● Scheduled
Input trigger	The transmission trigger. Values can include: ● Cyclic ● Change of state or application
Output size	The number of bytes reserved for output data, from 0 to 509.
Output mode	The transmission type: ● Multicast ● Point to Point

Setting	Description
Output type	Ethernet packet type—fixed or variable length—to be transmitted. **NOTE:** The Ethernet communication module supports only **Fixed** length packets.
Output priority	The transmission priority. The value depends upon the device DTM. Values can include: • Low • High • Scheduled

Chapter 5
Configuring Remote Devices

Overview

This chapter describes how to use the Control Expert **Device Editor** to configure remote device properties, including properties that define:

- the connection between the remote device and the communication module
- the degree to which the actual remote device must match the remote device described in the Control Expert project configuration
- other settings required by the remote device manufacturer that are unique to the remote device

NOTE:

- Before connecting with remote devices, check whether the module's FTP/TFTP services are enabled, and if they are not, enable them *(see page 154)*.
- Before you can configure settings for a device, the device DTM must be disconnected from the remote device itself. If necessary, you can disconnect a DTM by selecting its node in the **DTM Browser** then selecting **Edit** → **Disconnect**.
- The DTM for a device determines its configuration template. The topics presented in this chapter described configuration settings that apply to devices manufactured by Schneider Electric and to non-manufacturer-specific generic devices.

What Is in This Chapter?

This chapter contains the following topics:

Displaying Remote Device and DTM Properties

Introduction

Use this page to view properties that describe:

- the remote device, and
- its DTM

To display this page, select a remote device in the **DTM Browser** to open its DTM. Then, in the left pane of the **Device Editor**, select the node that displays the assigned device name.

NOTE: When this page is displayed, if this device is capable of supporting an additional connection, you can use the **Add Connection** command to create a new connection for this device *(see page 183)*.

Properties

The properties displayed in this page are read-only and are determined by the manufacturer of the remote device. The source of the displayed property values is the device DTM. The following list presents an example of the self-explanatory properties you may see displayed for a Schneider Electric device:

- File Name
- File:
  - Description
  - File Creation Date
  - File Creation Time
  - Last Modification Date
  - Last Modification Time
  - EDS Revision
- Device:
  - Vendor Name
  - Device Type
  - Major Revision
  - Minor Revision
  - Product Name
  - Catalog Number

Adding and Removing Connections

Introduction

Connections are created and configured in the DTM for a remote device.

Use the **Device Editor** to access the DTM for a remote device, where you can add and remove connections between the remote device and the Ethernet communication module.

Adding a Connection

To add a connection between a remote device and the communication module:

Step	Action
1	In the **DTM Browser**, double-click a remote device. Its DTM opens in the **Device Editor**.
2	In the left pane of the **Device Editor**, select the node displaying the name of the remote device. **NOTE:** ● If the device is capable of supporting additional connections, the **Add Connection** button becomes enabled. ● If the **Add Connection** button remains disabled, the device is presently supporting its maximum number of connections. In this case, a new connection can be added only after an existing connection is removed.
3	Click the **Add Connection** button. The **Select the connection to add** dialog opens.
4	In the **Connection to add** lit, select a connection type. **NOTE:** The types of connections available in the list depends upon the connection types supported by the specific remote device.
5	Click **OK** to close the dialog. The new connection appears in the tree control in the left pane.
6	Click on the following tabbed pages, and configure the properties in each page (as necessary): ● General *(see page 185)* ● Identity Check *(see page 187)* ● Configuration Settings *(see page 189)*
7	Do one of the following: ● click **Apply** to save your edits and leave the window open, or ● click **OK** to save your edits and close the window

Removing a Connection

To remove a connection between a remote device and the communication module:

Step	Action
1	In the **DTM Browser**, double-click a remote device. Its DTM opens in the **Device Editor**.
2	In the left pane of the **Device Editor**, beneath the remote device name, select the connection node you wish to remove.
3	Click the **Remove Connection** button. The dialog opens. The connection disappears from the tree control.
4	Do one of the following: ● click **Apply** to save your edits and leave the window open, or ● click **OK** to save your edits and close the window

Configuring EtherNet/IP Connections

Overview

Use this page to configure connection properties that are required by the remote device DTM. An EtherNet/IP connection provides a communication link between two or more devices. Properties for a single connection must be configured in the DTMs for each of the connected devices (typically a communication module and a remote device).

Open this page:

Step	Action
1	Double-click on the remote device in the **DTM Browser** to open its DTM in the **Device Editor**.
2	In the navigation tree in the left pane of the **Device Editor**, select the connection node you want to configure.
3	In the right pane of the **Device Editor**, click the **General** tab.
4	Refer to the topic Configuring Properties in the Device Editor for instructions on how to edit properties.

NOTE: When this page is open, you can use the **Remove Connection** command to delete the selected connection.

Remote Device Connection Properties

A connection to a remote Schneider Electric device can present these properties:

Property	Description
RPI	RPI indicates the refresh period for this connection in milliseconds. (This parameter can also be set in the DTM for the communication module device.)
Input size	This is the number of bytes (0 ... 505) that are reserved for input data.
Input mode	This mode is the input transmission type: ● Multicast ● Point to Point
Input type (read only)	This is the Ethernet packet type (fixed or variable length) for transmission. **NOTE:** The Ethernet communication module supports only **Fixed** length packets.
Input priority	This transmission priority value depends upon the device DTM. These are the available values: ● Low ● High ● Scheduled
Input trigger	These are the available values for the transmission trigger: ● Cyclic ● Change of state or application

Property	Description
Output size	This is the number of bytes (0 ... 509) that are reserved for output data.
Output mode	This mode is the output transmission type: ● Multicast ● Point to Point
Output type (read only)	This is the Ethernet packet type (fixed or variable length) for transmission. **NOTE:** The Ethernet communication module supports only **Fixed** length packets.
Output priority	This transmission priority value depends upon the device DTM. These are the available values: ● Low ● High ● Scheduled

Checking Remote Device Identity

Overview

Use this page to specify the degree to which a remote device (detected on the network) conforms to the configuration settings for the same remote device in the Control Expert application project. Control Expert does not maintain connections to a remote device that does not pass this identity check.

Open this page:

Step	Action
1	Double-click on the remote device in the **DTM Browser** to open its DTM in the **Device Editor**.
2	In the navigation tree in the left pane of the **Device Editor** select the connection node you want to configure.
3	In the right pane of the **Device Editor**, click the **Identity Check** tab.
4	Refer to the topic Configuring Properties in the Device Editor for instructions on how to edit properties.

NOTE: When this page is open, you can use the **Remove Connection** command to delete the selected connection.

Remote Device Identity Properties

A connection to a remote Schneider Electric device can present these properties:

<table>
<tr><th>Property</th><th>Description</th></tr>
<tr><td>Check Identity</td><td>This property defines the rule that Control Expert uses to compare the configured versus the actual remote device. These are the available settings:
• Must match exactly: The DTM or EDS file exactly matches the remote device.
• Disable: The checking function does not run. The identity portion of the connection is filled with zero values (the default setting).
• Must be compatible: When the remote device is not the same as defined by the DTM/EDS, it emulates the DTM/EDS definitions.
• None—no checking occurs; the identity portion of the connection is omitted
• Custom: Enable the following parameter settings individually.</td></tr>
<tr><td colspan="2">When Check identity is set to Custom, complete these fields:</td></tr>
<tr><td>Compatibility Mode</td><td>• True: For each of the following selected tests, the DTM/EDS and remote device are compatible.
• False: For each of the following selected tests, the DTM/EDS and remote device match exactly.</td></tr>
<tr><td>Minor Version</td><td rowspan="5">For each of these, select a setting:
• Compatible: Include the parameter in the test.
• Not checked: Do not include the parameter in the test.</td></tr>
<tr><td>Major Version</td></tr>
<tr><td>Product Code</td></tr>
<tr><td>Product Type</td></tr>
<tr><td>Product Vendor</td></tr>
</table>

Configuration Settings

Introduction

Use the **Configuration Settings** page to complete the configuration of the connection to this remote device. The information added in this page extends the address path to the remote device.

To open this page:

Step	Action
1	Double-click on the remote device in the **DTM Browser** to open its DTM in the **Device Editor**.
2	In the navigation tree in the left pane of the **Device Editor** select the connection node you want to configure.
3	In the right pane of the **Device Editor**, click the **Configuration Settings** tab.
4	Refer to the topic Configuring Properties in the Device Editor for instructions on how to edit properties.

NOTE: When this page is open, you can use the **Remove Connection** command to delete the selected connection.

Configuration Settings

The content of this page can vary, depending upon the DTM—selected in the **Add** dialog—that defines this device. Examples of DTM properties that may be configured in this page include:

This DTM type...	Can require this content...	
	Property	Description
Generic Device	Configuration[1]:	A hexidecimal extension to the addressing path.
Advanced Generic Device	Input Instance[1]:	The device specific assembly number associated with input (T -> O) transmissions.
	Output Instance[1]:	The device specific assembly number associated with output (O -> T) transmissions.
	Configuration Instance[1]:	The device specific assembly number associated with device configuration settings.
	Configuration[1]:	A hexidecimal extension to the addressing path.
Device with EDS	(The list of properties is defined by, and varies with, each specific DTM.)	
1. The value, or range of values, that can be used to configure this property must be obtained from the manufacturer of the specific device and device DTM.		

Configuring Modular Devices

Introduction

The **Chassis/Modules** page applies only to modular devices—i.e., remote devices that combine a network interface module, chassis, and input/output modules. Use the chassis page to configure the chassis by:

- selecting a chassis type, and specifying the number of chassis slots
- inserting one or more modules into the chassis
- removing a module from the chassis
- moving a module to a different position in the chassis

You can add any chassis—and any module suitable for a selected chassis—that appears in the **Device Library**.

To open this page:

Step	Action
1	Double-click on the remote device in the **DTM Browser** to open its DTM in the **Device Editor**.
2	In the navigation tree in the left pane of the **Device Editor** select the **Chassis/Modules** node.
3	Refer to the topic Configuring Properties in the Device Editor for instructions on how to edit properties.

Configuring the Chassis

To configure the chassis for a modular device:

Step	Action
1	Select a chassis type in the **Chassis Type Available in the Device Library** list. The selected number of slots appear in the **Configured Modules** list beneath the selected remote adapter.
2	Select a module in the **Available Modules for the Chassis** list.
3	Click the button to insert the selected module into the first available (i.e., lowest numbered) open slot.
4	Use the following buttons to move a module within the chassis: moves the selected module up to the next available slot moves the selected module down to the next available slot deletes the selected module from the chassis
5	Repeat steps 2 through 4 for all modules you want to add to the chassis.
6	Do one of the following: ● Click **Apply** to save your edits and leave the page open ● Click **Save** to save your edits and close the page

Chapter 6
Online Action

Overview

This chapter describes online actions you can undertake in Control Expert. Depending on the type and protocol of the selected communication module or remote device, you can perform these tasks:

- Display CIP objects.
- View and edit port configuration parameters.
- Ping a module or device to confirm that it is active on the Ethernet network.
- Connect to a remote device and ...
  - View device default parameter settings.
  - View the current parameter settings for the device.
  - Edit the parameter settings for the device.

NOTE: Before you can perform online actions for a communication module or remote device, connect its DTM to the physical module or device. (That is, select the module or device node in the **DTM Browser** and select **Edit** → **Connect**.)

What Is in This Chapter?

This chapter contains the following topics:

Online Action - EtherNet/IP Object

Overview

Use the **EtherNet/IP Object** page of the **Online Action** window to perform these tasks:

- Retrieve and display the data that describes the current state of CIP objects for the selected communication module or remote device.
- Reset the selected communication module or remote device.

NOTE: Before you can perform online actions for a communication module or remote device, connect its DTM to the physical module or device. (That is, select the module or device node in the **DTM Browser** and select **Edit** → **Connect**.)

Choose an operating mode in Control Expert to select the CIP object information that this page displays:

Mode	CIP Objects
Standard mode	• Identity object
Advanced mode	• Identity object • Connection Manager object • TCP/IP Interface object • Ethernet Link object • QoS object

Retrieve and Display EtherNet/IP Object Data

Display CIP object data for a communications module or remote device:

Step	Action
1	Select a communication module in the **DTM Browser**.
2	Open the **Online Action** window. (Right-click the pop-up menu and scroll to **Device menu** → **Online Action**.)
3	Select a communication module or device in the left pane of the **Online Action** window.
4	In the right pane, click the **EtherNet/IP Object** tab to open that page.
5	Observe these requirements for the selected operating mode in Control Expert. **Advanced Mode**: Select a CIP object: • Identity • Connection Manager • TCP/IP • Ethernet Link • QoS **Standard Mode**: Control Expert displays data only for the CIP Identity object.
6	Click the **Refresh** button.

Reset a Communication Module or Remote Device

Reset a communications module or remote device:

Step	Action
1	Select a communication module in the **DTM Browser**.
2	Open the **Online Action** window. (Right-click the pop-up menu and scroll to **Device menu → Online Action**.)
3	In the left pane of the **Online Action** window, select a communications module or device.
4	In the right pane, click on the **EtherNet/IP Object** tab to open that page.
5	Click the **Reset Device** button.

Online Action - Port Configuration

Overview

Use the **Port Configuration** page of the **Online Action** window to view and edit communications port properties for a remote device. Specifically, you can use this page to perform these tasks:
- Get port configuration settings from a remote EtherNet/IP device.
- Use a **Set** command to write edited values to the same remote EtherNet/IP device.

Configuration edits transmitted from this page are sent as EtherNet/IP explicit messages and employ the **Address** and **Messaging** settings configured in the **EtherNet/IP Explicit Messaging** window.

NOTE: Before you can perform online actions for a remote device, connect its DTM to the physical device. (That is, select the device node in the **DTM Browser** and select **Edit** → **Connect**.)

Get Port Configuration Settings

To get settings from a remote EtherNet/IP device on the network:

Step	Action
1	In the **DTM Browser**, select the communication module upstream of the remote EtherNet/IP device.
2	Click the right mouse button, and in the pop-up menu select **Device menu** → **EtherNet/IP Explicit Message**. The **EtherNet/IP Explicit Message** window opens.
3	In the EtherNet/IP Explicit Messaging page, complete the **Address** section. **Note**: Port configuration explicit messages are sent as unconnected messages.
4	Return to the **DTM Browser** and again select the communication module upstream of the remote EtherNet/IP device.
5	Click the right mouse button, and in the pop-up menu select **Device menu** → **Online Action**. The **Online Action** window opens.
6	In the left pane of the **Online Action** window, select a remote EtherNet/IP device.
7	In the right pane, click on the **Port Configuration** tab to open that page.
8	If the remote device consists of more than one port, select the port number in the **Physical Interface Instance** list.
9	In the **Port Configuration** page, click the **Get Values from Device** button. The table displays the returned values of the communication properties for the selected remote device and port.

Edit and Set Port Configuration Settings

To edit and set port configuration settings that were retrieved using the above-described **Get Port Configuration Settings** process:

Step	Action
1	Double-click the left mouse button in the **Value** cell for the parameter you want to edit. The cell becomes editable. **Note:** The page also displays a **Description** of the selected parameter.
2	Type in, or select, the new value.
3	Repeat the above steps for each parameter you want to edit.
4	Perform one of these tasks: ● Click the **Set All Values to Device** to write all values to the remote device. ● If you edited parameters for only one part or group of the collection of remote device values, perform these steps: **1** In the **Set Part of Values** area, select one property group. **2** Click the **Set Values to Device** button. Control Expert sends the property value edits to the remote device via an EtherNet/IP explicit message, and displays the results in the **Description** area.

Online Action - Ping

Overview

Use the **EtherNet/IP Object** page of the **Online Action** window to send an ICMP echo request to a target communication module or remote device to determine:

- if the target device is present, and if so
- the elapsed time to receive an echo response from the target device

The target device is identified by its IP address setting.

NOTE: Before you can perform online actions, connect the DTM for the communication module or remote device to the module or device itself. To do this, select the module or device node in the **DTM Browser**, then select **Edit** → **Connect**.

Pinging a Network Device

To ping a network device:

Step	Action
1	In the **DTM Browser**, select a communication module.
2	Right-click the pop-up menu and select **Device menu** → **Online Action**. **Result**: The **Online Action** window opens.
3	In the left pane of the **Online Action** window, select a communication module or device.
4	In the right pane, click the **Ping** tab to open that page. **NOTE:** The read-only **IP Address** of the selected module or device is pre-selected.
5	To send... ● a single ping, de-select **Repeat**. ● a series of pings (1 every 100 ms), select **Repeat**.
6	(**Optional**) Select **Stop on Error** to stop pinging if an error is detected.
7	Click **Ping** once to begin pinging. The result of the ping is displayed in the **Ping Result** area. Click **Clear** to empty the **Ping Result** contents
8	Click **Ping** a second time to stop looped pinging, where no error has been detected.

Viewing and Editing Online Settings for a Remote Device

About Online Parameters

Use the **Online Parameters** window to perform these tasks:
- View the remote device's default parameter settings.
- View the remote device's current parameter settings.
- Edit and download to the remote device its editable parameter settings.

Parameter setting edits that are transmitted from this page are sent as EtherNet/IP explicit messages. These edits employ the **Address** and **Messaging** settings configured in the **EtherNet/IP Explicit Messaging** window.

NOTE: Before you can view and edit online settings for a remote device, connect its DTM to the physical device. (That is, select the device node in the **DTM Browser** and select **Edit → Connect**.)

Online Parameters Window

Open the **Online Parameters** window:

Step	Action
1	Select the node for a remote device in the **DTM Browser**.
2	Right-click in the pop-up menu and scroll to **Device menu → Online Parameters**. The **Online Parameters** window opens for the selected remote device.
3	In the left pane of the **Online Parameters** window, select a connection node. Control Expert displays the parameters relating to the selected connection in the right pane. **NOTE:** The list of parameters displayed in the **Online Parameters** window depends upon the the device that is selected in the **DTM Browser** and the connection that is selected in the left pane of the **Online Parameters** window.

Read-only parameters are identified by a locked icon .

Editable parameters are identified by a blue arrowhead .

Displaying Default Parameter Settings

Click the **Get Values from EDS** button to view the default parameter settings for the remote device. Control Expert reads the default device values from its EDS file and displays them on-screen.

Displaying Online Parameter Settings

View the current parameter settings for the remote device:

Step	Action
1	Open the **Synchronize Action** dialog box. (Click the **Synchronize** button while a connection is selected in the left pane.)
2	In the message box, select **Read values from the device**.
3	Click **OK**. The message box closes. Look at the **Online Parameters** window: ● The **Status** field displays the results of the read transaction. ● The parameter list displays the current values.

Editing Online Parameter Settings

Edit parameter settings for the remote device:

Step	Action
1	With a connection selected in the left pane, display one of these settings: ● default device settings ● current device settings
2	In the **Value** column, type in or select a new value for each setting that you want to edit. **NOTE:** When you select a parameter, the **Description** area displays an explanation of the parameter and its available settings.
3	Open the **Synchronize Action** dialog box. (Click the **Synchronize** button.)
4	In the message box, select **Write data to the device**.
5	Click **OK**. The message box closes. In the **Online Parameters** window, the **Status** field displays the results of the write transaction.

Get and Set Rack Size

Introduction

Use the rack size page to set the chassis (rack) size through a direct online communication from the Control Expert Ethernet Configuration Tool software to the remote device.

NOTE: For some modular remote devices, it is possible for the actual rack size to differ from the configured rack size. In this case, use the controls in this page to synchronize your application program with the device configuration.

Transmissions made in this page are sent as EtherNet/IP explicit messages and employ the **Address** and **Messaging** settings configured in the **EtherNet/IP Explicit Messaging** window.

NOTE: Before you can get or set rack size data in this page, connect the device DTM to the physical device. (That is, select the device node in the **DTM Browser** and select **Edit** → **Connect**.)

Rack Size Page

Open the **Get/Set Rack Size** page:

Step	Action
1	Select the remote device node in the **DTM Browser**.
2	Click the right mouse button, then in the pop-up menu select **Device menu** → **Get/Set Rack Size**. The **Get/Set Rack Size** window opens.

Get Rack Size

Click the **Get Rack Size** button to obtain the actual configured rack size from the remote modular device.

The actual rack size is displayed as a read-only value in the text box to the left of the **Get/Rack Size** button. The **Status** field displays the result of the explicit messaging transaction.

Set Rack Size

Write a new rack size setting to the remote modular device:

Step	Action
1	In the editable text box to the left of the **Set Rack Size** button, type in the desired rack size.
2	Click the **Set Rack Size** button. The remote modular device is re-configured with the new rack size. **NOTE:** The **Status** field displays the result of the explicit messaging transaction.

Chapter 7
Working With Derived Data Types

Overview

This chapter describes how to complete your project by creating, updating, and viewing derived data type (DDT) variables in Control Expert.

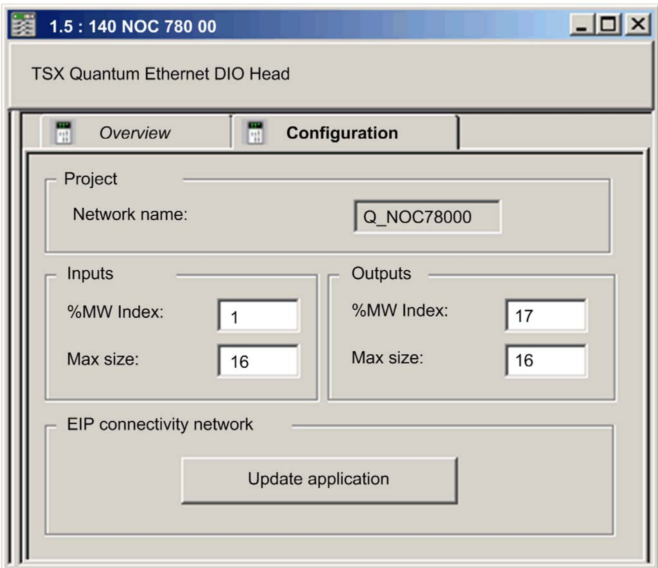
What Is in This Chapter?

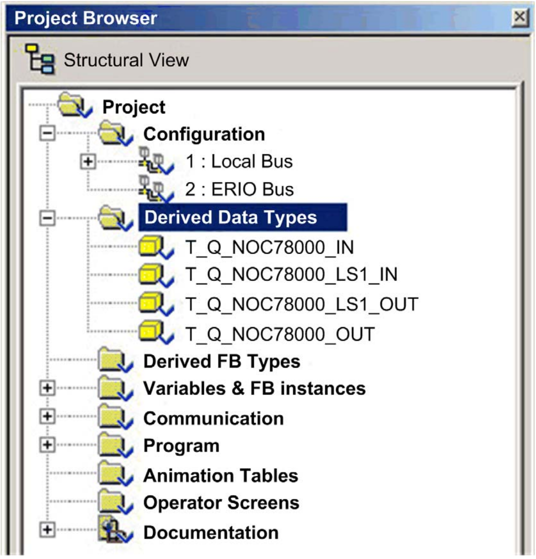
This chapter contains the following topics:

Creating and Updating Derived Data Types

Creating or Updating Derived Data Types

After you have completed your edits in the **Device Editor**, the next step is to let Control Expert create the necessary program objects — in the form of derived data types (DDTs) and variables — that will support your network design. To do this, follow these steps:

Step	Action
1	In the **Project Browser**, navigate to and select the communication module.
2	Do one of the following: ● Click the right mouse button, and select **Open** in the pop-up menu — or — ● In the **Edit** menu, select **Open**. **Result**: The **Configuration** page of the Ethernet communication module opens: 
3	Click the **Update application** button. **NOTE:** ● Every time you use the **Device Editor** to make changes to your Control Expert project, return to this screen and click the **Update application** button to save your edits. ● Control Expert refreshes the collection of DDTs and variables — by adding, editing, or deleting previously generated DDTs and variables — each subsequent time you click the **Update application** button.

Step	Action
4	Click **OK**. **Result**: The **Project Browser** displays the new or edited derived data types: 

Working with Derived Data Type Variables

Derived Data Type Variables

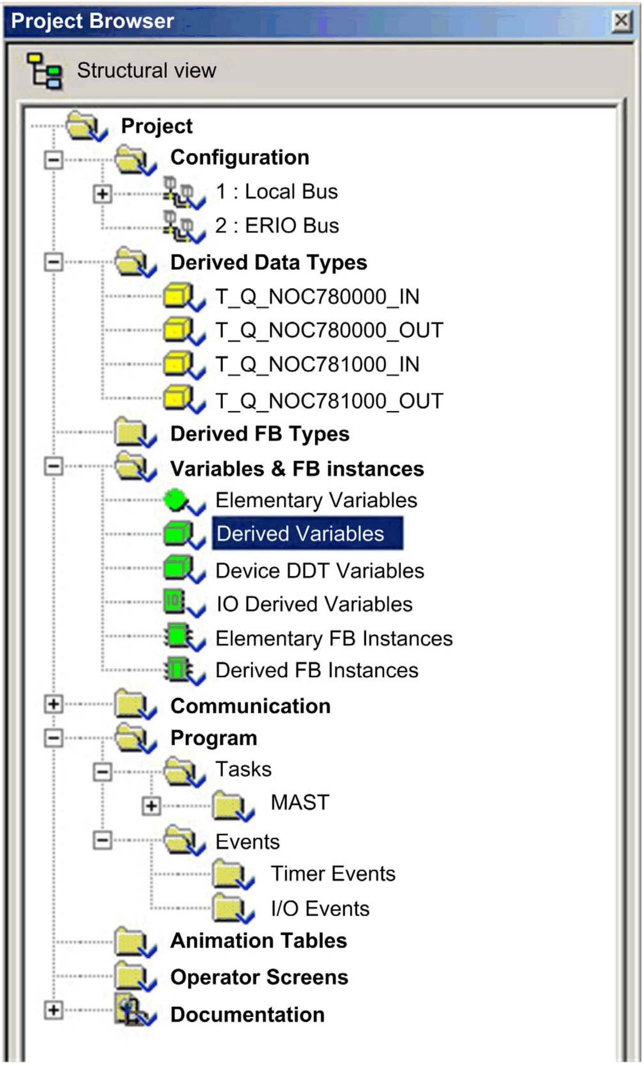
When you click the **Update application** button, Control Expert creates a collection of derived data types and variables. These are used by Control Expert to support communication and data transfer between the PLC and the various local slaves, remote devices, and their I/O items. You can access these derived data types and variables in the Control Expert **Data Editor** and add them to a user-defined **Animation Table**, where you can monitor read-only variables and edit read-write variables.

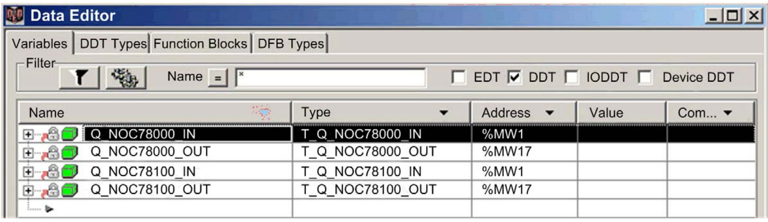
Use these data types and variables to:

- view the status of all connections from the communication module to remote EtherNet/IP and Modbus TCP devices, where:
  - the status of all connections is displayed in the form of a HEALTH_BITS array consisting of 32 bytes
  - each connection is represented by a single bit in the array
  - a bit value of 1 indicates the connection is healthy
  - a bit value of 0 indicates the connection is lost, or the communication module can no longer communicate with the remote device
- toggle a connection ON (1) or OFF (0) by writing to a selected bit in a 32 byte CONTROL_BITS array
  NOTE: Be alert to the distinction between toggling a bit in the CONTROL_BITS array on or off versus enabling or disabling a remote device.
- monitor the value of local slave and remote device input and output items you created in the Control Expert **Device Editor**

Identifying Derived Variables in the Data Editor

To view your Control Expert application's derived data type variables:

Step	Description
1	In the **Project Browser**, navigate to and double-click the left mouse button on **Variables & FB instances** → **Derived Variables**: 

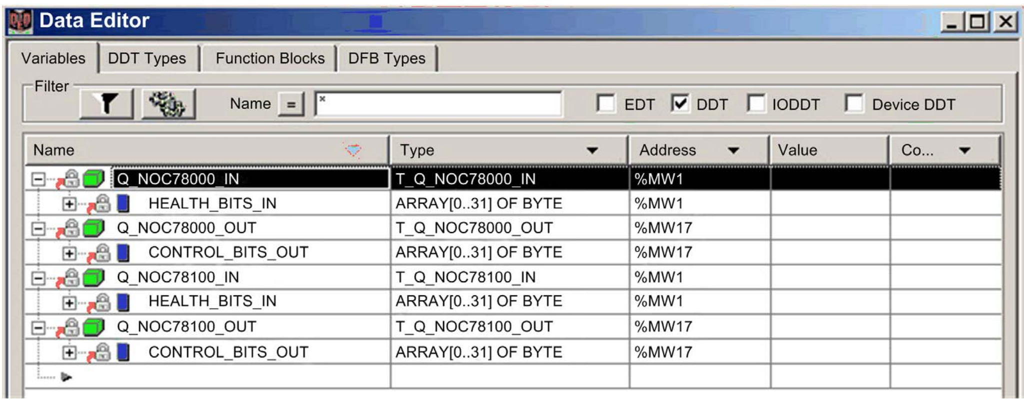
<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>2</td><td>The **Data Editor** opens, displaying the **Variables** page:


NOTE:

A check mark appears in the DDT checkbox. (If not, select the DDT checkbox to display these variables.)
The red arrow and lock icons indicate the variable name was auto-generated by Control Expert based on the configuration of the local slave or remote device and cannot be edited.
</td></tr>
</table>

Displaying the Order of Input and Output Items in PLC Memory

The **Data Editor** displays the address of each input and output variable. Click the **Address** column header to sort input and output addresses in ascending order. When you open the first input and output variables, you can see both the connection health bits and the connection control bits:



Refer to the order of inputs and outputs in the above example. Recall that the user defines the size and location of inputs and outputs. However, within the reserved area for both inputs and outputs, Control Expert assigns addresses to variables in the following order:

<table>
<tr><th>Inputs</th><th>Order</th><th>Outputs</th></tr>
<tr><td>Health bits [1]</td><td>1</td><td>Control bits [1]</td></tr>
<tr><td>Modbus TCP input variables [2]</td><td>2</td><td>Modbus TCP output variables [2]</td></tr>
<tr><td>Local Slave input variables [3]</td><td>3</td><td>Local Slave output variables [3]</td></tr>
<tr><td>EtherNet/IP input variables [2]</td><td>4</td><td>EtherNet/IP output variables [2]</td></tr>
<tr><td colspan="3">1. Health and control bits are sub-ordered as follows:
i. by device type: a. Modbus TCP; b. local slave; c. EtherNet/IP
ii. within each device type:
a. by device or local slave number
b. within a device: by connection number
2. Device variables are sub-ordered as follows:
i. by device number
ii. within a device: by connection number
iii. within a connection: by item offset
3. Local slave variables are sub-ordered as follows:
i. by local slave number
ii. within each local slave: by item offset</td></tr>
</table>

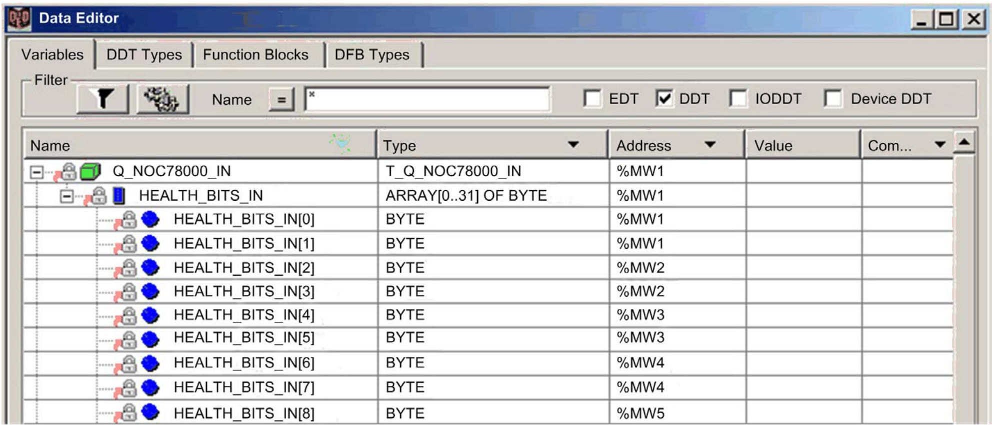
NOTE: When a device is added to or removed from the project, or when the active status of an existing device or a local slave changes, the specific location of inputs and outputs in PLC memory also changes.

Identifying the Connection Health Bits

The Ethernet communication module can support up to 128 connections to remote devices. The health of each connection is represented in a single bit value. A health bit value of:

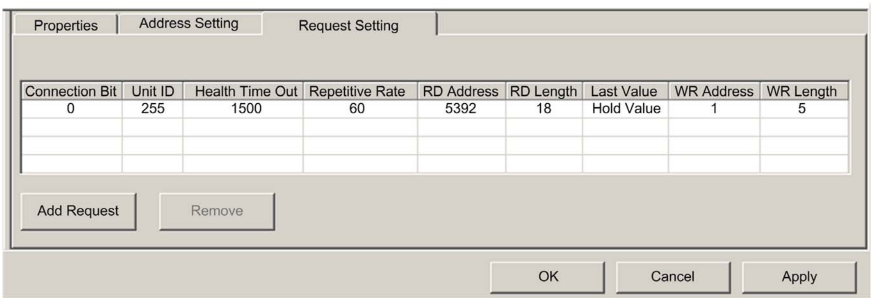
- 1 indicates the connection is active
- 0 indicates the connection is inactive

The health bits are contained in a 32-byte array in the **Variables** page of the **Data Editor**. To display offline this byte array, first sort the variables in ascending order of address, then open the first input variable as shown below:

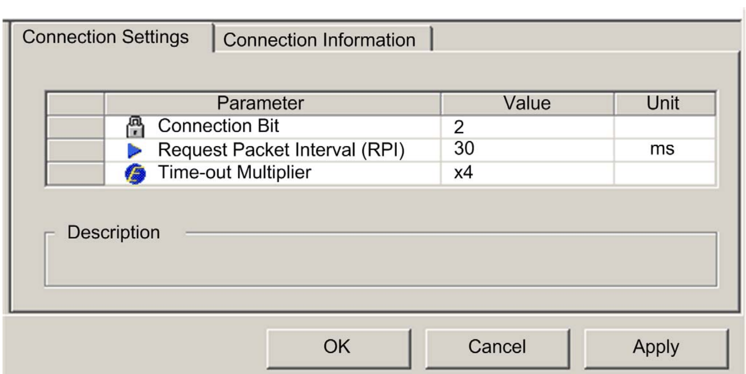


To determine which health bit is mapped to a specific remote device connection, in the **Device Editor** for the Ethernet communication module:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>In the Device Editor for the Ethernet communication module, under the Device List node, navigate to and select:
● for Modbus TCP devices: the main device node
● for EtherNet/IP devices: a connection node</td></tr>
<tr><td>2</td><td>For a Modbus TCP device, open the Request Setting page and look for the Connection Bit number:



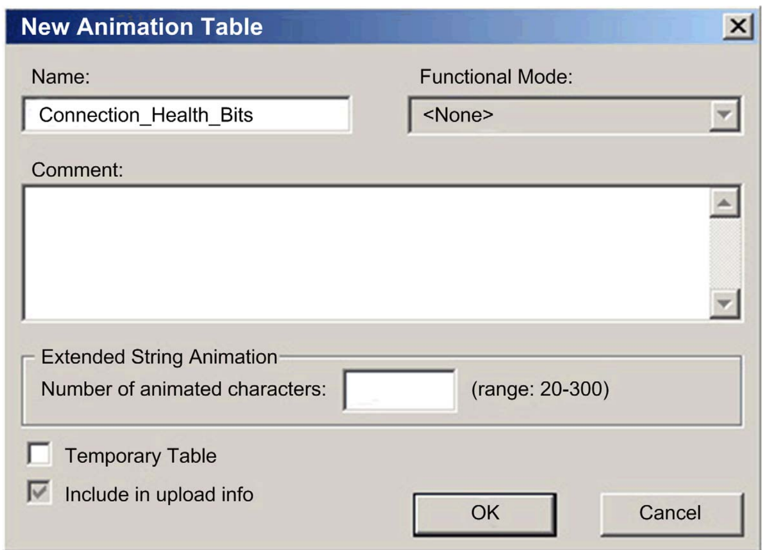
In the above example (which displays the left portion of a truncated Request Setting page), the Connection Bit value of 0 maps to the first bit in the first byte of the HEALTH_BITS_IN array, which can be represented as <code>HEALTH_BITS_IN[0].0</code>.</td></tr>
<tr><td>3</td><td>For an EtherNet/IP device, open the Connection Settings page and look for the Connection Bit number:

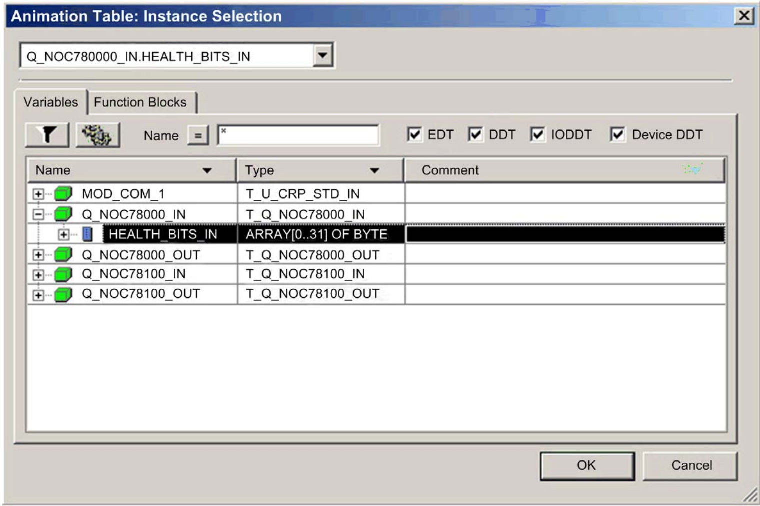
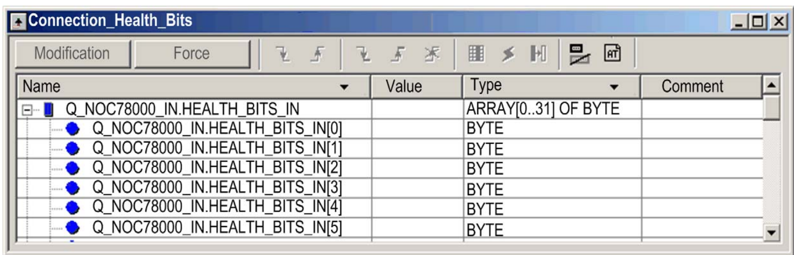


In the above example, the Connection Bit value of 2 maps to the third bit in the first byte of the HEALTH_BITS_IN array, which can be represented as <code>HEALTH_BITS_IN[0].2</code>.</td></tr>
<tr><td>4</td><td>For a local slave, open the local slave configuration page and look for the Connection Bit number:</td></tr>
</table>

Monitoring Connection Health Bits in an Animation Table

Use an animation table to monitor the status of connection health bits and other variables. To add health bits to an animation table, follow these steps:

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>1</td><td colspan="2">In the Project Browser, select the Animation Tables node and click the right mouse button. A pop-up menu opens.</td></tr>
<tr><td>2</td><td colspan="2">Select New Animation Table.</td></tr>
<tr><td rowspan="4">3</td><td colspan="2">In the New Animation Table dialog, type in values for the following fields:</td></tr>
<tr><td>Name</td><td>Type a name for the new animation table. In this example, type Connection_Health_Bits.</td></tr>
<tr><td>Number of animated characters</td><td>Accept the default value of 100.</td></tr>
<tr><td colspan="2">The completed dialog looks like this:

</td></tr>
<tr><td>4</td><td colspan="2">Click OK.
Result: The dialog closes, and the new Connection_Health_Bits animation table opens.</td></tr>
<tr><td>5</td><td colspan="2">Double-click the first empty row in the Name column, then click the ellipsis button .
Result: The Instance Selection dialog opens.</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>6</td><td>In the Instance Selection dialog, navigate to and select the entire <code>HEALTH_BITS_IN</code> array:

</td></tr>
<tr><td>7</td><td>Click OK to add the array to the Connection_Health_Bits animation table:


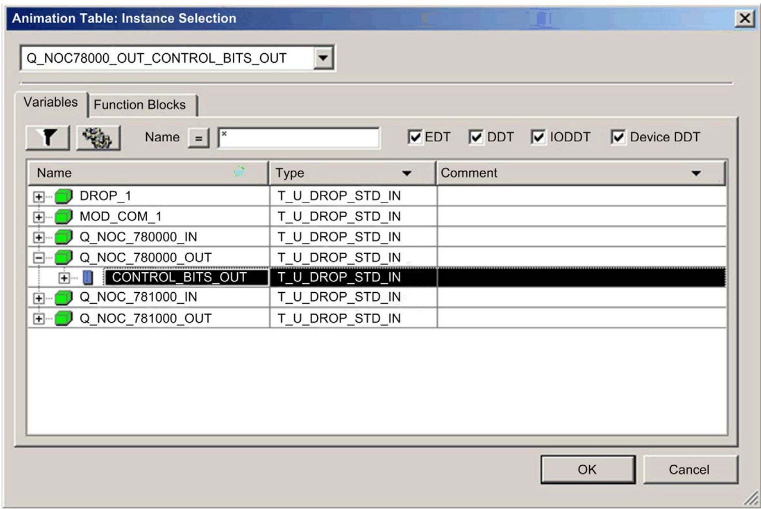
Keep in mind that each row represents a byte, which contains 8 individual connection health bits. When the DTM for the Ethernet communication module is connected to the physical module, the Value field displays a value for the entire byte.</td></tr>
</table>

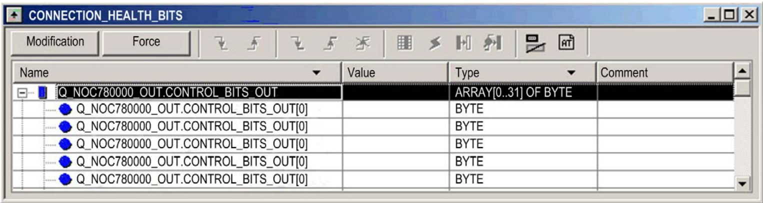
Modifying Connection Control Bits in an Animation Table

You can also use an animation table to modify the value of a control bit, toggling it on or off.

NOTE: Using control bits to toggle a connection on or off (as described below) is the preferred way of regulating communication with a remote device. Toggling a connection control bit on and off does not affect the address location of I/O items. In either case — on or off — the I/O items remain a part of the configuration at the same address locations.
By contrast, enabling and disabling the **Active Configuration** property for a device or local slave either adds I/O items to, or removes I/O items from, the application. This has the rippling effect of changing the addresses not only for the items of the enabled/disabled device, but also for I/O items relating to other devices in the configuration.

The following example shows you how to add connection control bits to the **Connection_Health_Bits** animation table that you created, above, and use the animation table's **Modification** function to toggle control bits on or off:

Step	Action
1	With the **Connection_Health_Bits** animation table open, double-click the next empty row in the **Name** column, then click the ellipsis button . **Result**: The **Instance Selection** dialog opens.
2	In the **Instance Selection** dialog, navigate to and select the entire `CONTROL_BITS_OUT` array: 

Step	Action
3	Click **OK** to add the control bit array to the **Connection_Health_Bits** animation table:  Keep in mind that each row represents a byte, which contains 8 individual connection control bits. When the DTM for the Ethernet communication module is connected to the physical module, the **Value** field displays a value for the entire byte.
4	With the DTM for the Ethernet communication module connected to the physical module, double-click the **Value** column for the row (byte) that contains the control bit you want to toggle.
5	Type a value that toggles the bit (or bits) in the byte you want to change to on or off. For example, suppose the **Value** field of the control bit displays an initial value of 7. This indicates that the first three (0, 1, and 2) are not established. If you intend to establish the third connection (i.e. connection number 2), modify the corresponding bit to 0 (type a value of 3). **NOTE:** When the control bit is 0, the connection is established. When the control bit is 1, the connection is closed.
6	On your keyboard, press **Enter**. **Result**: The control bit for the third connection (i.e. connection number 2) is toggled off.

Effect of Activating and De-activating Devices on I/O %MW Memory Addresses

Introduction

Control Expert assigns a located address in %MW memory to each input and output variable for a remote device and local slave, when that device or slave is activated.

In addition, Control Expert removes from %MW memory each located variable address whenever the related device or slave is de-activated.

In each case, because of the ordered structure of I/O items in PLC memory, the activation and de-activation of a single device causes a rippling effect on the address locations of other I/O variables throughout the application.

Because activating and de-activating devices can cause substantial changes to located variable addresses, Schneider Electric recommends the following practices:

- Activate all the devices and local slaves your application is likely to use, and allow these devices to remain activated.
- If it subsequently becomes necessary to disable communications to a device or slave, do not de-activate it; instead, use the appropriate control bits to toggle off all connections to that slave or device.
- When configuring function blocks in Control Expert, do not directly assign input and output pins to a specific %MW address. Instead, assign specific input and output pins only to the derived data types and variables automatically created by Control Expert.

The Sample Network

The sample network is a part of the same physical network that has been the subject of our continuing configuration example, and includes:

- the Ethernet communication module, named Q_NOC78000
- an STB NIC 2212 EtherNet/IP network interface module with I/O modules, named NIC2212_01

Note that, when a new network is created, Control Expert presents 3 local slave nodes that can be activated and pre-assigns them device numbers 000, 001, and 002. By default, each local slave is not activated. Therefore, each local slave's inputs and outputs are not initially assigned a %MW memory address.

The following example describes the effect of activating a local slave function after another remote device has already been configured and added to the network. In this case:

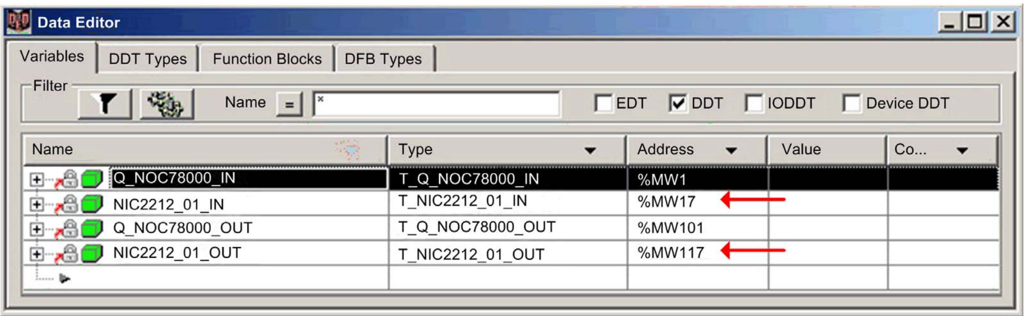
The sample Ethernet network has been configured as follows:

- Total network inputs and outputs are set in the **Configuration** page of the Ethernet communication module in Control Expert:
  - 100 input words are reserved, beginning at %MW01
  - 100 output words are reserved, beginning at %MW101
- Connection bits for the project include:
  - 32 input bytes (16 words) for health bits with an instance name of Q_NOC78000_IN
  - 32 output bytes (16 words) for control bits with an instance name of Q_NOC78000_OUT

- Local slave inputs and outputs include:
  - 8 input bytes (4 words) are reserved with an instance name of Q_NOC78000_LS1_IN
  - 2 output bytes (1 word) is reserved with an instance name of Q_NOC78000_LS1_OUT
- Remote EtherNet/IP device inputs and outputs include:
  - 19 input bytes (10 words) are reserved with an instance name of NIC2212_01_IN
  - 6 output bytes (3 words) are reserved with an instance name of NIC2212_01_OUT

I/O Assignment Without an Activated Local Slave

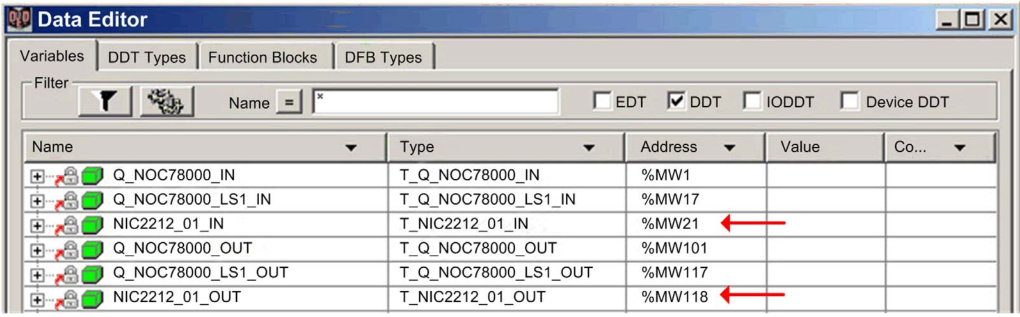
When you click the **Update application** button in the Ethernet communication module **Configuration** page, with the local slave de-activated, Control Expert auto-generates a collection of variables in support of the application's I/O items at the following instance locations:



Notice the address locations of the remote EtherNet/IP device's inputs (%MW17) and outputs (%MW117). As you will see, below, when the local slave is activated, these address locations will change.

I/O Assignment With an Activated Local Slave

The following example displays input and output variables for the same project. However, in this example the **Active Configuration** setting for the first local slave was set to **Enabled** in the local slave configuration page *(see page 155)*, before the input and output variables were created. As a result clicking the **Update application** button in the Ethernet communication module **Configuration** page generated the following collection of variables:



Notice how the address locations for the remote EtherNet/IP device have shifted:

- inputs (NIC2212_01_IN) have shifted from %MW17 to %MW21
- outputs (NIC2212_01_OUT) have shifted from %MW117 to %MW118

This shift of %MW input and output memory address assignments occurs because the local slave was activated, and local slave I/O variables are placed in a located memory address position ahead of remote EtherNet/IP device I/O variables.

A similar shift of addresses would occur — with respect to both local slave and EtherNet/IP device I/O variable addresses — if a Modbus TCP remote device is activated. This is because Modbus TCP device I/O variables are places in a located memory address position ahead of both local slave and EtherNet/IP I/O variables.

As stated above, to help prevent this shift of I/O memory addresses, activate all local slaves and remote devices that your project may require, and then allow them to remain active. If you later disable a device, use the appropriate control bits to toggle off all connections to that device.

Chapter 8
Explicit Messaging

Overview

EtherNet/IP uses the TCP/IP and UDP/IP protocols to implement both explicit and implicit messaging.

This chapter explains the 140 NOC 78• 00 head module and its use of explicit messaging for request-and-reply communications for non-real-time information (like configuration and diagnostic data). A network node that receives a TCP/IP-encapsulated explicit message processes the message and generates a response.

NOTE:

- The 140 NOC 78• 00 head module can process 16 MBP_MSTR blocks per MAST cycle.
- The 140 NOC 78• 00 head module processes MBP_MSTR blocks to reach data on the control network only.

NOTE: If you receive a detected error message — stating that the 140CRP31200 module does not have a link (has no cable), MSTR functionality may not be available. If this happens, check that your cables are connected properly.

This chapter describes how to use both Control Expert function block logic and the Control Expert interface to send explicit messages.

What Is in This Chapter?

This chapter contains the following sections:

Section 8.1
Explicit Messaging Using the MBP_MSTR Block

Configuring Explicit Messaging Using MBP_MSTR

Overview

You can use the MBP_MSTR function block to configure both Modbus TCP and EtherNet/IP connected and unconnected explicit messages.

The operation begins when the input to the EN pin is turned ON. The operation ends if the ABORT pin is turned ON, or if the EN pin is turned OFF.

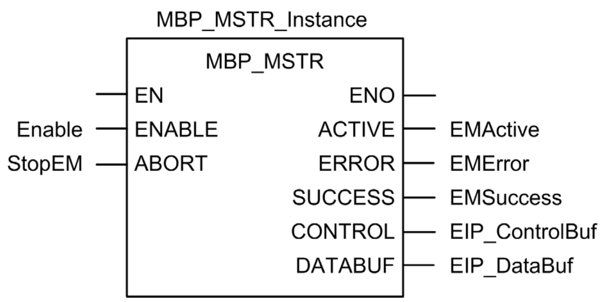
The CONTROL and DATABUF output parameters define the operation.

NOTE: The structure and content of the CONTROL and DATABUF output parameters differ for explicit messages configured using the EtherNet/IP and Modbus TCP protocols. Refer to the topics Configuring the Control Parameter for EtherNet/IP and Configuring the Control Parameter for Modbus TCP for instructions on how to configure these parameters for each protocol.

The ACTIVE output turns ON during operation; the ERROR output turns ON if the operation aborts without success; the SUCCESS output turns ON at the successful completion of the operation.

EN and ENO can be configured as additional parameters.

Representation in FBD



Input Parameters

Parameter	Data type	Description
`ENABLE`	`BOOL`	When ON, the explicit message operation (specified in the first element of the `CONTROL` pin) is executing.
`ABORT`	`BOOL`	When ON, the operation is aborted.

Output Parameters

Parameter	Data type	Description
`ACTIVE`	`BOOL`	ON when the operation is active. OFF at all other times.
`ERROR`	`BOOL`	ON when the operation is aborted without success. OFF before operation, during operation, and if operation succeeds.
`SUCCESS`	`BOOL`	ON when the operation concludes successfully. OFF before operation, during operation, and if operation does not conclude successfully.
`CONTROL`[1]	`WORD`	This parameter contains the control block. The first element contains a code describing the operation to be performed. The content of the control block depends on the operation. The structure of the control block depends on the protocol (EtherNet/IP or Modbus TCP). **Note:** Assign this parameter to a located variable.
`DATABUF`[1]	`WORD`	This parameter contains the data buffer. For operations that: • provide data — e.g., a write operation — this parameter is the data source • receive data — e.g., a read operation — this parameter is the data destination **Note:** Assign this parameter to a located variable.
1. Refer to the topics Configuring the Control Block for EtherNet/IP and Configuring the Control Block for Modbus TCP for instructions on how to configure these parameters for the EtherNet/IP and Modbus TCP communication protocols.		

Section 8.2
EtherNet/IP Explicit Messaging Using MBP_MSTR

Overview

This section shows you how to configure the `MBP_MSTR` function block for EtherNet/IP explicit messages.

What Is in This Section?

This section contains the following topics:

EtherNet/IP Explicit Messaging Services

Overview

Every EtherNet/IP explicit message performs a service. Each service is associated with a service code (or number). You will need to identify the explicit messaging service by its name, decimal number, or hexadecimal number.

You can execute EtherNet/IP explicit messages using either a Control Expert `MBP_MSTR` function block or the Control Expert Ethernet Configuration Tool's **EtherNet/IP Explicit Message Window**.

NOTE: Configuration edits made to an Ethernet communication module from the Control Expert Ethernet Configuration Tool's EtherNet/IP Explicit Message Window are not saved to the operating parameters stored in the CPU and, therefore, are not sent by the CPU to the module on startup.

You can use Control Expert to construct a request that executes any service supported by the target device that is compliant with the EtherNet/IP protocol.

Services

The services supported by Control Expert include the following standard explicit messaging services:

Service Code		Description	Available in...	
Hex	Dec		MBP_MSTR block	Control Expert GUI
1	1	Get_Attributes_All	X	X
2	2	Set_Attributes_All	X	X
3	3	Get_Attribute_List	X	—
4	4	Set_Attribute_List	X	—
5	5	Reset	X	X
6	6	Start	X	X
7	7	Stop	X	X
8	8	Create	X	X
9	9	Delete	X	X
A	10	Multiple_Service_Packet	X	—
D	13	Apply_Attributes	X	X
E	14	Get_Attribute_Single	X	X
10	16	Set_Attribute_Single	X	X
11	17	Find_Next_Object_Instance	X	X
"X" = the service is available. "—" = the service is not available.				

Service Code		Description	Available in...	
Hex	Dec		MBP_MSTR block	Control Expert GUI
14	20	Detected Error Response (DeviceNet only)	—	—
15	21	Restore	X	X
16	22	Save	X	X
17	23	No Operation (NOP)	X	X
18	24	Get_Member	X	X
19	25	Set_Member	X	X
1A	26	Insert_Member	X	X
1B	27	Remove_Member	X	X
1C	28	GroupSync	X	—
"X" = the service is available. "—" = the service is not available.				

Configuring the CONTROL and DATABUF Parameters

Overview

The CONTROL and DATABUF output parameters define the operation performed by the MBP_MSTR function block. For the EtherNet/IP protocol, the structure of the CONTROL and DATABUF output parameters remains the same for every explicit messaging service *(see page 221)*.

Configuring the Control Parameter

The Control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
CONTROL[0]	Operation	• 14 = unconnected • 270 = connected
CONTROL[1]	Detected error status	Holds the event code *(see page 373)* (read-only).
CONTROL[2]	Data buffer length	Data buffer length, in words
CONTROL[3]	Response offset	Offset for the beginning of the response in the data buffer, in 16-bit words **Note:** To avoid overwriting the request, confirm that the response offset value is greater than the request length CONTROL[7].
CONTROL[4]	Slot	High byte = slot location on backplane
		Low byte = 0 (not used)
CONTROL[5] [1]	IP address	High byte = byte 4 of the IP address (MSB)
		Low byte = byte 3 of the IP address
CONTROL[6] [1]		High byte = byte 2 of the IP address
		Low byte = byte 1 of the IP address (LSB)
CONTROL[7]	Request length	Length of the CIP request, in bytes
CONTROL[8]	Response length	Length of the response received, in bytes Read only—set after completion
1. For example, the Control parameter handles the IP address 192.168.1.6 in the following order: Byte 4 = 192, Byte 3 = 168, Byte 2 = 1, Byte 1 = 6.		

Configuring the Data Buffer

The data buffer varies in size. It consists of contiguous registers that include—in sequence—both the CIP request and the CIP response. To avoid overwriting the request, confirm that the data buffer is large enough to simultaneously contain both the request and response data.

<table>
<tr><td rowspan="2">Data Buffer:
Variable size: set in <code>CONTROL[2]</code></td><td>CIP Request:
Request size: set in <code>CONTROL[7]</code></td></tr>
<tr><td>CIP Response:
Starting position: set in <code>CONTROL[3]</code>
Response size: reported in <code>CONTROL[8]</code>
NOTE: If the response offset is smaller than the request size, the response data overwrites part of the request.</td></tr>
</table>

The format of the data buffer's CIP request and CIP response is described, below.

NOTE: Structure both the request and response in little endian order.

Request:

Byte offset	Field	Data type	Description
0	Service	Byte	Service of the explicit message
1	Request_Path_Size	Byte	The number of words in the Request_Path field
2	Request_Path	Padded EPATH	This byte array describes the path of the request—including class ID, instance ID, etc.—for this transaction
...	Request_Data	Byte array	Service specific data to be delivered in the explicit message request—if none, this field is empty

Response:

Byte offset	Field	Data type	Description
0	Reply Service	Byte	Service of the explicit message + 16#80
1	Reserved	Byte	0
2	General Status	Byte	EtherNet/IP General Status *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)*
3	Size of Additional Status	Byte	Additional Status array size—in words
4	Additional Status	Word array	Additional status[1]
...	Response Data	Byte array	Response data from request, or additional detected error data if General Status indicates a detected error
1. Refer to *The CIP Networks Library, Volume 1, Common Industrial Protocol* at section 3-5.6 *Connection Manager Object Instance Detected Error Codes*;			

MBP_MSTR Example: Get_Attributes_Single

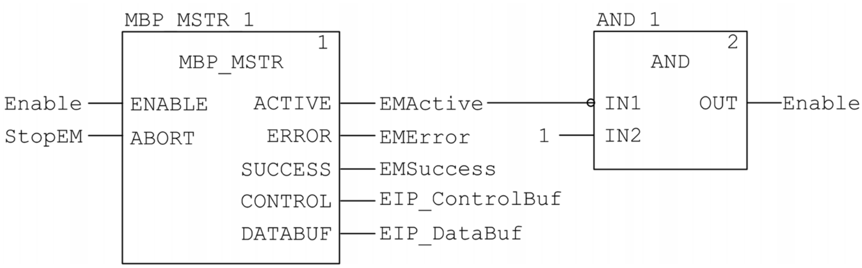
Overview

The following unconnected explicit messaging example shows you how to use the `MBP_MSTR` function block to retrieve diagnostic information for an STB island from an STB NIC 2212 network interface module, by using the Get_Attributes_Single service.

You can perform the same explicit messaging service using the **EtherNet/IP Explicit Message Window** of the Control Expert Ethernet Configuration Tool *(see page 241)*.

Implementing the MBP_MSTR Function Block

To implement the `MBP_MSTR` function block, you need to create and assign variables, then connect it to an `AND` block. In the following example, the logic will continuously send an explicit message upon receiving notice of success:



Input Variables

Variables need to be created and assigned to input pins. For the purpose of this example, variables have been created — and named — as described below. (You can use different variable names in your explicit messaging configurations.)

Input Pin	Variable	Data Type
`ENABLE`	Enable	`BOOL`
`ABORT`	StopEM	`BOOL`

Output Variables

Variables also need to be created and assigned to output pins. (The names assigned to output variables apply only to this example, and can be changed in your explicit messaging configurations.)

Output Pin	Variable	Data Type
ACTIVE	EMActive	BOOL
ERROR	EMError	BOOL
SUCCESS	EMSuccess	BOOL
CONTROL	EIP_ControlBuf	Array of 10 WORDs
DATABUF	EIP_DataBuf	Array of 100 WORDs

NOTE: To simplify configuration, you can assign the CONTROL and DATABUF output pins to a byte array consisting of located variables. When configured in this manner, you will not need to be aware of the location of data within a word (for example, high versus low byte, and big or little endian format).

Control Array

The control array parameter (EIP_ControlBuf) consists of 9 contiguous words. You need to configure only some control words; other control words are read-only and are written to by the operation. In this example, the control array defines the operation as an unconnected explicit message, and identifies the target device:

Register	Description	Configure	Setting (hex)
CONTROL[0]	Operation: High byte = ● 00 (unconnected), or ● 01 (connected) Low byte = 0E (CIP explicit message)	Yes	16#000E (unconnected)
CONTROL[1]	Detected error status: read-only (written by operation)	No	16#0000
CONTROL[2]	Data buffer length = 100 words	Yes	16#0064
CONTROL[3]	Response offset: offset — in words — for the beginning of the explicit message response in the databuffer	Yes	16#0004
CONTROL[4]	High byte = slot location of the communication module in the backplane Low byte = 0 (not used)	Yes	16#0400
CONTROL[5] [1]	IP address of the Ethernet communication module: High byte = byte 4 of the IP address Low byte = byte 3 of the IP address	Yes	16#C0A8

Register	Description	Configure	Setting (hex)
`CONTROL[6]` [1]	IP address of the Ethernet communication module: High byte = byte 2 of the IP address Low byte = byte 1 of the IP address	Yes	16#0106
`CONTROL[7]`	CIP request length (in bytes)	Yes	16#0008
`CONTROL[8]`	Length of received response (written by operation)	No	16#0000
1. In this example, the control parameter handles the IP address 192.168.1.6 in the following order: Byte 4 = 192, Byte 3 = 168, Byte 2 = 1, Byte 1 - 6.			

CIP Request

The CIP request is located at the beginning of the databuffer and is followed by the CIP response. In this example, the CIP request calls for the return of a single attribute value (diagnostic data), and describes the request path through the target device's object structure leading to the target attribute:

Request word	High byte		Low byte	
	Description	Value (hex)	Description	Value (hex)
1	Request path size (in words)	16#03	EM Service: Get_Attributes_Single	16#0E
2	Request path: class assembly object	16#04	Request path: logical class segment	16#20
3	Request path: instance	16#64	Request path: logical instance segment	16#24
4	Request path: attribute	16#03	Request path: logical attribute segment	16#30

Combining the high and low bytes, above, the CIP request would look like this:

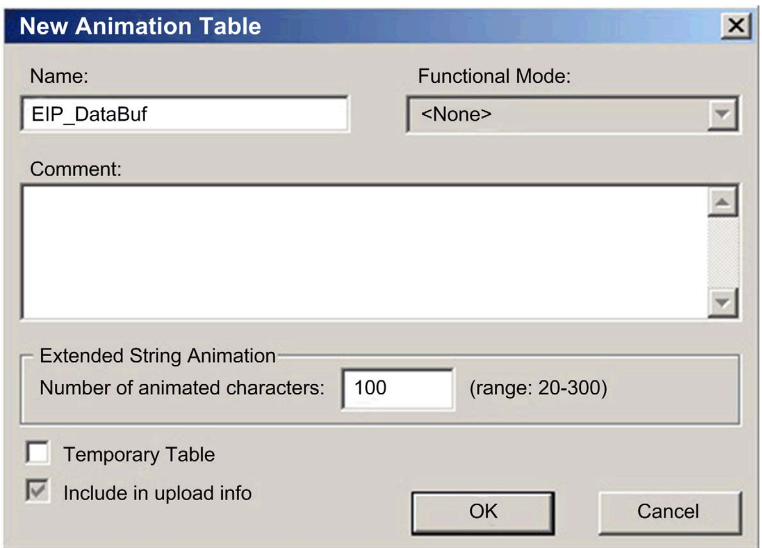
Request word	Value
1	16#030E
2	16#0420
3	16#6424
4	16#0330

Viewing the Response

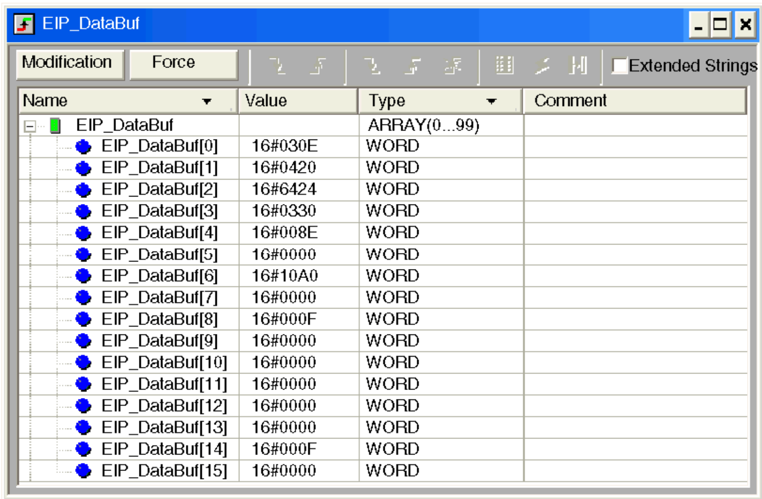
Use a Control Expert Animation table to display the EIP_DataBuf variable array. Note that the EIP_DataBuf variable array consists of the entire data buffer, which includes the:

- CIP request (4 words) located in EIP_DataBuf(1-4)
- CIP service type (1 word) located in EIP_DataBuf(5)
- CIP request status (1 word) located in EIP_DataBuf(6)
- CIP response (in this case, 10 words) located in EIP_DataBuf(7-16)

To display the CIP response, follow these steps:

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>1</td><td colspan="2">In Control Expert, select Tools → Project Browser to open the Project Browser.</td></tr>
<tr><td>2</td><td colspan="2">In the Project Browser, right-click Animation Tables → New Animation Table.
Result: A new animation table opens.</td></tr>
<tr><td rowspan="5">3</td><td colspan="2">In the New Animation Table dialog, edit the following values:</td></tr>
<tr><td>Name</td><td>Type in a table name. For this example: EIP_DataBuf.</td></tr>
<tr><td>Functional Mode</td><td>Accept the default <None>.</td></tr>
<tr><td>Comment</td><td>Leave blank.</td></tr>
<tr><td>Number of animated characters</td><td>Type 100, representing the size of the data buffer in words.</td></tr>
<tr><td>4</td><td colspan="2">The completed dialog looks like this:


Click OK to close the dialog.</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>5</td><td>In the animation table’s Name column, type in the name of the variable assigned to the databuffer: EIP_DataBuf and press Enter. The animation table displays the EIP_DataBuf variable.</td></tr>
<tr><td>6</td><td>Expand the EIP_DataBuf variable to display its word array, where you can view the CIP response at words EIP_DataBuf(7-16):


<table>
<tr><th>Name</th><th>Value</th><th>Type</th><th>Comment</th></tr>
<tr><td>EIP_DataBuf</td><td></td><td>ARRAY(0...99)</td><td></td></tr>
<tr><td>EIP_DataBuf[0]</td><td>16#030E</td><td>WORD</td><td></td></tr>
<tr><td>EIP_DataBuf[1]</td><td>16#0420</td><td>WORD</td><td></td></tr>
<tr><td>EIP_DataBuf[2]</td><td>16#6424</td><td>WORD</td><td></td></tr>
<tr><td>EIP_DataBuf[3]</td><td>16#0330</td><td>WORD</td><td></td></tr>
<tr><td>EIP_DataBuf[4]</td><td>16#008E</td><td>WORD</td><td></td></tr>
<tr><td>EIP_DataBuf[5]</td><td>16#0000</td><td>WORD</td><td></td></tr>
<tr><td>EIP_DataBuf[6]</td><td>16#10A0</td><td>WORD</td><td></td></tr>
<tr><td>EIP_DataBuf[7]</td><td>16#0000</td><td>WORD</td><td></td></tr>
<tr><td>EIP_DataBuf[8]</td><td>16#000F</td><td>WORD</td><td></td></tr>
<tr><td>EIP_DataBuf[9]</td><td>16#0000</td><td>WORD</td><td></td></tr>
<tr><td>EIP_DataBuf[10]</td><td>16#0000</td><td>WORD</td><td></td></tr>
<tr><td>EIP_DataBuf[11]</td><td>16#0000</td><td>WORD</td><td></td></tr>
<tr><td>EIP_DataBuf[12]</td><td>16#0000</td><td>WORD</td><td></td></tr>
<tr><td>EIP_DataBuf[13]</td><td>16#0000</td><td>WORD</td><td></td></tr>
<tr><td>EIP_DataBuf[14]</td><td>16#000F</td><td>WORD</td><td></td></tr>
<tr><td>EIP_DataBuf[15]</td><td>16#0000</td><td>WORD</td><td></td></tr>
</table>

Note: Each word presents 2 bytes of data in little endian format, where the least significant byte is stored in the smallest memory address. For example, ‘0E’ in EIP_DataBuf[0] is the low byte, and ‘03’ is the high byte.</td></tr>
</table>

Section 8.3
Modbus TCP Explicit Messaging Using MBP_MSTR

Overview

This section shows you how to configure the MBP_MSTR function block to send explicit messages using the Modbus TCP protocol.

What Is in This Section?

This section contains the following topics:

Modbus TCP Explicit Messaging Function Codes

Overview

Every Modbus TCP explicit message performs a function. Each function is associated with a code (or number). You will need to identify the explicit messaging function by its name, decimal number, or hexadecimal number.

You can execute Modbus TCP explicit messages using either a Control Expert `MBP_MSTR` function block or the Control Expert Ethernet Configuration Tool's **Modbus Explicit Message Window**.

NOTE: Configuration edits made to an Ethernet communication module from the Control Expert Ethernet Configuration Tool are not saved to the operating parameters stored in the CPU and, therefore, are not sent by the CPU to the module on startup.

Services

The function codes supported by Control Expert include the following standard explicit messaging functions:

Function Code		Description	Available in...	
Hex	Dec		MBP_MSTR block	Control Expert GUI
1	1	Write data	X	X
2	2	Read data	X	X
3	3	Get local statistics	X	X
4	4	Clear local statistics	X	X
7	7	Get remote statistics	X	X
8	8	Clear remote statistics	X	X
A	10	Reset module	X	X
17	23	Read / write data	X	X
FFF0	65520	Enable / disable HTTP and FTP/TFTP services	X	-
"X" = the service is available. "—" = the service is not available.				

Configuring the Control Parameter for Modbus TCP Explicit Messaging

Overview

The `CONTROL` and `DATABUF` output parameters define the operation performed by the `MBP_MSTR` *(see page 218)* function block. For the Modbus TCP protocol, both the structure and the content of the `CONTROL` output parameter vary, depending upon the function code *(see page 231)*.

The structure of the `CONTROL` parameter is described, below, for each supported function code.

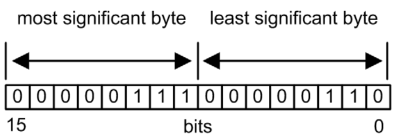
Refer to the *Quantum Ethernet I/O System Planning Guide* for an example of an MSTR block created in a Control Expert application to read the ports of a dual-ring switch (DRS) to diagnose a sub-ring break.

Control Parameter Routing Register

The `CONTROL[5]` routing register specifies the source and destination node addresses for network data transfer, and consists of the following 2 bytes:

- Most Significant Byte (MSB): contains the source node address, for example, the slot number of the 140 NOC 78• 00
- Least Significant Byte (LSB): contains the destination node address — a value representing either a direct or a bridge address. The LSB is required for devices that are reached through a bridge, for example, an Ethernet to Modbus bridge or an Ethernet to Modbus Plus bridge. The values of the LSB are as follows:
  - If no bridge is used: LSB is set to zero(0).
  - If a bridge is used: LSB contains the Modbus Plus on Ethernet Transporter (MET) mapping index value. This value, also known as the Unit ID, indicates the device to which the message is directed.

The `CONTROL[5]` routing register:



When the Ethernet communication module acts as a server, the LSB indicates the destination of a message received by the communication module:

- messages with an LSB value from 0 to 254 are forwarded to and processed by the CPU
- messages with an LSB value of 255 are retained and processed by the Ethernet communication module

NOTE: Unit ID 255 should be used when requesting diagnostic data from the Ethernet communication module.

Write Data

The control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
CONTROL[1]	Operation	1 = write data
CONTROL[2]	Detected error status	Holds the event code *(see page 373)* (read-only)
CONTROL[3]	Data buffer length	Number of addresses sent to the slave
CONTROL[4]	Starting register	Start address of the slave to which the data is written, in 16-bit words
CONTROL[5]	Routing register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index
CONTROL[6] [1]	IP address	Byte 4 of the IP address (MSB)
CONTROL[7] [1]		Byte 3 of the IP address
CONTROL[8] [1]		Byte 2 of the IP address
CONTROL[9] [1]		Byte 1 of the IP address (LSB)
1. For example, the control parameter handles the IP address 192.168.1.7 in the following order: Byte 4 = 192, Byte 3 = 168, Byte 2 = 1, Byte 1 = 7.		

Read Data

The control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
CONTROL[1]	Operation	2 = read data
CONTROL[2]	Detected error status	Holds the event code *(see page 373)* (read-only)
CONTROL[3]	Data buffer length	Number of addresses to be read from the slave
CONTROL[4]	Starting register	Determines the %MW starting register in the slave from which the data is read. For example: 1 = %MW1, 49 = %MW49)
CONTROL[5]	Routing register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index
CONTROL[6] [1]	IP address	Byte 4 of the IP address (MSB)
CONTROL[7] [1]		Byte 3 of the IP address
CONTROL[8] [1]		Byte 2 of the IP address
CONTROL[9] [1]		Byte 1 of the IP address (LSB)
1. For example, the control parameter handles the IP address 192.168.1.7 in the following order: Byte 4 = 192, Byte 3 = 168, Byte 2 = 1, Byte 1 = 7.		

Get Local Statistics

The control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
`CONTROL[1]`	Operation	3 = read local statistics
`CONTROL[2]`	Detected error status	Holds the event code *(see page 373)* (read-only)
`CONTROL[3]`	Data buffer length	Number of addresses to be read from local statistics (0...37)
`CONTROL[4]`	Starting register	First address from which the statistics table is read (Reg1=0)
`CONTROL[5]`	Routing register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index
`CONTROL[6]`	(not used)	—
`CONTROL[7]`		
`CONTROL[8]`		
`CONTROL[9]`		

Module Response: A TCP/IP Ethernet module responds to the `Get Local Statistics` command with the following information:

<table>
<tr><th>Word</th><th colspan="4">Description</th></tr>
<tr><td>00...02</td><td colspan="4">MAC Address</td></tr>
<tr><td rowspan="7">03</td><td colspan="4">Board Status — this word contains the following bits:</td></tr>
<tr><td>Bit 15</td><td>0 = Link LED off;
1 = Link LED ON</td><td>Bit 3</td><td>Reserved</td></tr>
<tr><td>Bits 14...13</td><td>Reserved</td><td>Bit 2</td><td>0 = half duplex; 1 = full duplex</td></tr>
<tr><td>Bit 12</td><td>0 = 10 Mbit;
1 = 100 Mbit</td><td>Bit 1</td><td>0 = not configured; 1 = configured</td></tr>
<tr><td>Bits 11...9</td><td>Reserved</td><td>Bit 0</td><td>0 = PLC not running;
1 = PLC or NOC running</td></tr>
<tr><td rowspan="2">Bits 8...4</td><td colspan="3">Module Type — this bit presents the following values:</td></tr>
<tr><td colspan="2">• 0 = NOE 2x1
• 1 = ENT
• 2 = M1E
• 3 = NOE 771 00
• 4 = ETY
• 5 = CIP
• 6 = (reserved)
• 7 = 140 CPU 651 x0
• 8 = 140 CRP 312 00
• 9 = (reserved)
• 10 = 140 NOE 771 10</td><td>• 11 = 140 NOE 771 01
• 12 = 140 NOE 771 11
• 13 = (reserved)
• 14 = 140 NOC 78• 00
• 15...16 = (reserved)
• 17 = M340 CPU
• 18 = M340 NOE
• 19 = BMX NOC 0401
• 20 = TSX ETC 101
• 21 = 140 NOC 771 01</td></tr>
</table>

Word	Description
04 and 05	Number of receiver interrupts
06 and 07	Number of transmitter interrupts
08 and 09	Transmit_timeout detected error count
10 and 11	Collision_detect error count
12 and 13	Missed packets
14 and 15	(reserved)
16 and 17	Number of times driver has restarted
18 and 19	Receive framing detected error
20 and 21	Receiver overflow detected error
22 and 23	Receive CRC detected error
24 and 25	Receive buffer detected error
26 and 27	Transmit buffer detected error
28 and 29	Transmit silo underflow
30 and 31	Late collision
32 and 33	Lost carrier
34 and 35	Number of retries
36 and 37	IP address

Clear Local Statistics

The control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
`CONTROL[1]`	Operation	4 = clear local statistics
`CONTROL[2]`	Detected error status	Holds the event code *(see page 373)* (read-only)
`CONTROL[3]`	(not used)	—
`CONTROL[4]`	(not used)	—
`CONTROL[5]`	Routing register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index
`CONTROL[6]`	(not used)	—
`CONTROL[7]`		
`CONTROL[8]`		
`CONTROL[9]`		

Get Remote Statistics

The control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
CONTROL[1]	Operation	7 = get remote statistics
CONTROL[2]	Detected error status	Holds the event code *(see page 373)* (read-only)
CONTROL[3]	Data buffer length	Number of addresses to be read from the statistics data field (0...37)
CONTROL[4]	Starting register	First address from which the node statistics table is read
CONTROL[5]	Routing register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index
CONTROL[6] [1]	IP address	Byte 4 of the IP address (MSB)
CONTROL[7] [1]		Byte 3 of the IP address
CONTROL[8] [1]		Byte 2 of the IP address
CONTROL[9] [1]		Byte 1 of the IP address (LSB)
1. For example, the control parameter handles the IP address 192.168.1.7 in the following order: Byte 4 = 192, Byte 3 = 168, Byte 2 = 1, Byte 1 = 7.		

Clear Remote Statistics

The control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
CONTROL[1]	Operation	8 = clear remote statistics
CONTROL[2]	Detected error status	Holds the event code *(see page 373)* (read-only)
CONTROL[3]	(not used)	—
CONTROL[4]	(not used)	—
CONTROL[5]	Routing register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index
CONTROL[6] [1]	IP address	Byte 4 of the IP address (MSB)
CONTROL[7] [1]		Byte 3 of the IP address
CONTROL[8] [1]		Byte 2 of the IP address
CONTROL[9] [1]		Byte 1 of the IP address (LSB)
1. For example, the control parameter handles the IP address 192.168.1.7 in the following order: Byte 4 = 192, Byte 3 = 168, Byte 2 = 1, Byte 1 = 7.		

Reset Module

The control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
CONTROL[1]	Operation	10 = reset module
CONTROL[2]	Detected error status	Holds the event code *(see page 373)* (read-only)
CONTROL[3]	(not used)	—
CONTROL[4]	(not used)	—
CONTROL[5]	Routing register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index
CONTROL[6]	(not used)	—
CONTROL[7]		
CONTROL[8]		
CONTROL[9]		

Read/Write Data

The control parameter consists of 11 contiguous words, as described below:

Register	Function	Description
CONTROL[1]	Operation	23 = read / write data
CONTROL[2]	Detected error status	Holds the event code *(see page 373)* (read-only)
CONTROL[3]	Data buffer length	Number of addresses sent to the slave
CONTROL[4]	Starting register	Determines the %MW starting register in the slave to which the data will be written. For example: 1 = %MW1, 49 = %MW49)
CONTROL[5]	Routing register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index
CONTROL[6] [1]	IP address	Byte 4 of the IP address (MSB)
CONTROL[7] [1]		Byte 3 of the IP address
CONTROL[8] [1]		Byte 2 of the IP address
CONTROL[9] [1]		Byte 1 of the IP address (LSB)
CONTROL[10]	Data buffer length	Number of addresses to be read from the slave
CONTROL[11]	Starting register	Determines the %MW starting register in the slave from which the data is read. For example: 1 = %MW1, 49 = %MW49)
1. For example, the control parameter handles the IP address 192.168.1.7 in the following order: Byte 4 = 192, Byte 3 = 168, Byte 2 = 1, Byte 1 = 7.		

Enable/Disable HTTP or FTP/TFTP Services

When HTTP or FTP/TFTP has been enabled using Control Expert configuration tools *(see page 153)*, an MSTR block can be used to change the enabled state of the service while the application is running. The MSTR block cannot change the state of the HTTP or FTP/TFTP services if the service was disabled using one of the configuration tools.

The control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
`CONTROL[1]`	Operation	FFF0 (hex) 65520 (dec) = enable / disable HTTP or FTP/TFTP
`CONTROL[2]`	Detected error status	Holds the event code (read-only). Codes returned include: 0x000 (Success): MSTR block with operational code 0xFFF0 was called and the enabled state of HTTP or FTP/TFTP was changed. 0x5068 (Busy): MSTR block with operational code 0xFFF0 was called within 2 seconds of the previous call (regardless of return code from previous call). 0x4001 (Same state): MSTR block with operational code 0xFFF0 was called to change the enabled state of HTTP and FTP/TFTP to the states they were already in. 0x2004 (Invalid data): MSTR block with operational code 0xFFF0 was called and the data in the control block did not match the specifications. 0x5069 (Disabled): If the HTTP or FTP/TFTP service was already disabled via the Control Expert interface when the MSTR block with operational code 0xFFF0 was called to change the state of the disabled service.
`CONTROL[3]`		Set this register to 1.
`CONTROL[4]`		
`CONTROL[5]`	Module slot number and destination ID	High byte = Module slot number communication module slot
		Low byte = Destination ID
`CONTROL[6]`	Request mode	Bit 0 (LSB) = 1: Enable FTP/TFTP Bit 0 (LSB) = 0: Disable FTP/TFTP Bit 1 = 1: Enable HTTP Bit 1 = 0: Disable HTTP
`CONTROL[7]`		Set this register to 0.
`CONTROL[8]`		
`CONTROL[9]`		

HTTP, FTP, and TFTP service state changes made by MSTR with operation code FFF0 (hex) are overridden by the configured value when the module is power-cycled or reset and when a new application is downloaded to the module.

Here are some examples:

State Configured By Control Expert	Action attempted using MSTR with operation code FFF0 (hex)	Result
Disabled	Any	MSTR returns detected error code 0x5069 (service was already disabled by configuration)
Enabled	Disable	MSTR returns code 0x000 (success). ● Another MSTR block action enables the service --OR-- ● The module is reset or power-cycled --OR-- ● A new application is downloaded with the service disabled by configuration
	Enable	MSTR returns detected error code 0x4001 (same state). No change made.

Section 8.4
Explicit Messaging via the Control Expert GUI

What Is in This Section?

This section contains the following topics:

Sending Explicit Messages to EtherNet/IP Devices

Overview

Use the **EtherNet/IP Explicit Message** window to send an explicit message from Control Expert to an EtherNet/IP module or device on the network.

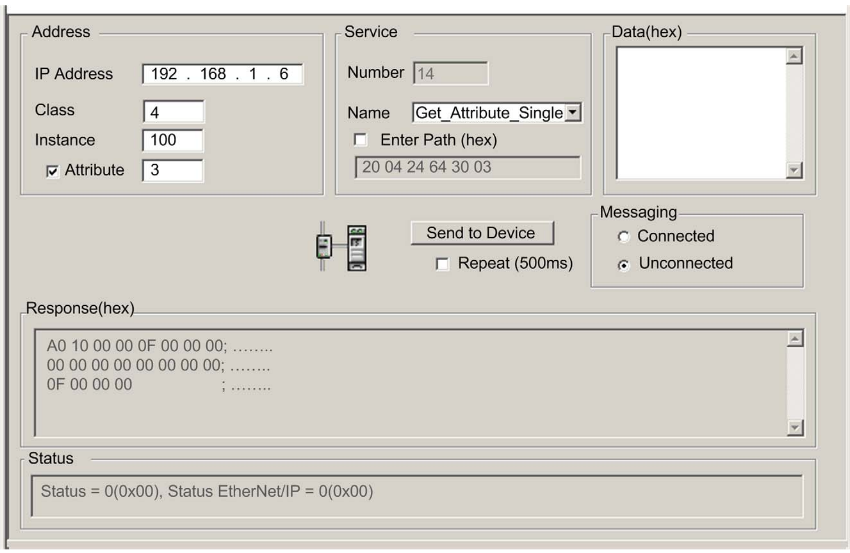
An explicit message can be sent as either a connected, or an unconnected message:

- an unconnected message requires path — or addressing — information identifying the destination device and, optionally, device attributes
- a connected explicit message contains both path information and a connection identifier to the target device

You can use explicit messaging to perform many different services. Not every EtherNet/IP device supports every service.

NOTE: Before you can perform explicit messaging, connect the DTM for the upstream communication module to the module itself. To do this, select the module node in the **DTM Browser**, then select **Edit** → **Connect**.

The **EtherNet/IP Explicit Message** window, below, presents an example of both the configuration of an EtherNet/IP explicit message and the response. The explicit message is addressed to a remote STB NIC 2212 network interface module to obtain diagnostic information.



Sending Explicit Messages

The following steps explain how to execute the EtherNet/IP explicit message, depicted above:

Step	Action	
1	In the **DTM Browser**, select the communication module that is upstream of the target device.	
2	Click the right mouse button, and in the pop-up menu select **Device menu → EtherNet/IP Explicit Message**. **Result**: The **EtherNet/IP Explicit Message** window opens.	
3	Configure the explicit message using the following fields:	
	IP Address	The IP address of the target device, used to identify the target of the explicit message. In the above example: **192.168.1.6**.
	Class	The class identifier of the target device, used in the construction of the message path. An integer from 1 to 65535. In this example: **4**.
	Instance	The class instance of the target device, used in the construction of the message path. An integer from 0 to 65535. In this example: **100**.
	Attribute	(Optional) The specific device attribute — or property — that is the target of the explicit message, used in the construction of the message path. An integer from 0 to 65535. In this example: **3** **NOTE:** Select the check box to enable this field.
	NOTE: Refer to your EtherNet/IP device user manual for class, instance and attribute values.	
	Number	The integer associated with the service to be performed by the explicit message. An integer from 1 to 127. **NOTE:** If you select **Custom Service** as the named service, type in a service number. This field is read-only for all other services.
	Name	Select the service the explicit message is intended to perform. In this example: **Get_Attribute_Single**.
	Enter Path	(Optional) Select this check box to enable the message path field, where you can manually enter the entire path to the target device. In this example, the path is not manually entered. **NOTE:** Displayed only when **Advanced Mode** is enabled.
	Data	The data to be sent to the target device, for services that send data. In this example, leave blank.
	Messaging	Select the type of explicit message to send: • **Connected** • **Unconnected** In this example, select **Unconnected**.
	Repeat 500 ms	Select this check box to re-send the explicit message every 500 ms. In this example, leave this blank.

Step	Action
4	After your explicit message is configured, click **Send to Device**. The **Response** area displays the data sent to the configuration tool by the target device in hexadecimal format. The **Status** area displays messages indicating whether or not the explicit message has succeeded.
5	Click **Close** to close the window.

Sending Explicit Messages to Modbus TCP Devices

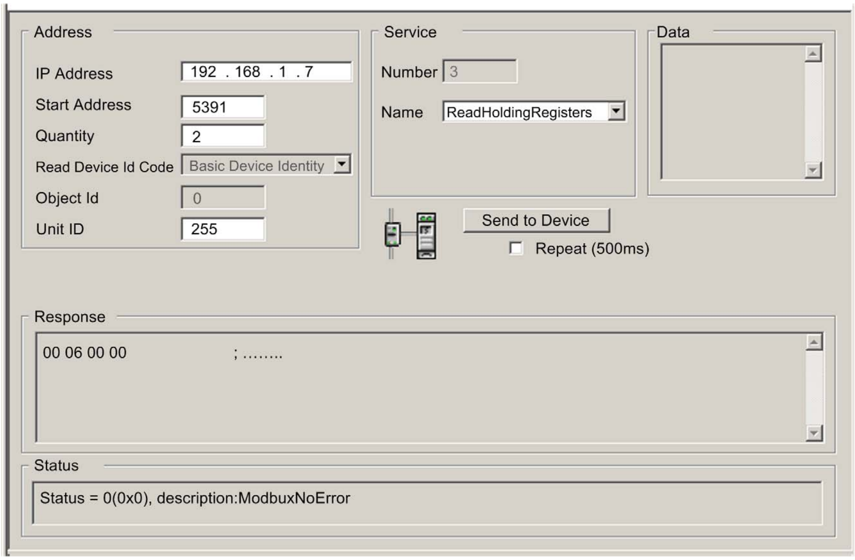
Overview

Use the **Modbus Explicit Message** window to send an explicit message from Control Expert to a Modbus TCP module or device on the network.

You can use explicit messaging to perform many different services. Not all Modbus TCP devices support all services.

NOTE: Before you can perform explicit messaging, connect the DTM for the upstream communication module to the module itself. To do this, select the module node in the **DTM Browser**, then select **Edit** → **Connect**.

The **Modbus TCP Explicit Message** window, below, presents an example of both the configuration of a Modbus TCP explicit message, and the response. In this example, the explicit message is used to read 2 registers in the remote STB NIP 2212 network interface module, starting at offset 5391.



Sending Explicit Messages

To send an explicit message to a target Modbus TCP device:

Step	Action	
1	In the **DTM Browser**, select the communication module that is upstream of the target device.	
2	Click the right mouse button, and in the pop-up menu select **Device menu → Modbus Explicit Message**. **Result**: The **Modbus Explicit Message** window opens.	
3	Configure the explicit message using the following fields:	
	IP Address	The IP address of the target device, used to identify the target of the explicit message. In this example: **192.168.1.7**.
	Start Address	A component of the addressing path. In this example **5391**.
	Quantity	A component of the addressing path. In this example **2**.
	Read Device Id Code	(read-only) The service the explicit message is intended to perform. In this example **Basic Device Identity**. Not used in this example.
	Object Id	(read-only) Specify the object the explicit message is intended to access. In this example **0**. Not used in this example.
	Refer to your Modbus TCP device user manual for Start Address, Quantity, Read Device Id Code, and Object Id values.	
	Unit Id	The number of the device, or module, that is the target of the connection. A value of: ● 255 (the default) used to access the Ethernet communication module itself ● 0...254 identifies the device number of the target device, behind a Modbus TCP to Modbus gateway
	Number	(read-only) The integer associated with the service to be performed by the explicit message. An integer from 0...255.
	Name	Select the service the explicit message is intended to perform. In this example **ReadHoldingRegisters**
	Repeat 500ms	Select this check box to re-send the explicit message every 500 ms. Leave this check box de-selected.
4	After your explicit message is configured, click **Send to Device**. The **Response** area displays any data sent to the configuration tool by the target device in hexadecimal format. The **Status** area displays messages indicating whether or not the explicit message has succeeded.	
5	Click **Close** to close the window.	

Chapter 9
Implicit Messaging

EtherNet/IP Implicit Messaging

Introduction

The recommended RPI for EtherNet/IP implicit message connections are 1/2 of MAST cycle time. If the resulting RPI is less than 25 ms, the implicit message connections may be adversely affected when the diagnostic features of the 140NOC78•00 module are accessed via explicit messaging or DTM.

In this situation, the following timeout multiplier settings are recommended. Refer to the *Configuring Communication Module Connection Settings* *(see page 178)* topic to set the timeout multiplier.

RPI (ms)	Recommended Timeout Multiplier	Connection Timeout
2	64	128
5	32	160
10	16	160
20	8	160
25	4	100

NOTE: If you use RPI values lower than recommended, unnecessary bandwidth is consumed on the network, and the module system performance is impacted.

Chapter 10
Diagnostics

Overview

This chapter describes the diagnostics for the Quantum EIO modules. For details on diagnostics at the system level, refer to the systems diagnostics topic in the *Quantum Ethernet I/O System Planning Guide*.

What Is in This Chapter?

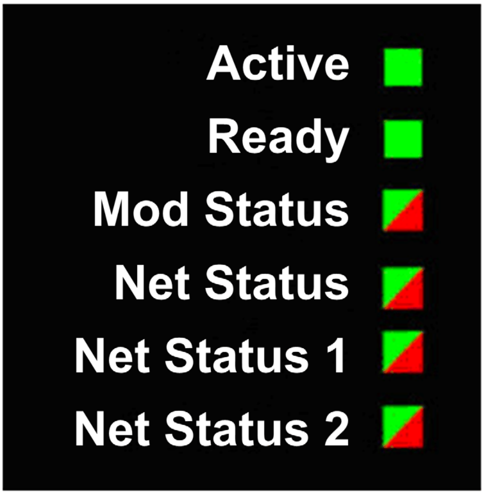
This chapter contains the following sections:

Section 10.1
LED Indicators

LED Indicators on the 140NOC78•00 Module

Display

These LEDs are on the front of the 140NOC78•00 module:



NOTE: The Net Status 1 and Net Status 2 LEDs are not functional for the 140NOC78000 distributed I/O head module.

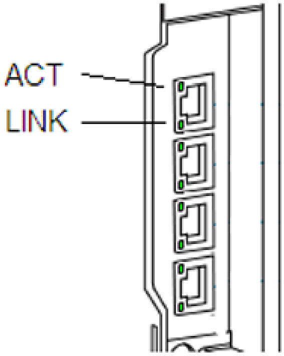
Indications

LED conditions:

Description		Active	Ready	Mod Status		Net Status		Net Status 1		Net Status 2	
		green	green	green	red	green	red	green	red	green	red
general	component not operating	—	off	off	on	off	off	off	off	off	off
	invalid configuration	—	off	off	flash	off	off	off	off	off	off
	not configured	—	off	flash	off	off	off	off	off	off	off
	configured	—	blink	on	off	on/flash	off	on/flash	off	on/flash	off
	no/default module MAC	—	blink 2	off	flash	off	off	off	off	off	off
	no/default port MAC	—	blink 2	off	flash	off	off	off	off	off	off
	no link	—	blink 3	—	—	—	—	—	—	—	—
power-up sequence	blink (.25 sec on; .25 sec off)	5	6	1	2	3	4	5	6	7	8
IP address	duplicate IP	—	blink 4	on	off	off	on	off	on	off	on
	waiting for IP	—	blink 5	on	off	off	off	off	off	off	off
	default IP address assigned	—	blink 6	on	off	off	off	off	off	off	off
	configured IP address assigned	—	on	on	off	flash	off	flash	off	flash	off
	invalid configuration	—	blink 7	on	off	off	off	off	off	off	off
I/O data commun-ication	no I/O or CIP connections	—	on	on	off	flash	off	flash	off	flash	off
	at least one I/O data connection to a remote I/O drop	—	on	on	off	on	off	on	off	on	off
	at least one CIP connection	—	on	on	off	off	flash	off	flash	off	flash

NOTE: The 140NOC78100 control head module has the unique functionality of providing multiple Ethernet network interfaces. The Ready LED indicates the status on **any** of the configured Ethernet network interfaces. For example, when the interlink cable is disconnected, the Ready LED flashes 5 times, even though the 140NOC78100 module is still connected to the control network.

Ethernet Port Indications



These LEDs report the status of the Ethernet port:

Name	Color	Status	Description
LINK (valid for ETH 1 and ETH 2 only)	green	on	100 Mbps link detected
	yellow	on	10 Mbps link detected
	—	off	no detected link
ACT	green	blinking	active Ethernet link (transmit or receive)
	green	off	inactive Ethernet link

Section 10.2
Diagnostics Available through the CPU

System Diagnostics

Introduction

System diagnostics are performed locally on the CPU with system bits (%S) and system words (%SW).

Local Rack Diagnostics

Local rack diagnostics are accessible for 140 NOC 78• 00, 140 NOC 771 ••, and 140 NOE 771 •• modules within the standard system words (%SW180 to %SW183).

System Bits and Words

This table describes new or modified system bits and words that represent detected errors:

System Bits/Words	Symbol	Description
%S117	EIOERR	detected remote I/O error on the Ethernet I/O network
%SW101	EIO_ CCOTF_COUNT	EIO CCOTF counting status register
%SW108	FORCED_DISCRETE_COUNT	forced bit counting status register
%SW152 ... %SW153	EIO_DROP_ERROR	detected Ethernet remote I/O drop status The bit is set to 0 if at least one I/O module in the drop as a detected error. The bit is set to 1 if all modules are operating properly. ● %SW152.0: drop #1 ● %SW152.1: drop #2 ● .... ● %SW153.14: drop #31
%SW172 ... %SW175	EIO_CONNECT_STATUS	Ethernet I/O communication health status for drops in standalone and primary systems
%SW176 ... %SW179	SDBY_EIO_CONNECT_STATUS	Ethernet I/O communication health status for drops in standby systems
%SW641 ... %SW702	EIO_MOD_HEALTH	Ethernet remote I/O module health bit status

NOTE: Refer to the *EcoStruxure™ Control Expert, System Bits and Words, Reference Manual* for a detailed explanation of system bits and words.

Section 10.3
Diagnostics Available through Modbus/TCP

Modbus Diagnostic Codes

Supported Diagnostic Codes

Modbus function code 3 provides access to a variety of diagnostic functions, including basic network diagnostics, Ethernet port diagnostics, and Modbus port 502 diagnostics.

To access the function code 3 diagnostics from the local device, set the **unit ID** to 100.

The following modules support these Modbus diagnostic codes.
- Quantum remote I/O head module (140CRP31200)
- Quantum distributed I/O head module (140NOC78000)
- Quantum control head module (140NOC78100)
- Quantum remote I/O adapter module (140CRA31200)
- M340 distributed I/O head module (BMXNOC0401)
- M340 remote I/O adapter module (BMXCRA31200)
- Quantum IEC 61850 module (140NOP85000)
- M580 communications module (BMENOC03•1)
- M580 communications module (BMENOP0300)

Modbus Function Code 3: Basic Network Diagnostics

Basic network diagnostics start at address 40001(decimal) as described in the following table.

Starting Address (Decimal)	Length (Words)	Register Byte Order		Comments
		MS BYTE	LS BYTE	
40001	2	MS Byte 00	Byte 01	Basic network diagnostic validity
		Byte 02	LS Byte	
40003	1	MS Byte	LS Byte 03	Communication global status
40004	1	MS Byte	LS Byte	Supported communication services
40005	1	MS Byte	LS Byte	Status of communication services
40006	2	IP 1	IP 2	IP address (IP1.IP2.IP3.IP4)
		IP 3	IP 4	
40008	2	SM 1	SM 2	Subnet mask (SM1.SM2.SM3.SM4)
		SM 3	SM 4	

Starting Address (Decimal)	Length (Words)	Register Byte Order		Comments
		MS BYTE	LS BYTE	
40010	2	GW 1	GW 2	Default gateway (GW1.GW2.GW3.GW4)
		GW 3	GW 4	
40012	3	MAC 1	MAC 2	MAC address (MAC1:MAC2:MAC3:MAC4:MAC5:MAC6.
		MAC 3	MAC 4	
		MAC 5	MAC 6	
40015	3	MS Byte 00	01	Ether frame format capability / configuration / operational
		02	03	
		04	LS Byte 05	
40018	2	C00	C01	Ethernet receive frames OK
		C02	C03	
40020	2	C00	C01	Ethernet transmit frames OK
		C02	C03	
40022	1	MS Byte	LS Byte	Number of open client connections
40023	1	MS Byte	LS Byte	Number of open server connections
40024	2	C00	C01	Number of Modbus detected error messages sent
		C02	C03	
40026	2	C00	C01	Number of Modbus messages sent
		C02	C03	
40028	2	C00	C01	Number of Modbus messages received
		C02	C03	
40030	8	Char 1	Char 2	Device name
		Char 3	Char 4	
		Char 5	Char 6	
		Char 7	Char 8	
		Char 9	Char 10	
		Char 11	Char 12	
		Char 13	Char 14	
		Char 15	Char 16	
40038	2	MS Byte 00	Byte 01	IP assignment mode capability / operational
		Byte 02	LS Byte 03	

Example: Reading Basic Network Diagnostics with Modbus Function Code 3

Here is an example of how to read registers 40018 and 40019, the *Ethernet receive frames OK* count registers. The request contains 7 bytes. The starting address, shown as a hex value in byte 2 below, is calculated as follows:

40018 – 40001 = 17 dec = (11 hex)

The number of registers to be diagnosed (2 hex) is shown in byte 4:

Byte Number	Value
0	Function code = 03 (hex)
1	Starting Address Hi = 00 (hex)
2	Starting Address Low = 11 (hex)
3	No. of Registers Hi = 00 (hex)
4	No. of Registers Low = 2 (hex)
5	CRC high byte (inserted by the Modbus sending application)
6	CRC low byte (inserted by the Modbus sending application)

The normal response is returned in 8 bytes. In this example, the expected response is 14229 hex; this value is shown in bytes 2 through 5 of the response:

Byte Number	Value
0	Function code = 03 (hex)
1	Byte count = 4 (hex)
2	1st register data, high byte = 00 (hex)
3	1st register data, low byte = 01 (01 hex)
4	2nd register data, high byte = 42 (hex)
5	2nd register data, low byte = 29 (hex)
6	CRC high byte
7	CRC low byte

Byte Number	Value
0	Original function code + 80 hex (= 83 hex)
1	Detected error code
2	CRC high byte
3	CRC low byte

For more information on Modbus function code 3 and other function codes, refer to the *Modicon Modbus Protocol Reference Guide* (PI-MBUS-300).

Modbus Function Code 3: Ethernet Internal Port Diagnostic Data

Internal port diagnostics start at address 40040 (decimal) as described in the following table.

Starting Address (Decimal)	Length (Words)	Register Byte Order		Comments
		MS BYTE	LS BYTE	
40040	1	MS Byte	LS Byte	Internal port Diagnostics Data Validity
40041	1	MS Byte	LS Byte	Internal port Logical/Physical Port Number
40042	1	MS Byte	LS Byte	Internal port Ethernet Control Capability
40043	1	MS Byte	LS Byte	Internal port Link Speed Capability
40044	1	MS Byte	LS Byte	Internal port Ethernet Control Configuration
40045	1	MS Byte	LS Byte	Internal port Link Speed Configuration
40046	1	MS Byte	LS Byte	Internal port Ethernet Control Operational
40047	1	MS Byte	LS Byte	Internal port Link Speed Operational
40048	3	MAC 1	MAC 2	MAC Address (MAC1:MAC2:MAC3:MAC4:MAC5:MAC6)
		MAC 3	MAC 4	
		MAC 5	MAC 6	
40051	2	MSB C00	C01	Internal port Media Counters Data Validity
		C02	LSB C03	
40053	2	MSB C00	C01	Internal port Num Frames Transmitted OK
		C02	LSB C03	
40055	2	MSB C00	C01	Internal port Num Frames Received OK
		C02	LSB C03	
40057	2	MSB C00	C01	Internal port Num Ether Collisions
		C02	LSB C03	
40059	2	MSB C00	C01	Internal port Carrier Sense Errors detected
		C02	LSB C03	
40061	2	MSB C00	C01	Internal port Num Ether Excessive Collisions
		C02	LSB C03	
40063	2	MSB C00	C01	Internal port CRC Errors detected
		C02	LSB C03	
40065	2	MSB C00	C01	Internal port FCS Errors detected
		C02	LSB C03	
40067	2	MSB C00	C01	Internal port Alignment Errors detected
		C02	LSB C03	
40069	2	MSB C00	C01	Internal port Num Internal MAC Tx Errors detected
		C02	LSB C03	

Starting Address (Decimal)	Length (Words)	Register Byte Order		Comments
		MS BYTE	LS BYTE	
40071	2	MSB C00	C01	Internal port Late Collisions
		C02	LSB C03	
40073	2	MSB C00	C01	Internal port Num Internal MAC Rx Errors detected
		C02	LSB C03	
40075	2	MSB C00	C01	Internal port Multiple Collisions
		C02	LSB C03	
40077	2	MSB C00	C01	Internal port Single Collisions
		C02	LSB C03	
40079	2	MSB C00	C01	Internal port Deferred Transmissions
		C02	LSB C03	
40081	2	MSB C00	C01	Internal port Frames Too Long
		C02	LSB C03	
40083	2	MSB C00	C01	Internal port Frames Too Short
		C02	LSB C03	
40085	2	MSB C00	C01	Internal port SQE Test Error detected
		C02	LSB C03	
40087	1	MS Byte	LS Byte	Internal port Interface Label Length
40088	32	IL char64	IL char63	Internal port Interface Label characters
40089		IL char62	IL char61	
...		...	...	
40118		IL char04	IL char03	
40119		IL char02	IL char01	
40120	1	MS Byte	LS Byte	Internal port Interface Counters Diagnostic Validity
40121	2	MSB C00	C01	Internal port Num Octets Received
		C02	LSB C03	
40123	2	MSB C00	C01	Internal port Num Unicast Packets Received
		C02	LSB C03	
40125	2	MSB C00	C01	Internal port Num Non Unicast Packets Received
		C02	LSB C03	
40127	2	MSB C00	C01	Internal port Num Inbound Packets Discarded
		C02	LSB C03	
40129	2	MSB C00	C01	Internal port Num Inbound Packets Error detected
		C02	LSB C03	

Starting Address (Decimal)	Length (Words)	Register Byte Order		Comments
		MS BYTE	LS BYTE	
401331	2	MSB C00	C01	Internal port Num Inbound Packets Unknown
		C02	LSB C03	
40133	2	MSB C00	C01	Internal port Num Octets Sent
		C02	LSB C03	
40135	2	MSB C00	C01	Internal port Num Unicast Packets Sent
		C02	LSB C03	
40137	2	MSB C00	C01	Internal port Num Non Unicast Packets Sent
		C02	LSB C03	
40139	2	MSB C00	C01	Internal port Num Outbound Packets Discarded
		C02	LSB C03	
40141	2	MSB C00	C01	Internal port Num Outbound Packets Error detected
		C02	LSB C03	

Modbus Function Code 3: Ethernet Port 1 Diagnostic Data

Port 1 diagnostics start at address 40143 (decimal). As described in the following table, port 1 diagnostic data fields are the same as for the internal port, with the appropriate starting address offset.

Starting Address	Length	Register Byte Order		Comments
		MS BYTE	LS BYTE	
40143	1	MS Byte	LS Byte	Port 1 Port Diagnostics Data Validity
...	...	...	...	...
40244	2	MSB C00	C01	Port 1 Num Outbound Packets Error detected
		C02	LSB C02	

Modbus Function Code 3: Ethernet Backplane Port Diagnostic Data

Backplane port diagnostics start at address 40246 (decimal). As described in the following table, backplane port diagnostic data fields are the same as for the internal port and port 1, with the appropriate starting address offset.

Starting Address	Length	Register Byte Order		Comments
		MS BYTE	LS BYTE	
40246	1	MS Byte	LS Byte	Backplane Port Diagnostics Data Validity
...	...	...	...	...
40347	2	MSB C00	C01	Backplane Port Num Outbound Packets Error detected
		C02	LSB C02	

Modbus Function Code 3: Ethernet Port 3 Diagnostic Data

Port 3 diagnostics start at address 40349 (decimal). As described in the following table, port 3 diagnostic data fields are the same as for the internal port and ports 1 and 2, with the appropriate starting address offset.

Starting Address	Length	Register Byte Order		Comments
		MS BYTE	LS BYTE	
40349	1	MS Byte	LS Byte	Port 3 Port Diagnostics Data Validity
...	...	...	...	...
40450	2	MSB C00	C01	Port 3 Num Outbound Packets Error detected
		C02	LSB C03	

Modbus Function Code 3: Ethernet Port 4 Diagnostic Data

Port 4 diagnostics start at address 40452 (decimal).As described in the following table, port 4 diagnostic data fields are the same as for the internal port and ports 1-3, with the appropriate starting address offset.

Starting Address	Length	Register Byte Order		Comments
		MS BYTE	LS BYTE	
40452	1	MS Byte	LS Byte	Port 4 Port Diagnostics Data Validity
...	...	...	...	...
40553	2	MSB C00	C01	Port 4 Num Outbound Packets Error detected
		C02	LSB C03	

Ethernet Port Not Present

If an Ethernet port is not physically present on the device, the relevant Modbus registers will return data = 0.

Modbus Function Code 3: Modbus TCP Port 502 Diagnostic Data

Modbus TCP port 502 diagnostics start at address 40555 (decimal) as described in the following table.

Starting Address	Length	Register Byte Order		Comments	
		MS BYTE	LS BYTE		
40555	2	MS Byte 00	Byte 01	Modbus TCP/Port 502 Diagnostic Data Validity	
		Byte 02	LS Byte 03		
40557	1	MS Byte	LS Byte	Port 502 Status	
40558	1	MS Byte	LS Byte	Num Open Connections	
40559	2	MSB C00	C01	Num MB Messages Sent	
		C02	LSB C03		
40561	2	MSB C00	C01	Num MB Messages Received	
		C02	LSB C03		
40563	1	MS Byte	LS Byte	Num MB Open Client Connections	
40564	1	MS Byte	LS Byte	Num MB Open Server Connections	
40565	1	MS Byte	LS Byte	Max Num Connections	
40566	1	MS Byte	LS Byte	Max Num Client Connections	
40567	1	MS Byte	LS Byte	Max Num Server Connections	
40568	2	MSB C00	C01	Num MB Detected Error Messages Sent	
		C02	LSB C03		
40570	1	MS Byte	LS Byte	Num Open Priority Connections	
40571	1	MS Byte	LS Byte	Max Num Priority Connections	
40572	1	MS Byte	LS Byte	Num Entries in Unauthorized Table	
40573	2	MSB - IP1	IP2	Remote IP Address 1	Connection 1
		IP3	LSB - IP4		
40575	1	MS Byte	LS Byte	Num Attempts to Open Unauthorized Connection 1	
40576	2	MSB - IP1	IP2	Remote IP Address 2	Connection 2
		IP3	LSB - IP4		
40578	2	MS Byte	LS Byte	Num Attempts to Open Unauthorized Connection 2	
...	...	...	...	...	
40663	2	MSB - IP1	IP2	Remote IP Address 31	Connection 31
		IP3	LSB - IP4		
40665	1	MS Byte	LS Byte	Num Attempts to Open Unauthorized Connection 31	

Starting Address	Length	Register Byte Order		Comments	
		MS BYTE	LS BYTE		
40666	2	MSB - IP1	IP2	Remote IP Address 32	Connection 32
		IP3	LSB - IP4		
40668	1	MS Byte	LS Byte	Num Attempts to Open Unauthorized Connection 32	

Modbus Function Code 3: Modbus TCP Port 502 Connection Table Data

Modbus TCP port 502 connection table data starts at address 40669 (decimal) as described in the following table.

Starting Address	Length	Register Byte Order		Comments	
		MS BYTE	LS BYTE		
40669	1	MS Byte	LS Byte	Connection Table Validity	
40670	1	MS Byte	LS Byte	Number of Entries	
40671	1	MS Byte	LS Byte	Starting Entry Index	
40672	1	MS Byte	LS Byte	Connection 1 Index	Connection 1
40673	2	IP1	IP2	Connection 1 Remote IP Address	
		IP3	IP4		
40675	1	MS Byte	LS Byte	Connection 1 Remote Port Number	
40676	1	MS Byte	LS Byte	Connection 1 Local Port Number	
40677	1	MS Byte	LS Byte	Num MB Messages Sent on Connection 1	
40678	1	MS Byte	LS Byte	Num MB Messages Received on Connection 1	
40679	1	MS Byte	LS Byte	Num MB Detected Error Messages Sent on Connection 1	
40680	1	MS Byte	LS Byte	Connection 2 Index	Connection 2
40681	2	IP1	IP2	Connection 2 Remote IP Address	
		IP3	IP4		
40683	1	MS Byte	LS Byte	Connection 2 Remote Port Number	
40684	1	MS Byte	LS Byte	Connection 2 Local Port Number	
40685	1	MS Byte	LS Byte	Num MB Messages Sent on Connection 2	
40686	1	MS Byte	LS Byte	Num MB Messages Received on Connection 2	
40687	1	MS Byte	LS Byte	Num MB Detected Error Messages Sent on Connection 2	

Starting Address	Length	Register Byte Order		Comments	
		MS BYTE	LS BYTE		
...	...	...	...	...	
41168	1	MS Byte	LS Byte	Connection 63 Index	Connection 63*
41169	2	IP1	IP2	Connection 63 Remote IP Address	
		IP3	IP4		
41171	1	MS Byte	LS Byte	Connection 63 Remote Port Number	
41172	1	MS Byte	LS Byte	Connection 63 Local Port Number	
41173	1	MS Byte	LS Byte	Num MB Messages Sent on Connection 63	
41174	1	MS Byte	LS Byte	Num MB Messages Received on Connection 63	
41175	1	MS Byte	LS Byte	Num MB Detected Error Messages Sent on Connection 63	
41176	1	MS Byte	LS Byte	Connection 64 Index	Connection 64*
41177	2	IP1	IP2	Connection 64 Remote IP Address	
		IP3	IP4		
415179	1	MS Byte	LS Byte	Connection 64 Remote Port Number	
41180	1	MS Byte	LS Byte	Connection 64 Local Port Number	
41181	1	MS Byte	LS Byte	Num MB Messages Sent on Connection 64	
41182	1	MS Byte	LS Byte	Num MB Messages Received on Connection 64	
41183	1	MS Byte	LS Byte	Num MB Detected Error Messages Sent on Connection 64	
*140 CRA 312 10 and BMX CRA 312 •0 RIO adapter modules support a maximum of 8 connections.					

Section 10.4
Diagnostics Available through EtherNet/IP CIP Objects

Introduction

Quantum Ethernet I/O applications use CIP within a producer/consumer model to provide communication services in an industrial environment. This section describes the available CIP objects for Quantum EIO modules.

What Is in This Section?

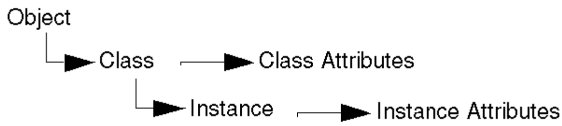
This section contains the following topics:

About CIP Objects

Overview

The Ethernet communication module can access CIP data and services located in connected devices. The CIP objects and their content depend on the design of each device.

CIP object data and content are exposed—and accessed—hierarchically in the following nested levels:



NOTE:
You can use explicit messaging to access these items:
- Access a collection of instance attributes by including only the class and instance values for the object in the explicit message.
- Access a single attribute by adding a specific attribute value to the explicit message with the class and instance values for the object.

This chapter describes the CIP objects that the Ethernet communication module exposes to remote devices.

Identity Object

Overview

The Identity object presents the instances, attributes and services described below.

Class ID

01

Instance IDs

The Identity object presents two instances:
- 0: class
- 1: instance

Attributes

Identity object attributes are associated with each instance, as follows:

Instance ID = 0 (class attributes):

Attribute ID	Description	GET	SET
01	Revision	X	—
02	Max Instance	X	—
X = supported — = not supported			

Instance ID = 1 (instance attributes):

Attribute ID		Description	Type	GET	SET
hex	dec				
01	01	Vendor ID	UINT	X	—
02	02	Device Type	UINT	X	—
03	03	Product Code	UINT	X	—
04	04	Revision	STRUCT	X	—
		Major	USINT		
		Minor	USINT		
X = supported — = not supported					

Attribute ID		Description	Type	GET	SET
hex	dec				
05	05	Status bit 2: 0x01=the module is configured bits 4-7: 0x03=no I/O connections established 0x06=at least 1 I/O connection in run mode 0x07=at least 1 I/O connection established, all in IDLE mode	Word	X	—
06	06	Serial Number	UDINT	X	—
07	07	Product Name	STRING	X	—
18	24	Modbus Identity	STRUCT	X	—
X = supported — = not supported					

Services

The Identity object performs the following services upon the listed object types:

Service ID		Description	Class	Instance	Notes
hex	dec				
01	01	Get_Attributes_All	X	X	Returns: • all class attributes (instance = 0) • instance attributes 1 to 7 (instance = 1)
0E	14	Get_Attribute_Single	X	X	Returns the value of the specified attribute.
X = supported — = not supported					

Assembly Object

Overview

The Assembly object consists of the attributes and services described below.

NOTE: You can send an explicit message to the Assembly object only when no other connections have been established that read from or write to this object. For example, you can send an explicit message to the Assembly object if a local slave instance is enabled, but no other module is scanning that local slave.

Class ID

04

Instance IDs

The Assembly object presents the following instance identifiers:

- 0: class
- 101, 102, 111, 112, 121, 122: instance

Attributes

The Assembly object consists of the following attributes:

Instance ID = 0 (class attributes):

Attribute ID	Description	GET	SET
01	Revision	X	—
02	Max Instance	X	—
03	Number of Instances	X	—
X = supported — = not supported			

Instance attributes:

Instance ID	Attribute ID	Description	Type	GET	SET
101	03	Local slave 1: T->O input data	Array of BYTE	X	—
102		Local slave 1: O>T	Array of BYTE	X	X
111		Local slave 2: T->O input data	Array of BYTE	X	—
112		Local slave 2: O>T	Array of BYTE	X	X
121		Local slave 3: T->O input data	Array of BYTE	X	—
122		Local slave 3: O>T	Array of BYTE	X	X
X = supported — = not supported					

Services

The CIP Assembly object performs these services upon the listed object types:

Service ID		Description	Class	Instance	Notes
hex	dec				
0E	14	Get_Attribute_Single	X	X	Returns the value of the specified attribute
10	16	Set_Attribute_Single[1]	—	X	Returns these values: 0E=attribute not settable: assembly is not o->T type 0F=permission denied: assembly is being used by an active connection 13=config too small: the Set_Attribute_Single command contains partial data 15=data too big: the Set_Attribute_Single command contains too much data
X = supported — = not supported					
1. When valid, the size of the data written to the Assembly object using the Set_Attribute_Single service equals the size of the Assembly object as configured in the target module.					

Connection Manager Object

Overview

The Connection Manager object presents the instances, attributes and services described below.

Class ID

06

Instance IDs

The Connection Manager object presents two instance values:
- 0: class
- 1: instance

Attributes

Connection Manager object attributes are associated with each instance, as follows:

Instance ID = 0 (class attributes):

Attribute ID	Description	GET	SET
01	Revision	X	—
02	Max Instance	X	—
X = supported — = not supported			

Instance ID = 1 (instance attributes):

Attribute ID		Description	Type	GET	SET	Value
hex	dec					
01	01	Open Requests	UINT	X	X	Number of Forward Open service requests received
02	02	Open Format Rejects	UINT	X	X	Number of Forward Open service requests that were rejected due to bad format
03	03	Open Resource Rejects	UINT	X	X	Number of Forward Open service requests that were rejected due to lack of resources
04	04	Open Other Rejects	UINT	X	X	Number of Forward Open service requests that were rejected for reasons other than bad format or lack of resources
05	05	Close Requests	UINT	X	X	Number of Forward Close service requests received
X = supported — = not supported						

Attribute ID		Description	Type	GET	SET	Value
hex	dec					
06	06	Close Format Requests	UINT	X	X	Number of Forward Close service requests that were rejected due to bad format
07	07	Close Other Requests	UINT	X	X	Number of Forward Close service requests that were rejected for reasons other than bad format
08	08	Connection Timeouts	UINT	X	X	Total number of connection timeouts that occurred in connections controlled by this connections manager
09	09	Connection Entry List	STRUCT	X	—	0 (Unsupported optional item
0B	11	CPU_Utilization	UINT	X	—	0 (Unsupported optional item
0C	12	MaxBuffSize	UDINT	X	—	0 (Unsupported optional item
0D	13	BufSize Remaining	UDINT	X	—	0 (Unsupported optional item
X = supported — = not supported						

Services

The Connection Manager object performs the following services on the listed object types:

Service ID		Description	Class	Instance	Notes
hex	dec				
01	01	Get_Attributes_All	X	X	Returns the value of all attributes.
0E	14	Get_Attribute_Single	X	X	Returns the value of the specified attribute.
X = supported — = not supported					

Modbus Object

Overview

The Modbus object converts EtherNet/IP service requests to Modbus functions, and Modbus exception codes to CIP General Status codes. It presents the instances, attributes and services described below.

Class ID

44 (hex), 68 (decimal)

Instance IDs

The Modbus object presents two instance values:
- 0: class
- 1: instance

Attributes

The Modbus object consists of the following attributes:

Instance ID = 0 (class attributes):

Attribute ID	Description	GET	SET
01	Revision	X	—
02	Max Instance	X	—
X = supported — = not supported			

Instance ID = 1 (instance attributes):

Attribute ID	Description	Type	GET	SET
—	No instance attributes are supported	—	—	—

Services

The Modbus object performs the following services upon the listed object types:

Service ID		Description	Class	Instance
hex	dec			
0E	14	Get_Attribute_Single	X	X
4B	75	Read_Discrete_Inputs	—	X
4C	76	Read_Coils	—	X
4D	77	Read_Input_Registers	—	X
4E	78	Read_Holding_Registers	—	X
4F	79	Write_Coils	—	X
50	80	Write_Holding_Registers	—	X
51	81	Modbus_Passthrough	—	X
X = supported — = not supported				

Quality Of Service (QoS) Object

Overview

The QoS object implements Differentiated Services Code Point (DSCP or *DiffServe*) values for the purpose of providing a method of prioritizing Ethernet messages. The QoS object presents the instances, attributes and services described below.

Class ID

48 (hex), 72 (decimal)

Instance IDs

The QoS object presents two instance values:
- 0: class
- 1: instance

Attributes

The QoS object consists of the following attributes:

Instance ID = 0 (class attributes):

Attribute ID	Description	GET	SET
01	Revision	X	—
02	Max Instance	X	—
X = supported — = not supported			

Instance ID = 1 (instance attributes):

Attribute ID	Description	Type	GET	SET	Value
04	DSCP Urgent	USINT	X	X	For CIP transport class 0/1 Urgent priority messages, default value = 55.
05	DSCP Scheduled	USINT	X	X	For CIP transport class 0/1 Urgent priority messages, default value = 47.
06	DSCP High	USINT	X	X	For CIP transport class 0/1 Urgent priority messages, default value = 43.
07	DSCP Low	USINT	X	X	For CIP transport class 0/1 Urgent priority messages, default value = 31.
08	DSCP Explicit	USINT	X	X	For CIP explicit messages (transport class 2/3 and UCMM), default value = 27.
X = supported — = not supported					

NOTE: A change in the instance attribute value takes effect on device re-start, for configurations made from flash memory.

Services

The QoS object performs the following services upon the listed object types:

Service ID		Description	Class	Instance
hex	dec			
0E	14	Get_Attribute_Single	X	X
10	16	Set_Attribute_Single	—	X
X = supported — = not supported				

TCP/IP Interface Object

Overview

The TCP/IP interface object presents the instances (per network), attributes and services described below.

Class ID

F5 (hex), 245 (decimal)

Instance IDs

The TCP/IP interface object presents 2 instance values:
- 0: class
- 1: instance

Attributes

TCP/IP interface object attributes are associated with each instance, as follows:

Instance ID = 0 (class attributes):

Attribute ID	Description	GET	SET
01	Revision	X	—
02	Max Instance	X	—
X = supported — = not supported			

Instance ID = 1 (instance attributes):

Attribute ID	Description	Type	GET	SET	Value
01	Status	DWORD	X	—	0x01
02	Configuration Capability	DWORD	X	—	0x01 = from BootP 0x11 = from flash 0x00 = other
03	Configuration Control	DWORD	X	X	0x01 = out-of-box default
04	Physical Link Object	STRUCT	X	—	
	Path Size	UINT			
	Path	Padded EPATH			
X = supported — = not supported					

Attribute ID	Description	Type	GET	SET	Value
05	Interface Configuration	STRUCT	X	X	0x00 = out-of-box default
	IP Address	UDINT			
	Network Mask	UDINT			
	Gateway Address	UDINT			
	Name Server	UDINT			
	Name Server 2	UDINT			
	Domain Name	STRING			
06	Host Name	STRING	X	—	
X = supported — = not supported					

Services

The TCP/IP interface object performs the following services upon the listed object types:

Service ID		Description	Class	Instance	Notes
hex	dec				
01	01	Get_Attributes_All	X	X	Returns the value of all attributes.
0E	14	Get_Attribute_Single	X	X	Returns the value of the specified attribute.
10	16	Set_Attribute_Single[1]	—	X	Sets the value of the specified attribute.
X = supported — = not supported					
1. The Set_Attribute_Single service can execute only when these preconditions are satisfied: ● Configure the Ethernet communication module to obtain its IP address from flash memory. ● Confirm that the PLC is in stop mode.					

Ethernet Link Object

Overview

The Ethernet Link object consists of the instances, attributes and services described below.

Class ID

F6 (hex), 246 (decimal)

Instance IDs

The Ethernet Link object presents the following instance values:

- 0: class
- 1: port 1
- 2: port 2
- 3: port 3
- 4: port 4

Attributes

The Ethernet Link object presents the following attributes:

Instance ID = 0 (class attributes):

Attribute ID	Description	GET	SET
01	Revision	X	—
02	Max Instance	X	—
03	Number of Instances	X	—
X = supported — = not supported			

Instance ID = 1 (instance attributes):

Attribute ID hex	dec	Description	Type	GET	SET	Value
01	01	Interface Speed	UDINT	X	—	Valid values include: 0, 10000000, 100000000
02	02	Interface Flags	DWORD	X	—	Bit 0: link status 0 = Inactive 1 = Active
						Bit 1: duplex mode 0 = half duplex 1 = full duplex
						Bits 2—4: negotiation status 3 = successfully negotiated speed and duplex 4 = forced speed and link
						Bit 5: manual setting requires reset 0 = automatic 1 = device need reset
						Bit 6: local hardware detected error 0 = no event 1 = event detected
03	03	Physical Address	ARRAY of 6 USINT	X	—	module MAC address
04	04	Interface Counters	STRUCT	X	—	
		In octets	UDINT			octets received on the interface
		In Ucast Packets	UDINT			unicast packets received on the interface
		In NUcast Packets	UDINT			non-unicast packets received on the interface
		In Discards	UDINT			inbound packets received on the interface, but discarded
		In Errors	UDINT			inbound packets with detected errors (does not include in discards)
		In Unknown Protos	UDINT			inbound packets with unknown protocol
		Out Octets	UDINT			octets sent on the interface
		Out Ucast Packets	UDINT			unicast packets sent on the interface
		Out NUcast Packets	UDINT			non-unicast packets sent on the interface
		Out Discards	UDINT			outbound packets discarded
		Out Errors	UDINT			outbound packets with detected errors
X = supported — = not supported						

Attribute ID		Description	Type	GET	SET	Value
hex	dec					
05	05	Media Counters	STRUCT	X	—	
		Alignment Errors	UDINT			frames that are not an integral number of octets in length
		FCS Errors	UDINT			bad CRC — frames received do not pass the FCS check
		Single Collisions	UDINT			successfully transmitted frames that experienced exactly 1 collision
		Multiple Collisions	UDINT			successfully transmitted frames that experienced more than 1 collision
		SQE Test Errors	UDINT			number of times the detected SQE test error is generated
		Deferred Transmissions	UDINT			frames for which first transmission attempt is delayed because the medium is busy
		Late Collisions	UDINT			number of times a collision is detected later than 512 bit times into the transmission of a packet
		Excessive Collisions	UDINT			frames that do not transmit due to excessive collisions
		MAC Transmit Errors	UDINT			frames that do not transmit due to a detected internal MAC sublayer transmit error
		Carrier Sense Errors	UDINT			times that the carrier sense condition was lost or not asserted when attempting to transmit a frame
		Frame Too Long	UDINT			frames received that exceed the maximum permitted frame size
		MAC Receive Errors	UDINT			frames not received on an interface due to a detected internal MAC sublayer receive error
X = supported — = not supported						

<table>
<tr><th colspan="2">Attribute ID</th><th>Description</th><th>Type</th><th>GET</th><th>SET</th><th>Value</th></tr>
<tr><th>hex</th><th>dec</th><th></th><th></th><th></th><th></th><th></th></tr>
<tr><td rowspan="4">06</td><td rowspan="4">06</td><td>Interface Control</td><td>STRUCT</td><td>X</td><td>X</td><td>API of the connection</td></tr>
<tr><td rowspan="2">Control Bits</td><td rowspan="2">WORD</td><td rowspan="3"></td><td rowspan="3"></td><td>Bit 0: Auto-negotiation
0 = disabled
1 = enabled
Note: When auto-negotiation is enabled, 0x0C (object state conflict) is returned when attempting to set either:
• forced interface speed
or
• forced duplex mode</td></tr>
<tr><td>Bit 1: forced duplex mode (if auto-negotiation bit = 0)
0 = half duplex
1 = full duplex</td></tr>
<tr><td>Forced Interface Speed</td><td>UINT</td><td>Valid values include:
10000000, 100000000
Note: Attempting to set any other value returns the detected error 0x09 (invalid attribute value)</td></tr>
<tr><td>10</td><td>16</td><td>Interface Label</td><td>SHORT_ STRING</td><td>X</td><td>—</td><td>A fixed textual string identifying the interface, that should include ‘internal’ for internal interfaces. Maximum number of characters is 64.</td></tr>
<tr><td colspan="7">X = supported
— = not supported</td></tr>
</table>

Services

The Ethernet Link object performs the following services upon the listed object types:

<table>
<tr><th colspan="2">Service ID</th><th rowspan="2">Description</th><th rowspan="2">Class</th><th rowspan="2">Instance</th></tr>
<tr><th>hex</th><th>dec</th></tr>
<tr><td>01</td><td>01</td><td>Get_Attributes_All</td><td>X</td><td>X</td></tr>
<tr><td>10</td><td>16</td><td>Set_Attribute_Single</td><td>—</td><td>X</td></tr>
<tr><td>0E</td><td>14</td><td>Get_Attribute_Single</td><td>X</td><td>X</td></tr>
<tr><td>4C</td><td>76</td><td>Get_and_Clear</td><td>—</td><td>X</td></tr>
<tr><td colspan="5">X = supported
— = not supported</td></tr>
</table>

EtherNet/IP Interface Diagnostics Object

Overview

The EtherNet/IP Interface Diagnostics object presents the instances, attributes and services described below.

Class ID

350 (hex), 848 (decimal)

Instance IDs

The EtherNet/IP Interface object presents two instance values:
- 0: class
- 1: instance

Attributes

EtherNet/IP Interface Diagnostics object attributes are associated with each instance, as follows:

Instance ID = 0 (class attributes):

Attribute ID	Description	GET	SET
01	Revision	X	—
02	Max Instance	X	—
X = supported — = not supported			

Instance ID = 1 (instance attributes):

Attribute ID	Description	Type	GET	SET	Value
01	Protocols Supported	UINT	X	—	
02	Connection Diagnostics	STRUCT	X	—	
	Max CIP IO Connections opened	UINT			Number of Class 1 connections opened since the last reset
	Current CIP IO Connections	UINT			Number of Class 1 connections currently opened
	Max CIP Explicit Connections opened	UINT			Number of Class 3 connections opened since the last reset
	Current CIP Explicit Connections	UINT			Number of Class 3 connections currently opened
	CIP Connections Opening Errors	UINT			Increments each time a Forward Open is not successful (Originator and Target)
	CIP Connections Timeout Errors	UINT			Increments when a connection times out (Originator and Target)
	Max EIP TCP Connections opened	UINT			Number of TCP connections (used for EIP, as client or server) opened since the last reset
	Current EIP TCP Connections	UINT			Number of TCP connections (used for EIP, as client or server) currently open
03	IO Messaging Diagnostics	STRUCT	X	X	
	IO Production Counter	UDINT			Increments each time a Class 0/1 message is sent
	IO Consumption Counter	UDINT			Increments each time a Class 0/1 message is received
	IO Production Send Errors Counter	UINT			Increments each time a Class 0/1 message is not sent
	IO Consumption Receive Errors Counter	UINT			Increments each time a consumption is received with a detected error
X = supported — = not supported					

Attribute ID	Description	Type	GET	SET	Value
04	Explicit Messaging Diagnostics	STRUCT	X	X	
	Class 3 Msg Send Counter	UDINT			Increments each time a Class 3 message is sent (client and server)
	Class 3 Msg Receive Counter	UDINT			Increments each time a Class 3 message is received (client and server)
	UCMM Msg Receive Counter	UDINT			Increments each time a UCMM message is sent (client and server)
	UCMM Msg Receive Counter	UDINT			Increments each time a UCMM message is received (client and server)
X = supported — = not supported					

Services

The EtherNet/IP Interface Diagnostics object performs the following services upon the listed object types:

Service ID		Description	Class	Instance	Notes
hex	dec				
01	01	Get_Attributes_All	X	X	Returns the value of all attributes.
0E	14	Get_Attribute_Single	—	X	Returns the value of the specified attribute.
4C	76	Get_and_Clear	—	X	Returns and clears the values of all instance attributes.
X = supported — = not supported					

EtherNet/IP IO Scanner Diagnostics Object

Overview

The EtherNet/IP IO Scanner Diagnostics object presents the instances, attributes and services described below.

Class ID

351 (hex), 849 (decimal)

Instance IDs

The EtherNet/IP IO Scanner Diagnostics object presents two instances:
- 0: class
- 1: instance

Attributes

EtherNet/IP IO Scanner Diagnostics object attributes are associated with each instance, as follows:

Instance ID = 0 (class attributes):

Attribute ID	Description	GET	SET
01	Revision	X	—
02	Max Instance	X	—
X = supported — = not supported			

Instance ID = 1 (instance attributes):

Attribute ID	Description	Type	GET	SET
01	IO Status Table	STRUCT	X	—
	Size	UINT		
	Status	ARRAY of UNINT		
X = supported — = not supported				

Services

The EtherNet/IP IO Scanner Diagnostics object performs the following services upon the listed object types:

Service ID		Description	Class	Instance	Notes
hex	dec				
01	01	Get_Attributes_All	X	X	Returns the value of all attributes.
0E	14	Get_Attribute_Single	X	X	Returns the value of the specified attribute.
X = supported — = not supported					

IO Connection Diagnostics Object

Overview

The IO Connection Diagnostics object presents the instances, attributes and services described below.

Class ID

352 (hex), 850 (decimal)

Instance IDs

The IO Connection Diagnostics object presents two instance values:
- 0: class
- 1...256: instance (The instance number is the connection number in the configuration.)

Attributes

IO Connection Diagnostics object attributes are associated with each instance, as follows:

Instance ID = 0 (class attributes):

Attribute ID	Description	GET	SET
01	Revision	X	—
02	Max Instance	X	—
X = supported — = not supported			

Instance ID = 1 to 256 (instance attributes):

Attribute ID	Description	Type	GET	SET	Value
01	IO Communication Diagnostics	STRUCT	X	X	
	IO Production Counter	UDINT			Increments at each production
	IO Consumption Counter	UDINT			Increments at each consumption
	IO Production Send Errors Counter	UINT			Increments each time a production is not sent
	IO Consumption Receive Errors Counter	UINT			Increments each time a consumption is received with a detected error
	CIP Connection Timeout Errors	UINT			Increments when a connection times out
	CIP Connection Opening Errors	UINT			Increments each time a connection is unable to open
	CIP Connection State	UINT			State of the Connection Bit
	CIP Last Error General Status	UINT			General status of the last error detected on the connection
	CIP Last Error Extended Status	UINT			Extended status of the last error detected on the connection
	Input Communication Status	UINT			Communication status of the inputs (see table, below)
	Output Communication Status	UINT			Communication status of the outputs (see table, below)
X = supported — = not supported					

Attribute ID	Description	Type	GET	SET	Value
02	Connection Diagnostics	STRUCT	X	X	
	Production Connection ID	UDINT			Connection ID for production
	Consumption Connection ID	UDINT			Connection ID for consumption
	Production RPI	UDINT			RPI for production
	Production API	UDINT			API for production
	Consumption RPI	UDINT			RPI for consumption
	Consumption API	UDINT			API for consumption
	Production Connection Parameters	UDINT			Connection parameters for production
	Consumption Connection Parameters	UDINT			Connection parameters for consumption
	Local IP	UDINT			—
	Local UDP Port	UINT			—
	Remote IP	UDINT			—
	Remote UDP Port	UINT			—
	Production Multicast IP	UDINT			Multicast IP used for production (or 0)
	Consumption Multicast IP	UDINT			Multicast IP used for consumption (or 0)
	Protocols Supported	UDINT			Protocol supported on the connection: 1 = EtherNet/IP
X = supported — = not supported					

The following values describe the structure of the instance attributes: *CIP Connection State*, *Input Communication Status*, and *Output Communication Status*:

Bit Number	Description	Values
15...3	*Reserved*	0
2	Idle	0 = no idle notification 1 = idle notification
1	Consumption inhibited	0 = consumption started 1 = no consumption
0	Production inhibited	0 = production started 1 = no production

Services

The EtherNet/IP Interface Diagnostics object performs the following services upon the listed object types:

Service ID		Description	Class	Instance	Notes
hex	dec				
01	01	Get_Attributes_All	X	X	Returns the value of all attributes.
0E	14	Get_Attribute_Single	—	X	Returns the value of the specified attribute.
4C	76	Get_and_Clear	—	X	Returns and clears the values of all instance attributes.
X = supported — = not supported					

EtherNet/IP Explicit Connection Diagnostics Object

Overview

The EtherNet/IP Explicit Connection Diagnostics object presents the instances, attributes and services described below.

Class ID

353 (hex), 851 (decimal)

Instance IDs

The EtherNet/IP Explicit Connection Diagnostics object presents two instance values:
- 0: class
- 1...*N*: instance (*N* = maximum concurrent number of explicit connections)

Attributes

EtherNet/IP Explicit Connection Diagnostics object attributes are associated with each instance, as follows:

Instance ID = 0 (class attributes):

Attribute ID hex	Description	Value	GET	SET
01	Revision	1	X	—
02	Max Instance	0...N	X	—
X = supported — = not supported				

Instance ID = 1 to *N* (instance attributes):

Attribute ID hex	Description	Type	GET	SET	Value
01	Originator connection ID	UDINT	X	—	Originator to target connection ID
02	Originator IP	UINT	X	—	
03	Originator TCP Port	UDINT	X	—	
04	Target connection ID	UDINT	X	—	Target to originator connection ID
05	Target IP	UDINT	X	—	
06	Target TCP Port	UDINT	X	—	
X = supported — = not supported					

Attribute ID hex	Description	Type	GET	SET	Value
07	Msg Send Counter	UDINT	X	—	Incremented each time a Class 3 CIP message is sent on the connection
08	Msg Receive counter	UDINT	X	—	Increments each time a Class 3 CIP message is received on the connection
X = supported — = not supported					

Services

The EtherNet/IP Explicit Connection Diagnostics object performs the following services upon the listed object type:

Service ID		Description	Class	Instance	Notes
hex	dec				
01	01	Get_Attributes_All	X	X	Returns the value of all attributes.
X = supported — = not supported					

EtherNet/IP Explicit Connection Diagnostics List Object

Overview

The EtherNet/IP Explicit Connection Diagnostics List object presents the instances, attributes and services described below.

Class ID

354 (hex), 852 (decimal)

Instance IDs

The EtherNet/IP Explicit Connection Diagnostics List object presents two instance values:

- 0: class
- 1...*N*: instance

Attributes

EtherNet/IP Explicit Connection Diagnostics List object attributes are associated with each instance, as follows:

Instance ID = 0 (class attributes):

Attribute ID	Description	GET	SET
01	Revision	X	—
02	Max Instance	X	—
X = supported — = not supported			

Instance ID = 1 to *N* (instance attributes):

Attribute ID	Description	Type	GET	SET	Value
01	Number of connections	UINT	X	—	Total number of opened explicit connections
02	Explicit Messaging Connections Diagnostic List	ARRAY of STRUCT	X	—	
	Originator connection ID	UDINT			O->T connection ID
	Originator IP	UINT			—
	Originator TCP port	UDINT			—
	Target connection ID	UDINT			T->O connection ID
	Target IP	UDINT			—
	Target TCP port	UDINT			—
	Msg Send counter	UDINT			Increments each time a Class 3 CIP message is sent on the connection
	Msg Receive counter	UDINT			Increments each time a Class 3 CIP message is received on the connection
X = supported — = not supported					

Services

The EtherNet/IP Explicit Connection Diagnostics object performs the following services upon the listed object types:

Service ID		Description	Class	Instance	Notes
hex	dec				
01	01	Get_Attributes_All	X	—	Returns the value of all attributes.
08	08	Create	X	—	—
09	09	Delete	—	X	—
4B	75	Explicit_Connections_Diagnostic_Read	—	X	—
X = supported — = not supported					

RSTP Diagnostics Object

Overview

The RSTP Diagnostics object presents the instances, attributes and services described below.

Class ID

355 (hex), 853 (decimal)

Instance IDs

The RSTP Diagnostics object presents these instance values:

- 0: class
- 1: instance

Attributes

RSTP Diagnostics object attributes are associated with each instance.

Instance ID = 0 (class attributes):

Attribute ID	Description	Type	GET	SET
01	Revision: This attribute specifies the current revision of the RSTP Diagnostic Object. The revision is increased by 1 at each new update of the object.	UINT	X	—
02	Max Instance: This attribute specifies the maximum number of instances that may be created for this object on a per device basis (for example, an RSTP Bridge). There is 1 instance for each RSTP port on a device.	UINT	X	—
X = supported — = not supported				

Instance ID = 1 to *N* (instance attributes):

Attribute ID	Description	Type	GET	CLEAR	Value
01	**Switch Status**	STRUCT	X	—	—
	Protocol Specification	UINT	X	—	Refer to RFC-4188 for attribute definitions and value range. In addition, the following value is defined: [4]: the protocol is IEEE 802.1D-2004 and IEEE 802.1W
	Bridge Priority	UDINT	X	—	Refer to RFC-4188 for attribute definitions and value range.
	Time Since Topology Change	UDINT	X	—	
	Topology Change Count	UDINT	X	—	Refer to RFC-4188 for attribute definitions and value range.
	Designated Root	String	X	—	Refer to RFC-4188 for attribute definitions and value range.
	Root Cost	UDINT	X	—	
	Root Port	UDINT	X	—	
	Max Age	UINT	X	—	
	Hello Time	UINT	X	—	
	Hold Time	UDINT	X	—	
	Forward Delay	UINT	X	—	
	Bridge Max Age	UINT	X	—	
	Bridge Hello Time	UINT	X	—	
	Bridge Forward Delay	UINT	X	—	
X = supported — = not supported					

Attribute ID	Description	Type	GET	CLEAR	Value
02	**Port Status**	STRUCT	X	X	—
	Port	UDINT	X	X	Refer to RFC-4188 for attribute definitions and value range.
	Priority	UDINT	X	X	
	State	UINT	X	X	
	Enable	UINT	X	X	
	Path Cost	UDINT	X	X	
	Designated Root	String	X	X	
	Designated Cost	UDINT	X	X	
	Designated Bridge	String	X	X	
	Designated Port	String	X	X	
	Forward Transitions Count	UDINT	X	X	Refer to RFC-4188 for attribute definitions and value range. Services: ● Get_and_Clear: The current value of this parameter is returned with the response message. ● other services: The current value of this parameter is returned without being cleared.
03	**Port Mode**	STRUCT	X	—	—
	Port Number	UINT	X	—	This attribute indicates the port number for a data query. The value range is configuration dependent. For a 4-port Ethernet device, as an instance, the valid range is 1...4.
	Admin Edge Port	UINT	X	—	This attribute indicates if this is a user-configured edge port: ● 1: true ● 2: false Other values are not valid.
	Oper Edge Port	UINT	X	—	This attribute indicates if this port is currently an edge port: ● 1: true ● 2: false Other values are not valid.
	Auto Edge Port	UINT	X	—	This attribute indicates if this port is a dynamically determined edge port: ● 1: true ● 2: false Other values are not valid.
X = supported — = not supported					

Services

The RSTP Diagnostics object performs these services:

Service ID		Description	Class	Instance	Notes
hex	dec				
01	01	Get_Attributes_All	X	X	This service returns: ● all attributes of the class ● all attributes of the instance of the object
02	02	Get_Attribute_Single	X	X	This service returns: ● the contents of a single attribute of the class ● the contents of the instance of the object as specified Specify the attribute ID in the request for this service.
32	50	Get_and_Clear	—	X	This service returns the contents of a single attribute of the instance of the object as specified. Then the relevant counter-like parameter(s) within the specified attribute are cleared. (Specify the attribute ID in the request for this service.)
X = supported — = not supported					

Service Port Control Object

Overview

The Service Port Control object is defined for port control purposes.

Class ID

400 (hex), 1024 (decimal)

Instance IDs

The Service Port Control object presents these instance Values:
- 0: class
- 1: instance

Attributes

Service Port Control object attributes are associated with each instance.

Required class attributes (instance 0):

Attribute ID	Description	Type	Get	Set
01	Revision	UINT	X	—
02	Max Instance	UINT	X	—
X = supported — = not supported				

Required instance attributes (instance 1):

Attribute ID		Description	Type	Get	Set	Value
hex	dec					
01	01	Port Control	UINT	X	X	0 (default): disabled 1: access port 2: port mirroring 3: extended port
02	02	Mirror	UINT	X	X	bit 0 (default): ETH2 port bit 1: ETH3 port bit 2: ETH4 port bit 3: internal port
X = supported — = not supported						

NOTE:
- If the SERVICE/EXTEND port is not configured for port mirroring, the mirror attribute is ignored. If the value of a parameter request is outside the valid range, the service request is ignored.
- In port mirroring mode, the SERVICE/EXTEND port acts like a read-only port. That is, you cannot access devices (ping, connection to Control Expert, etc.) through the SERVICE/EXTEND port.

Services

The Service Port Control object performs these services for these object types:

Service ID		Name	Class	Instance	Description
hex	dec				
01	01	Get_Attributes_All	X	X	Get all attributes in a single message.
02	02	Set_Attributes_All	—	X	Set all attributes in a single message.
0E	14	Get_Attribute_Single	X	X	Get a single specified attribute.
10	16	Set_Attribute_Single	—	X	Set a single specified attribute.
X = supported — = not supported					

Section 10.5
Diagnostics Available through Control Expert

Introduction

The Quantum EIO modules support online actions. Use the online actions to perform these tasks:
- Display EtherNet/IP objects for the head module or a remote EtherNet/IP device.
- View and edit the service/extend port configuration parameters for the head module/
- Ping the head module or a remote EtherNet/IP or Modbus TCP device to confirm it is active on the Ethernet network.
- Connect to a remote device to perform these actions:
  - View the remote device's default parameter settings.
  - View the remote device's current parameter settings.
  - Edit and download to the remote device its editable parameter settings.

What Is in This Section?

This section contains the following topics:

Using the Diagnostic Window

Introduction

Use the **Diagnostic** window to display:

- LED icons (in the left pane of the window) that indicate the operating status of modules, devices and connections
- pages (in the right pane of the window) that present diagnostic data for the following:
  - the communication module
  - local slave nodes activated for the communication module
  - EtherNet/IP connections between the communication module and a remote EtherNet/IP device

  Refer to the following topics for a description of the individual pages that are displayed in the right pane of the **Diagnostic** window.

NOTE: Before you can open the **Diagnostic** window, you must first connect the DTM for the target communication module to the physical module itself. To do this, select the module node in the **DTM Browser**, then select **Edit → Connect**.

To open the **Diagnostic** window:

Step	Action
1	In the **DTM Browser**, select the communication module and click the right mouse button. A pop-up menu opens.
2	In the menu, select **Device menu → Diagnostic**.

Diagnostic LED Icons

During the time that a communication module DTM is connected to the physical communication module, Control Expert sends an explicit message request once per second to detect the state of the communication module and of all the remote devices and EtherNet/IP connections linked to that module.

Control Expert places one of the following status icons over the module, device or connection in the left pane of the **Diagnostic** window to indicate its current status:

This icon...	Indicates the following state for a...	
	Communication module	Connection to a remote device
	Run state	The health bit for every EtherNet/IP connection and Modbus TCP request, to a remote device or to a sub-device or module, is set to active (1).
	One of the following states: • unknown • started • stopped • not connected	The health bit for at least one EtherNet/IP connection or Modbus TCP request, to a remote device or to a sub-device or module, is set to inactive (0).

Communication Module Ethernet Diagnostics

Introduction

Use the **Ethernet Diagnostic** page to display either dynamically generated or static data for the communication module's Ethernet port(s). The number of ports on the module determines the number of columns displayed in this page.

Use the **Refresh Every 500ms** checkbox to display static or dynamic data, as follows:

When the checkbox is...	This page...
Selected	• Displays data that is dynamically updated every 500 ms, and • Increments the number at the top of the table each time data is refreshed
De-selected	• Displays static data, and • Does not Increment the number at the top of the table, which instead displays a constant value

NOTE: Before you can open the **Diagnostic** window, you first must connect the DTM for the target communication module to the physical module itself. To do this, select the module node in the **DTM Browser**, then select **Edit → Connect**.

To open this page:

Step	Action
1	In the **DTM Browser**, select the communication module and click the right mouse button. A pop-up menu opens.
2	In the menu, select **Device menu → Diagnostic**.
3	In the left pane of the **Diagnostic** window, select the communication module node.
4	Click on the **Ethernet Diagnostic** tab to open that page.

Ethernet Diagnostic Parameters

The **Ethernet Diagnostic** page displays the following parameters for each communication module port:

Parameter	Description
General parameters:	
Interface Speed	Valid values include: 0, 10000000, 100000000 in Mbits/s
Interface Flags	Bit 0—Link Status: 0 = Inactive; 1 = Active
	Bit 1—Duplex Mode (see below)
	Bits 2...4—Negotiation Status (see below)
	Bit 5—Manual Setting Requires Reset (see below)
	Bit 6—Local Hardware Fault (see below)
Duplex Mode	0 = half duplex; 1 = full duplex
Negotiation Status	3 = successfully negotiated speed and duplex 4 = forced speed and link
Manual Setting Requires Reset	0 = automatic; 1 = device requires reset
Local Hardware Fault	0 = no event; 1 = event detected
Physical Address	Module MAC Address
Input parameters:	
Octets	Octets received on the interface
Unicast Packets	Unicast packets received on the interface
Non-Unicast Packets	Non-unicast packets received on the interface
Discards	Inbound packets received on the interface, but discarded
Errors	Inbound packets that contain errors (does not include In Discards)
Unknown Protocols	Inbound packets with unknown protocol
Output parameters:	
Octets	Octets received on the interface
Unicast Packets	Unicast packets received on the interface
Non-Unicast Packets	Non-unicast packets received on the interface
Discards	Inbound packets received on the interface, but discarded
Errors	Outbound packets that contain errors (does not include In Discards)
Unknown Protocols	Outbound packets with unknown protocol

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

Communication Module Bandwidth Diagnostics

Introduction

Use the **Bandwidth** page to display either dynamically generated or static data for the communication module's bandwidth usage.

Use the **Refresh Every 500ms** checkbox to display static or dynamic data, as follows:

When the checkbox is...	This page...
Selected	• Displays data that is dynamically updated every 500 ms, and • Increments the number at the top of the table each time data is refreshed
De-selected	• Displays static data, and • Does not Increment the number at the top of the table, which instead displays a constant value

NOTE: Before you can open the **Diagnostic** window, you must first connect the DTM for the target communication module to the physical module itself. To do this, select the module node in the **DTM Browser**, then select **Edit → Connect**.

To open this page:

Step	Action
1	In the **DTM Browser**, select the communication module and click the right mouse button. A pop-up menu opens.
2	In the menu, select **Device menu → Diagnostic**.
3	In the left pane of the **Diagnostic** window, select the communication module node.
4	Click on the **Bandwidth** tab to open that page.

Bandwidth Diagnostic Parameters

The **Bandwidth Diagnostic** page displays the following parameters for the communication module:

Parameter	Description
I/O - Scanner:	
EtherNet/IP Sent	The number of EtherNet/IP packets the module has sent, since the last reset, in packets/second.
EtherNet/IP Received	The number of EtherNet/IP packets the module has received, since the last reset, in packets/second.
Modbus TCP Requests	The number of Modbus TCP requests the module has sent, since the last reset, in packets/second.
Modbus TCP Responses	The number of Modbus TCP responses the module has received, since the last reset, in packets/second.
I/O - Adapter:	
EtherNet/IP Sent	The number of EtherNet/IP packets the module has sent—in the role of a local slave—since the last reset, in packets/second.
EtherNet/IP Received	The number of EtherNet/IP packets the module has received—in the role of a local slave—since the last reset, in packets/second.
I/O - Module	
Module Capacity	The maximum number of packets that the module can process, in packets per second.
Module Utilization	The percentage of communication module capacity being used by the application.
Messaging - Client:	
EtherNet/IP Activity	The number of I/O messages sent by the module—using the EtherNet/IP protocol—since last reset, in packets per second.
Modbus TCP Activity	The number of I/O messages sent by the module—using the Modbus TCP protocol—since last reset, in packets per second.
Messaging - Server:	
EtherNet/IP Activity	The number of I/O messages received by the module—using the EtherNet/IP protocol—since last reset, in packets per second.
Modbus TCP Activity	The number of I/O messages received by the module—using the Modbus TCP protocol—since last reset, in packets per second.
Module:	
Processor Utilization	The percent of Ethernet communication module processor capacity used by the present level of communication activity.

Local Slave / Connection Diagnostics

Introduction

Use the **Local Slave Diagnostic** page and the **Connection Diagnostic** page to display I/O status and production/consumption information for selected local slave or connection.

Use the **Refresh Every 500ms** checkbox to display static or dynamic data, as follows:

When the checkbox is...	This page...
Selected	• Displays data that is dynamically updated every 500 ms, and • Increments the number at the top of the table each time data is refreshed
De-selected	• Displays static data, and • Does not Increment the number at the top of the table, which instead displays a constant value

NOTE: Before you can open the **Diagnostic** window, you first must connect the communication module or remote device DTM to the physical module or device. To do this, select the appropriate node in the **DTM Browser**, then select **Edit → Connect**.

To open this page:

Step	Action
1	In the **DTM Browser**, select the communication module and click the right mouse button. A pop-up menu opens.
2	In the menu, select **Device menu → Diagnostic**.
3	In the left pane of the **Diagnostic** window, click on one of the following: • the communication module node, or • a connection node
4	Depending upon your selection in step 3, above, click on either the **Local Slave Diagnostic** tab or the **Connection Diagnostic** tab to open that page.

Local Slave / Connection Diagnostic Parameters

This page displays the following diagnostic parameters for the selected local slave or connection:

Parameter	Description
Status:	
Input	An integer representing input status.
Output	An integer representing output status.
General	An integer representing basic connection status.
Extended	An integer representing extended connection status.
Counter:	
Frame Error	Increments each time a frame is not sent by missing resources or is impossible to send.
Time-Out	Increments each time a connection times out.
Refused	Increments when connection is refused by the remote station.
Production	Increments each time a message is produced.
Consumption	Increments each time a message is consumed.
Production Byte	Total of produced messages, in bytes, since the communication module was last reset.
Consumption Byte	Total of consumed messages, in bytes, since the communication module wa last reset.
Theoretical Packets per second	Packets per second calculated sing current configuration value.
Real Packets per second	Actual number of packets per second generated by this connection.
Diagnostic:	
CIP Status	An integer representing CIP status.
Extended Status	An integer representing extended CIP status.
Production Connection ID	The connection ID.
Consumption Connection ID	The connection ID.
O -> T API	Accepted packet interval (API) of the output connection.
T -> O API	Accepted packet interval (API) of the input connection.
O -> T RPI	Requested packet interval (RPI) of the output connection.
T -> O RPI	Requested packet interval (RPI) of the input connection.

Parameter	Description
Socket Diagnostics:	
Socket ID	Internal Identification of the socket.
Remote IP Address	IP address of the remote station, for this connection.
Remote Port	Port number of the remote station, for this connection.
Local IP Address	IP address of the communication module, for this connection.
Local Port	Port number of the communication module, for this connection.
Production:	
Sequence Number	The number of the sequence in the production.
Max Time	Maximum time between two produced messages.
Min Time	Minimum time between two produced messages.
RPI	Current production time.
Over Run	Increments each time a produced message exceeds RPI.
Under Run	Increments each time a produced message is less than RPI.
Consumption:	
Sequence Number	The number of the sequence in the consumption.
Max Time	Maximum time between two consumptions.
Min Time	Minimum time between two consumptions.
RPI	Current consumption time.
Over Run	Increments each time a consumed message exceeds RPI.
Under Run	Increments each time a consumed message is less than RPI.

Local Slave or Connection I/O Value Diagnostics

Introduction

Use the **I/O Values** page to display both the input data image and output data image for the selected local slave or connection.

Use the **Refresh Every 500ms** checkbox to display static or dynamic data, as follows:

When the checkbox is...	This page...
Selected	• Displays data that is dynamically updated every 500 ms, and • Increments the number at the top of the table each time data is refreshed
De-selected	• Displays static data, and • Does not Increment the number at the top of the table, which instead displays a constant value

NOTE: Before you can open the **Diagnostic** window, you first must connect the communication module or remote device DTM to the physical module or device. To do this, select the appropriate node in the **DTM Browser**, then select **Edit → Connect**.

To open this page:

Step	Action
1	In the **DTM Browser**, select the communication module and click the right mouse button. A pop-up menu opens.
2	In the menu, select **Device menu → Diagnostic**.
3	In the left pane of the **Diagnostic** window, click on one of the following: • the communication module node, or • a connection node
4	Click on the **I/O Values** tab to open that page.

Local Slave / Connection I/O Values

This page displays the following parameters for either a local slave or a remote device connection input and output values:

Parameter	Description
Input/Output data display	A display of the local slave or remote device input or output data image.
Length	The number of bytes in the input or output data image.
Status	The Scanner Diagnostic object’s status, with respect to the read of the input or output data image.

Logging DTM Events to a Control Expert Logging Screen

Description

Control Expert maintains a log of events for:
- the Control Expert embedded FDT container
- each Ethernet communication module DTM
- each EtherNet/IP remote device DTM

Events relating to the Control Expert FDT container are displayed in the **FDT log event** page of the **Output Window**.

Events relating to a communication module or remote EtherNet/IP device are displayed:
- in configuration mode: in the **Device Editor**, by selecting the **Logging** node in the left pane
- in diagnostic mode: in the **Diagnostics** window, by selecting the **Logging** node in the left pane

Logging Attributes

The **Logging** window displays the result of an operation or function performed by Control Expert. Each log entry includes the following attributes:

Attribute	Description	
Date/Time	The time the event occurred, displayed in the format: yyyy-mm--dd hh:mm:ss	
Log Level	The level of event importance. Values include:	
	Information	A successfully completed operation.
	Warning	An operation that Control Expert completed, but which may lead to a subsequent error.
	Error	An operation that Control Expert was unable to complete.
Message	A brief description of the core meaning of the event.	
Detail Message	A more detailed description of the event, which may include parameter names, location paths, etc.	

Accessing the Logging Screen

In Control Expert:

Step	Action
1	Open a project that includes a BME NOC 03•1 Ethernet communication module.
2	Clock **Tools** → **DTM Browser** to open the **DTM Browser**.
3	In the **DTM Browser**, double-click the BME NOC 03•1 (or right-click **Open**) to open the configuration window.
4	Select **Logging** in the navigation tree in the left pane of the window.

Section 10.6
Hot Standby Services

What Is in This Section?

This section contains the following topics:

Hot Standby Synchronization

Introduction

As an example, you have CPU A and CPU B in a Quantum EIO Hot Standby system. CPU A is the primary CPU, and CPU B is the standby CPU. After a switchover, CPU B becomes the primary. The 140NOC78•00 modules in CPU A synchronize with the 140NOC78•00 modules in CPU B to update CPU B with the data from CPU A.

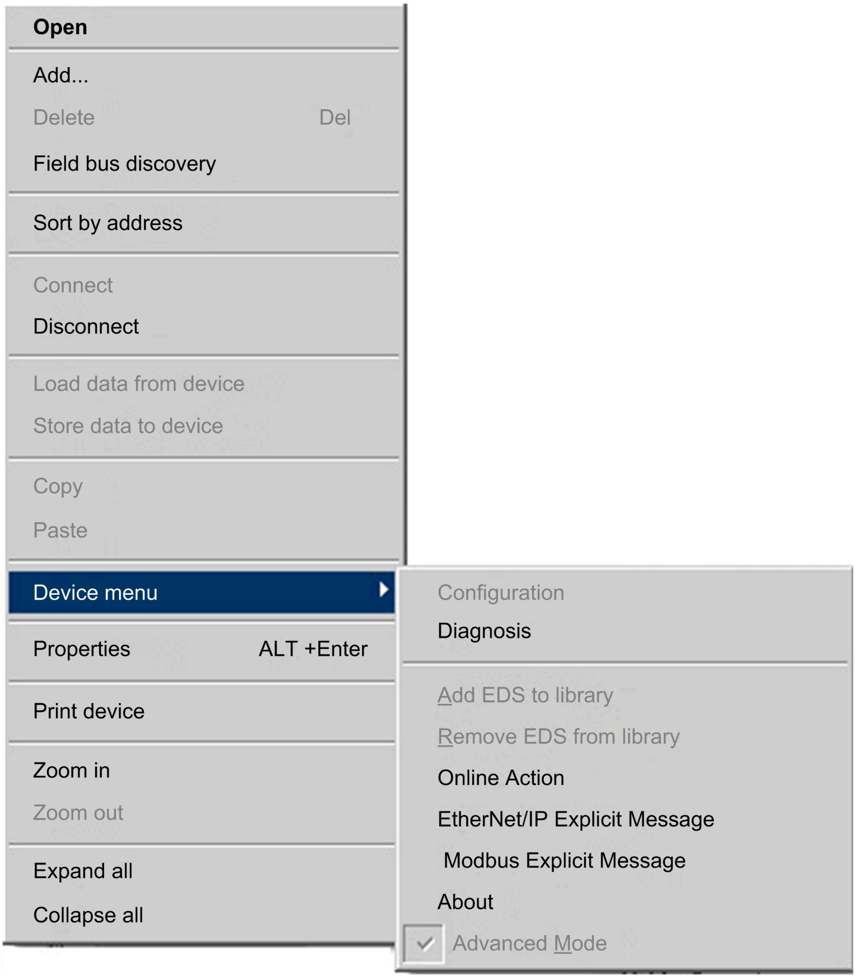
The 140NOC78•00 standby modules then synchronize with the primary modules every 10 seconds to verify that the data in the standby modules has been updated in the primary modules. If the standby modules unsuccessfully synchronize with the primary modules, they keep polling for the primary modules every 10 seconds.

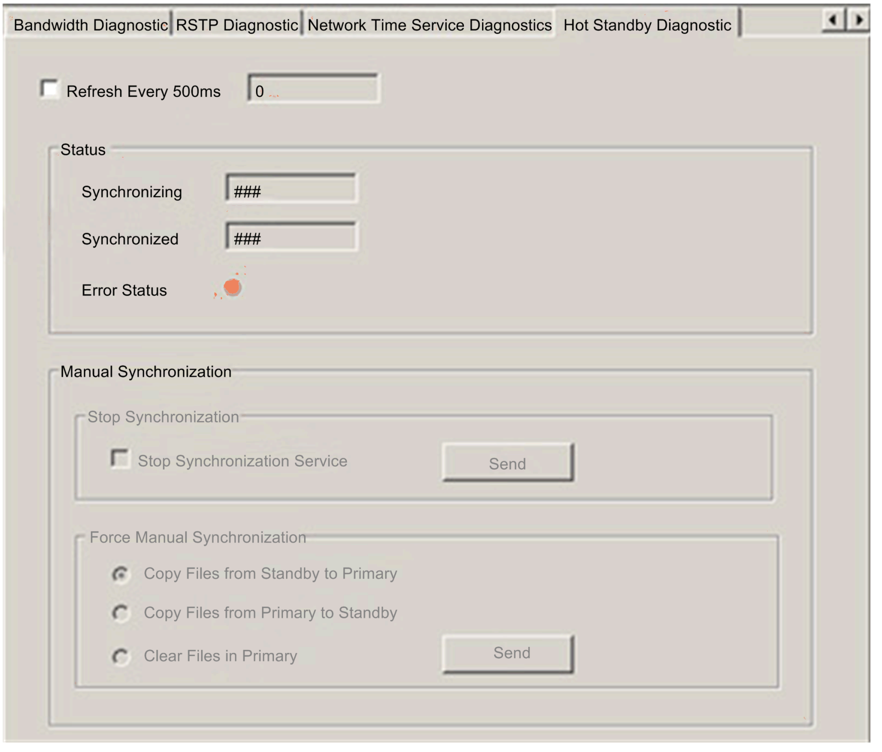
If the data in the standby and primary modules is different, the synchronization stops and a synchronization error is detected in the standby CPU. The purpose of this process is to check if data has been added to the previous primary module before the polling period expired when the Hot Standby system switchover occurred.

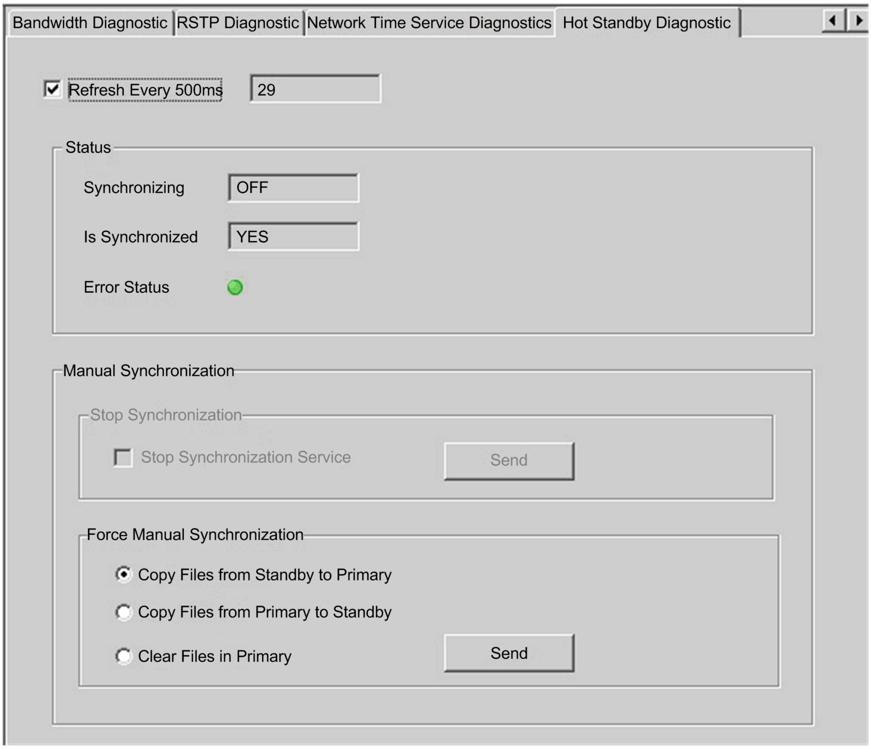
NOTE: When the 140NOC78•00 standby modules are offline, they do not synchronize.

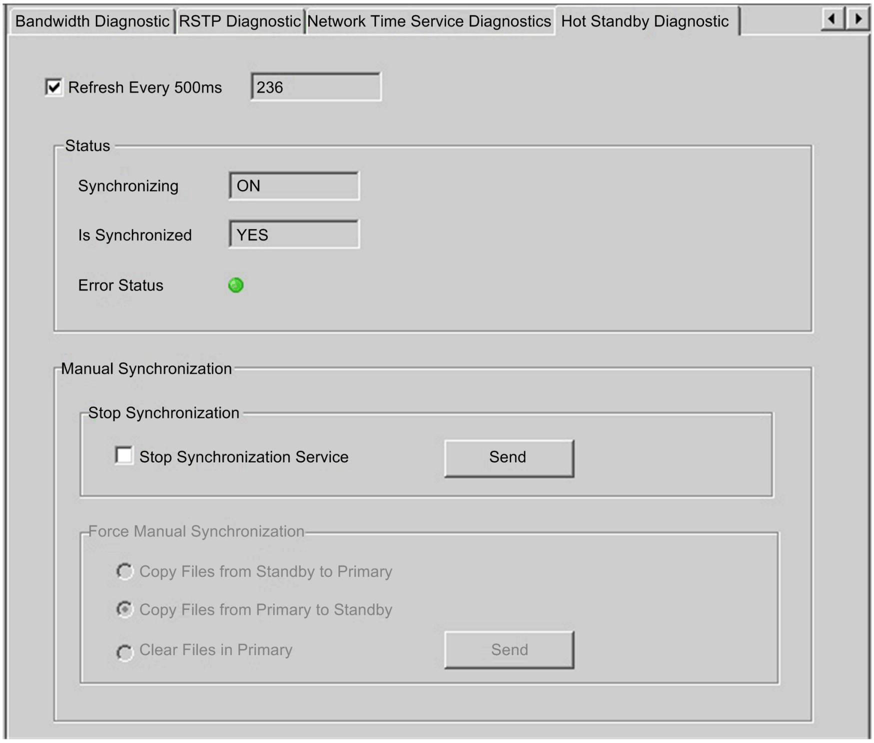
Recovering from a Synchronization Detected Error

If the synchronization between 140NOC78•00 modules does not work properly before the switchover occurs, follow these steps:

Step	Action
1	In the **DTM Browser** window, right-click the **140 NOC 78• 00 module** → **Connect**.
2	Right-click the **140 NOC 78• 00 module** → **Device menu** → **Diagnosis** as shown in the following figure: 

Step	Action
3	Click the **Hot Standby Diagnostic** tab. **Result**: The following screen displays: 

Step	Action
4	• Select the **Refresh Every 500ms** check box to view the synchronization status. • Click the **Copy Files from Standby to Primary** bullet in the **Force Manual Synchronization** field. • Click Send. **Result**: The synchronization status is off, and the modules are synchronized as the following screen shows: 

Step	Action
5	If you select **Manual Synchronization**, the **Force Manual Synchronization** field options are disabled. **Result**: The synchronization status is on, and the modules are synchronized, as the following screen shows: 

Hot Standby Switchover

140NOC78•00 IP Address Swap Time

The following table details the 140NOC78•00 module IP address swap time in a Quantum EIO Hot Standby system:

Maximum swap time	500 ms (IP address swapping) + connection establishment time (3 s)
Recommended setting for implicit message	Set RPI to 1/2 of MAST cycle time (50 ms maximum)

Timeout multiplier setting:

MAST Cycle Time (ms)	Recommended RPI (ms)	Timeout Multiplier	Connection Timeout (ms)
20	10	16	160
50	25	8	200
100	50	4	200
200	50	4	200
255	50	4	200

NOTE: The maximum swap time may increase if the end device does not respond in a timely manner.

NOTE: During the swap, there may be disruption in communication between the 140NOC78•00 module and the end device. Confirm that the application can tolerate this communication disruption.

Chapter 11
Firmware Upgrade

140 NOC 78• 00 Firmware Upgrade

OS

Use the Control Expert OS to upgrade the firmware on the 140 NOC 78• 00 head module.
OS Loader was installed on your PC when you installed Control Expert. (The minimum required version of OS Loader is V7.0. The compatible version is included with your copy of Control Expert.)

A complete firmware upgrade includes the installation of these discrete files:
- kernel
- exec

The kernel and exec files are installed independently. Therefore, perform the firmware upgrade process two times (once for each file).

The name of the firmware file indicates the upgrade type (kernel or exec). Examples:
- kernel file name: `CCS1_Noc_Kerl_OSLoader.bin`
- exec file name: `CCS1_Noc_Exec_OSLoader.bin`

NOTE: These instructions assume that you are familiar with Control Expert. For more information about the OS Loader, refer to *EcoStruxure™ Control Expert, OS Loader, User Manual*.

Upgrade Procedure

Follow these steps to upgrade either the firmware kernel or the firmware exec.

NOTE:

- Before performing the firmware upgrade procedure, check whether the module's FTP/TFTP services are enabled, and if they are not, enable them *(see page 154)*.
- We recommend that you update the kernel before you upgrade the exec. Both firmware upgrade files are installed in the same manner. The only difference is the name of the file you select.
- Interruption to power or communications during the firmware upgrade process can disrupt the upgrade. If that happens, restart the module.

Step	Action	Comment
1	Connect the PC that is running the Control Expert OS Loader directly to one of the module ports.	Available ports: ● SERVICE port ● INTERLINK port ● DRS port that is configured for a distributed I/O cloud
2	Launch OS Loader.	**Start** → **Programs** → **EcoStruxure Control Expert** → **OS Loader**.
3	Click **Next** to continue.	Go directly to the first installation step.
4	Select the **FTP** communication driver and press **Next** to continue.	The next screen displays a list of devices discovered by OS Loader. It also displays the FTP address for each discovered device.
5	In the **Target Address** area, type in the **FTP Address** of the 140 CRP 312 00 module that is the target of the upgrade.	—
6	Click **Next** to continue. Perform these tasks at the next installation screen: a Select **Download OS to device**. b Click the **Browse** button to navigate to and select the desired firmware upgrade file.	—
7	Click **Next**. Perform these tasks at the next installation screen: a Compare the selected firmware **File** against the firmware already loaded in the **Device**. b Confirm that the **Hardware ID** for both the file and the device are the same	—
8	Click **Next**. On the summary page, click **Download**.	The OS Loader displays the progress of the FTP session. The download is comple when it displays the word **SUCCESS**.
9	Click **Close**.	The firmware download is finished.

The upgrade process takes approximately 2 minutes:

- 1 minute for firmware upgrade
- 1 minute to reboot and reestablish I/O connections

NOTE: During the firmware upgrade, the I/O communications with the 140 NOC 780 00 head module are interrupted. After the hold up time expires, the I/O modules return to their fallback state.

Hot Standby

Use these steps to upgrade the 140 NOC 78• 00 firmware in Hot Standby configurations:

Step	Action
1	Use the preceding instructions to upgrade the firmware for the 140 NOC 78• 00 in the standby rack. **NOTE:** During the firmware upgrade, the I/O communications with the 140 NOC 78• 00 module in the primary rack are not interrupted.
2	When the firmware on the 140 NOC 78• 00 in the standby rack is upgraded, perform a manual switch-over that gives the newly upgraded standby rack the role of primary rack.

Chapter 12
Embedded Web Pages

Overview

This chapter describes the embedded web pages for the 140NOC78•00 head module.

The communication module includes a Hypertext Transfer Protocol (HTTP) server. The server transmits web pages for the purpose of monitoring, diagnosing, and controlling remote access to the communication module. The server provides easy access to the communication module from standard internet browsers, including, but not limited to, Internet Explorer.

Before attempting to view the module's embedded web pages, check whether the module's HTTP service is enabled, and if not, enable it *(see page 154)*.

What Is in This Chapter?

This chapter contains the following sections:

Section 12.1
Accessing the Embedded Web Server

Introduction

This section introduces the 140NOC78•00 head module's embedded web server, and describes how to access (and to control access to) the web pages.

What Is in This Section?

This section contains the following topics:

Introducing the Embedded Web Pages

Introduction

Use the 140NOC78•00 head module's embedded web server pages to:

- display real-time diagnostic data for both the module and other networked devices
- read the values of and write values to Control Expert application variables
- manage and control access to the embedded web pages by assigning separate passwords for:
  - viewing the diagnostic web pages
  - using the data editor to write values to Control Expert application variables

Requirements

The embedded web server presents module data in the form or standard HTML web pages. Access the embedded web pages using Internet Explorer version 4.0 or later, running the Java Runtime Environment (JRE) version 1.6 or later.

Accessing the Home Page

On First Use

Before you begin to use the 140NOC78000 distributed I/O head module's embedded web pages, you need to:

- navigate to the web server *(see Quantum using EcoStruxure™ Control Expert, 140 NOC 771 01 Ethernet Communication Module, User Manual)*
- access web page content by inputting the default username and password *(see Quantum using EcoStruxure™ Control Expert, 140 NOC 771 01 Ethernet Communication Module, User Manual)* combination
- change passwords *(see Quantum using EcoStruxure™ Control Expert, 140 NOC 771 01 Ethernet Communication Module, User Manual)* that are required for:
  - accessing web pages
  - writing data values using the data editor

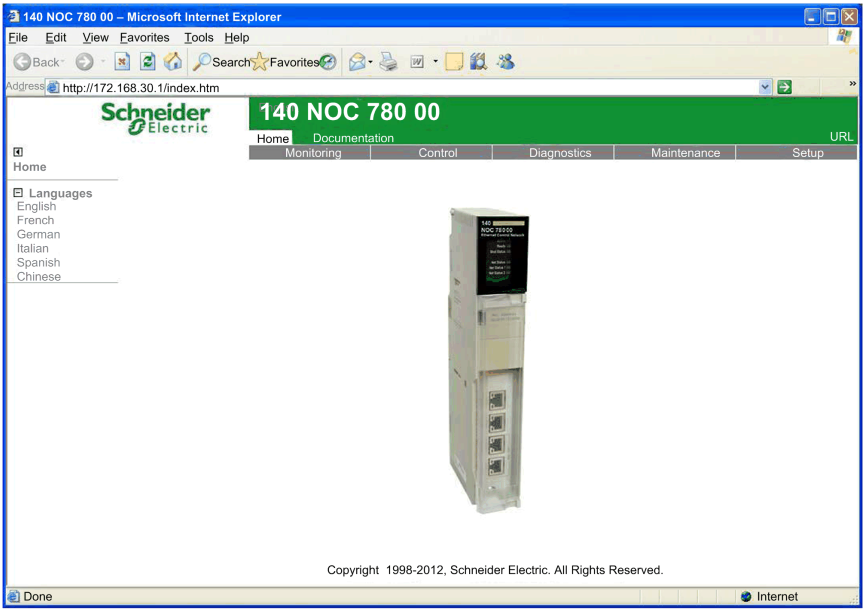
Navigating to the Web Server

To access the embedded web server:

Step	Action
1	Open an Internet browser.
2	Enter the IP address of the 140NOC78000 module in the format: *http://IP address*.
3	Click **Enter**.

NOTE: If a DNS name has been assigned to the module, the DNS name can be used instead of the IP address.

Result: The web server opens, displaying the **Home** page:



Use the **Home** page as the point of entry to the 140NOC78000 module's embedded web server. From here, you can navigate to every other web page.

Using and Editing a Username and Passwords

Inputting the Username and Web Page Access Password

A username and password are required to access web page content and edit application data. Username and password settings are case sensitive.

The embedded web pages support the use of a single, editable username for both web page access and data editing. The factory-default username setting is **USER**.

The embedded web pages require 2 different passwords, as follows:

- an HTTP access password, which grants read-only access to web page content
- a data editor write password, which permits the editing of data values using the **Data Editor**

Each password can be edited. The factory default setting for each password is **USER**.

To input a username and password combination

Step	Description
1	After navigating to the embedded web server *(see Quantum using EcoStruxure™ Control Expert, 140 NOC 771 01 Ethernet Communication Module, User Manual)*, select one of the main menu selections (for example, **Setup**).
2	Select a page name from the list of pages on the left side of the page (for example, **Security**). **Result**: The following dialog opens. 

Step	Description
3	Type in the required **Username** and **Password** combination, then click **OK**. **NOTE:** In the above example, the settings for both the Username and Password remain set to the default setting of USER.

Editing the Username and Passwords

The single username and both passwords can be edited in the **Security** web page. To edit username and passwords, follow these steps:

Step	Description
1	Navigate to and open the web server, *(see Quantum using EcoStruxure™ Control Expert, 140 NOC 771 01 Ethernet Communication Module, User Manual)* using the IP address of the communication module. **Result**: The **Home** page opens.
2	From the **Home** page, click the **Setup** main menu item. If required, input the username and web page password *(see Quantum using EcoStruxure™ Control Expert, 140 NOC 771 01 Ethernet Communication Module, User Manual)*. **Result**: The **Setup** page opens: 

<table>
<tr><th>Step</th><th colspan="2">Description</th></tr>
<tr><td>3</td><td colspan="2">On the left side of the page, click on the Security node. (If required, input the Username and web page access Password.)
The Security page opens:

</td></tr>
<tr><td rowspan="4">4</td><td colspan="2">To change the username and password combination used for web page access, in the HTTP access rights section of the page, enter values for the following fields:</td></tr>
<tr><td>Username:</td><td>• To change the username: type in a new username
• To retain the current username (for example, if you are changing only the password): type in the current username</td></tr>
<tr><td>New password:</td><td>• To change the password: type in a new password
• To keep the current password (for example, if you are changing only the username): type in the current password</td></tr>
<tr><td>Confirm password:</td><td>Type in the same password entered in the New password field, above.</td></tr>
<tr><td>5</td><td colspan="2">Click the Save User button.</td></tr>
<tr><td rowspan="4">6</td><td colspan="2">To change the password used for writing data values in the Data Editor, in the Data Editor Write Password section of the page, enter values for the following fields:</td></tr>
<tr><td>Data Editor write password:</td><td>Type in the current password that is required to write data using the Data Editor.</td></tr>
<tr><td>New write password:</td><td>Type in the new Data Editor password.</td></tr>
<tr><td>Confirm write password:</td><td>Type in the same password entered in the New write password field, above.</td></tr>
<tr><td>7</td><td colspan="2">Click the Change Write Password button.</td></tr>
</table>

Section 12.2
Monitoring the Control Expert Application

Overview

This section describes how to use the 140NOC78•00 head module’s embedded web pages to monitor the Control Expert application.

What Is in This Section?

This section contains the following topics:

Using the Monitoring Page

Monitoring Page

Click the main menu **Monitoring** command to display the **Monitoring** page:

To access a monitoring service, click either of the following links:

- **Data Editor Lite**
- **Data Editor Standard**

Data Editor (Standard)

Overview

The **Data Editor** is a Java applet that dynamically displays run-time application data. Use the **Data Editor** to create and edit data monitoring tables that provide read/write access to application data and device registers.

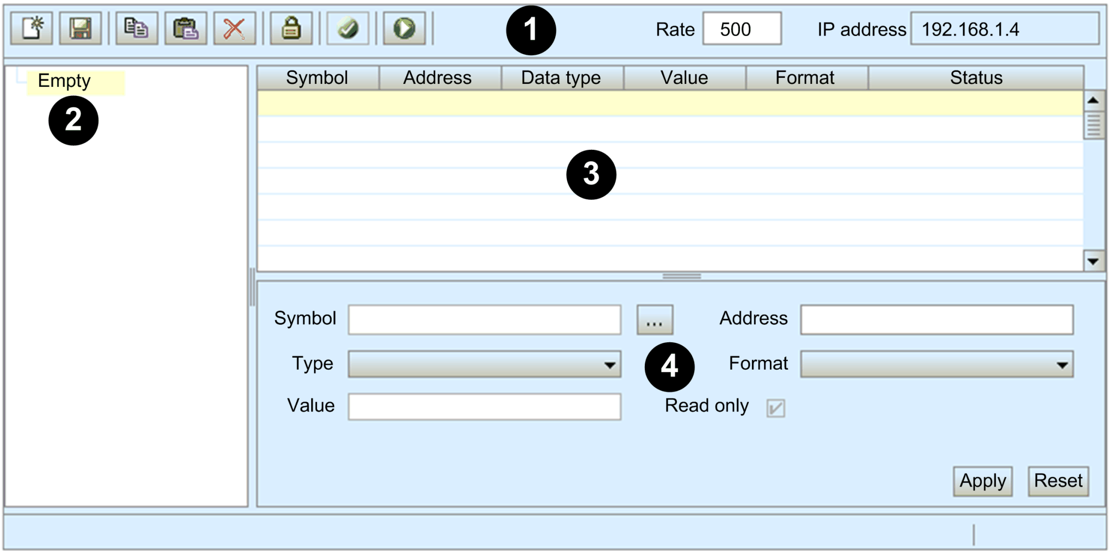
NOTE: Write access is password protected.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION The data editor makes it possible to write to application variables and change application data values. ● Use passwords to strictly limit access to write data functionality. ● Do not use weak passwords, including the default password and other obvious passwords. ● Limit access to trained personnel. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

This topic describes the **Data Editor** user interface.

Data Editor

The **Data Editor** presents the following controls:



1 toolbar
2 data template list
3 data template
4 configuration area

Toolbar

The **Data Editor** toolbar presents the following features:

Command or Field	Icon	Description
New		● If a node in the data template list is selected, this command opens the **New table** dialog for the creation of a new data template. The new data template is inserted below the selected node. ● If a row in the currently open data template is selected, this command inserts a new row below the selected row.
Save		Saves changes made to both the data template list and each data template.
Copy		● If a node in the data template list is selected, this command copies the selected data template. ● If an item (or row) in the currently open data template is selected, this command to copies the selected item.
Paste		● If the root, or Empty, node is selected in the data template list, this command pastes a previously copied data template into the list. ● If an empty item (or row) in the currently open data template is selected, this command pastes a previously copied item into the data template item at the selected row. **NOTE:** When adding a copied item, or row, to a data template, the paste command will overwrite item data in the selected row. To insert a copied row between existing rows, first use the **New** command to create an empty row, then paste the copied data into the new row.
Delete		Deletes the selected data template from the list, or the selected item from the data template.
Change password		Opens the **Change password** dialog, where you can change the Data Editor Write *(see Quantum using EcoStruxure ™ Control Expert, 140 NOC 771 01 Ethernet Communication Module, User Manual)* password. **NOTE:** The Data Editor Write password can also be changed in the **Setup → Security** web page.
Read PLC symbols		Loads the existing Control Expert symbol, or variable, names into the **Lookup Variable** dialog. Variables that have been loaded into this dialog can be added to the currently open data template.
Start animation		Starts the dynamic display of value and status for the items contained in the selected data template. **NOTE:** The Start animation icon is visible only when animation is turned OFF.
Stop animation		Stops the dynamic display of value and status for the items contained in the selected data template. **NOTE:** The Stop animation icon is visible only when animation is turned ON.

Command or Field	Icon	Description
Rate	—	The refresh rate of the dynamic display of data template items, in milliseconds.
IP address	—	The IP address of the Ethernet communication module and its embedded web server.

Data Template List

The data template list displays a node for each data template that was either:

- previously saved
  — or —
- created after the **Data Editor** was opened, but not yet saved

Select a data template in this list to view or edit its contents.

NOTE: If you create a new data template then navigate away from the **Data Editor** before clicking the **Save** button, the new data template will be lost.

Data Template

Use the data template when animation is turned ON to monitor the status and values of items for the template that is currently selected in the data template list.

Each data template item (or row) is defined in the configuration area. A data template item can contain the following fields:

<table>
<tr><th>Field</th><th colspan="2">Description</th></tr>
<tr><td>Symbol</td><td colspan="2">Contains the names of Control Expert symbols (variables).</td></tr>
<tr><td rowspan="6">Address</td><td colspan="2">Contains direct addresses and the addresses of Control Expert symbols (variables). Any direct address can be viewed by entering its reference in this field. Valid direct addresses include:</td></tr>
<tr><td>%Mi</td><td>same as for 0X coils</td></tr>
<tr><td>%Ii</td><td>same as 1x for discreet inputs</td></tr>
<tr><td>%IWi</td><td>same as 3x for input registers</td></tr>
<tr><td>%MWi, %MDi, %MFi</td><td>same as 4x for holding registers</td></tr>
<tr><td colspan="2">NOTE:
● A single bit of any word address (for example, %MWi, %IWi) can be specified by appending ".j" to the address, where "j" is a bit index in the range of 0 (LSB) to 15 (MSB). For example, bit 4 of the value at %MW101 would be specified as %MW101.4.
● A direct address can include an index specification that allows it to be treated as an array variable. Indexed addressing can be used with a %Mi, %MWi, %MDi, or %MFi address by appending "[j]" to the address of the beginning of the array, where "j" is an unsigned integer value. For example, the third value of an array of float values starting at %MF201 would be specified as %MF201[2].</td></tr>
</table>

Field	Description	
Data type	Contains the data type of the symbol (variable) or direct address. Symbol (variable) data types appear automatically when the symbol (variable) is located. Select direct address data types from a drop-down list. The following data types are valid:	
	INT	16-bit signed integer
	UINT	16-bit unsigned integer
	DINT	32-bit signed integer
	UDINT	32-bit unsigned integer
	REAL	32-bit IEEE floating point
	TIME	32-bit unsigned integer (in ms)
	DATE	Date (32-bit BCD)
	TOD	Time of day (32-bit BCD)
	BOOL	1 bit discrete (Boolean)
Value	When animation has started, this field displays the value of the symbol (variable) or direct address. This field is updated continuously.	
Format	Contains the format type for displaying the value of the symbol (variable) or direct address. The following formats are available:	
	bool	Boolean
	dec	Decimal
	hex	Hexadecimal
	binary	Binary
	ASCII	bytes displayed as ASCII characters
	time	day_hr_min_sec_ms
	date	YYYY-MM-DD or HH:MM:SS
Status	Contains messages describing the status of communication with the direct address:	
	if communication is normal	The status message reads **OK**
	if communication is interrupted	The status field displays a system message describing the interruption

Configuration Area

Open and close the configuration area by double-clicking on a row in the data template. The configuration area will display the configuration settings for the selected row. Use the up and down arrows on your keyboard to move between rows in the data template and display their settings in the configuration area.

Use the configuration area when data template animation is turned OFF to:

- create a new data template *(see page 341)*
- display the items contained in an existing data template *(see page 343)*
- add a direct address *(see page 343)* to a data template

Use the configuration area when data template animation is turned ON to write data to read/write application variables.

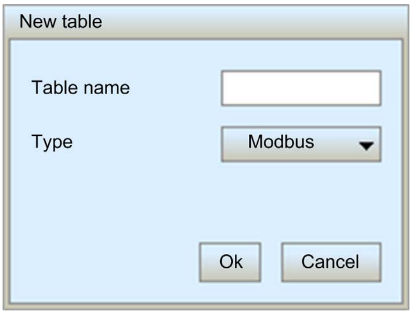
Refer to the topic Working With Data Templates *(see page 341)* for more information on how to use the controls in the configuration area.

Working With Data Templates

Creating a Data Template

To display and access application data, first create a data template.

To create a new data template, follow these steps:

Step	Description
1	Confirm that **Data Editor** animation is OFF. If necessary, click the **Stop animation** toolbar button.
2	Click the **New** table toolbar button. **Result**: The **New table** dialog opens: 
3	In the **Table name** field, type in the name of the new data template.
4	Click **Ok**. **Result**: The new data template appears as a node in the data template list.

NOTE: Save the new data template before performing any other task in the **Data Editor**. Moving to another page or creating a new data template in the current page before saving your work deletes the new data template.

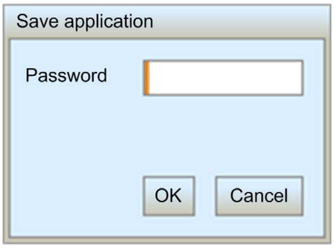
Saving a Data Template

After you save a new data template, you can re-use it to view or modify its contents.

NOTE:

- The last saved modification overwrites the pre-existing data template, even if the data template was originally created by someone other than yourself.
- If a data template is open for viewing by someone else, your edits to that data template will be seen only when that person next accesses the **Data Editor**.

To save a new data template, follow these steps:

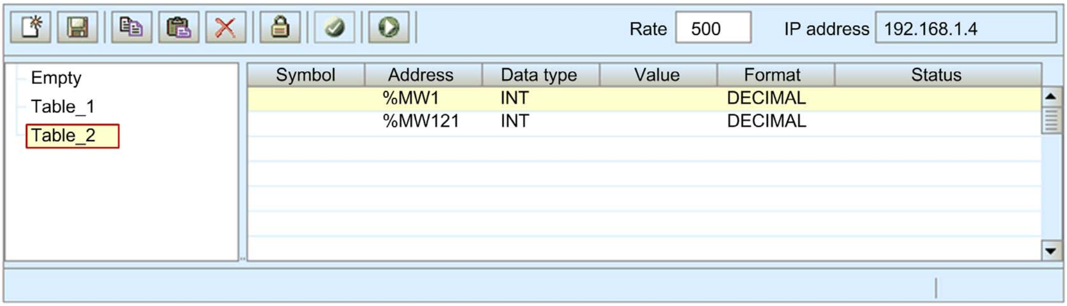
Step	Description
1	Click the **Save**  toolbar button. **Result**: The **Save application** dialog opens: 
2	In the **Password** field, type in the **Web Page (HTTP) Access** password. **NOTE:** The default password is **USER**.
3	Click **Ok**. The new data template is saved.

Displaying an Existing Data Template

When you open a saved data template, you can use it to:

- edit its contents by inserting either a direct address
- monitor the value and status of data items
- write data values to a read/write variables

The data template list, located on the left side of the **Data Editor**, displays the saved data templates. Select a data template node from the list to display that template's data items in the spreadsheet on the right:



Inserting a Direct Address Into a Data Template

You can add Control Expert direct address items (also called located registers) into a data template. After a direct address item is added, you can view or modify its value.

To add a direct address item to a data template, follow these steps:

Step	Description
1	In the data template spreadsheet, double-click on an empty row. **Result**: The **Data Editor** configuration area opens.
2	In the **Address** field of the configuration area, type the item's direct address.
3	In the configuration area, click **Apply**. **Result**: The selected row is updated.
4	**Save** your edits.

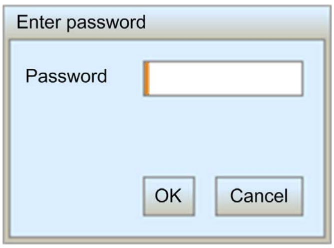
Modifying Data Values Using a Data Template

You can use the **Data Editor** to write data values to a direct address item, and send the new value to the controller.

For example, suppose that you have programmed a pushbutton object to jog a motor when the button is depressed and to stop jogging when the button is released. If communications are lost while the button is depressed, the motor will continue to jog even when the button is released. Graphic objects are not designed to be used to control situations like this, unless other interlock methods are installed in the system.

NOTE: You can only modify the value of data items that are defined as read/write in the Control Expert application.

To use the **Data Editor** to edit data, follow these steps:

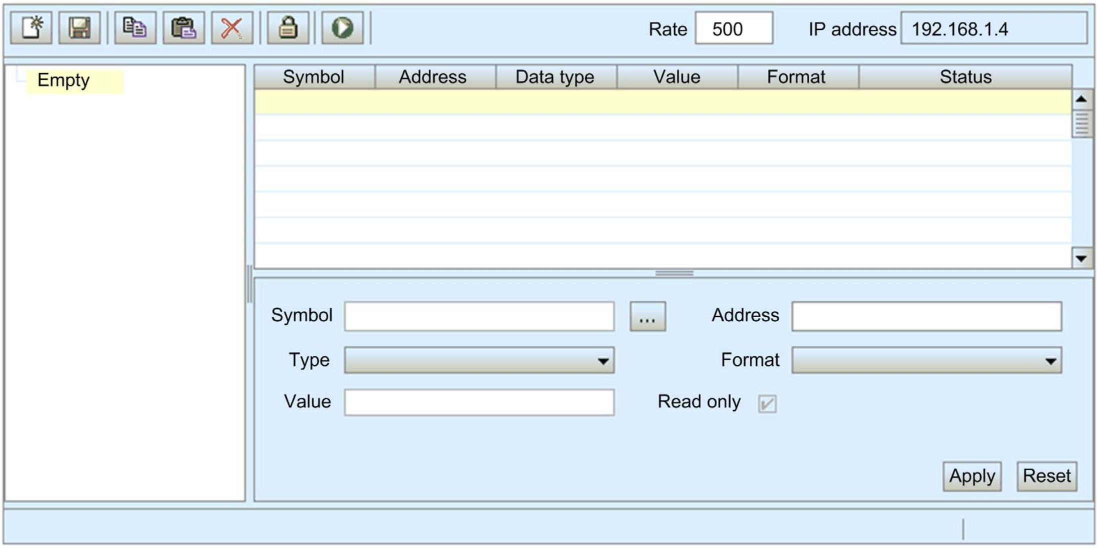
Step	Description
1	In the data template spreadsheet, double-click on the item you want to write data to. **Result**: The **Data Editor** configuration area opens, displaying the fields for the selected item.
2	In the **Value** field, type the desired data value.
3	Click **Apply**. **Result**: The **Enter password** dialog opens: 
4	In the **Password** field, type the Write Data password. **NOTE:** The default password is **USER**.
5	Click **OK**. **Result**: The new value is sent to the controller.

Data Editor (Lite)

Overview

Data Editor Lite is a version of the **Data Editor** that is smaller in size and therefore faster to download, especially for use via a dial-up connection.

Data Editor Lite presents the same interface as the **Data Editor**, with the exception that its toolbar does not include the **Read PLC Symbols** function:



Variables

Data Editor Lite accepts the following IEC variables:

Address	Type	Display
%MW IEC internal word	INT	DECIMAL
%MD IEC double word	DINT	DECIMAL
%M IEC internal bits	BOOL	BOOLEAN

NOTE: You cannot access the **Lookup Variable** dialog and insert symbols into a data template using **Data Editor Lite**. You can insert only direct addresses.

Re-Using Data Editor Templates

Data Editor Lite can reuse the same templates created with the **Data Editor**. However, **Data Editor** templates can use a wider range of variable types than **Data Editor Lite**. When **Data Editor Lite** encounters a variable it cannot manage, it displays `Not Supported` as the data type. In this case, the variable cannot be edited using **Data Editor Lite**.

Section 12.3
Diagnostics

Overview

This section describes the diagnostic services provided by the 140NOC78•00 head module.

What Is in This Section?

This section contains the following topics:

Using the Diagnostics Page

Diagnostics Page

Click the main menu **Diagnostics** command to display the **Diagnostics** page:

To access a monitoring service, click one of the following links:

- Status Summary *(see page 349)*
- Rack Viewer *(see page 352)*
- Ethernet:
  - Processor Load *(see page 353)*
  - Scanner Status *(see page 355)*
  - Messaging *(see page 357)*
  - QoS Configuration *(see page 361)*
  - Ethernet Statistics *(see page 359)*
  - Redundancy *(see page 363)*
  - Network Time Service *(see page 366)*
  - Email *(see page 364)*
- Properties *(see page 369)*

Status Summary

Introduction

Use the **Status Summary** page to view the status of:
- the LEDs *(see Quantum using EcoStruxure™ Control Expert, 140 NOC 771 01 Ethernet Communication Module, User Manual)* located on the front of the 140NOC78•00 head module
- the Ethernet services *(see Quantum using EcoStruxure™ Control Expert, 140 NOC 771 01 Ethernet Communication Module, User Manual)* supported by the 140NOC78•00 module
- the 140NOC78•00 module in its role as:
  - scanner
  - Modbus TCP server
  - EtherNet/IP messaging server

Status Summary Display

The **Status Summary** page looks like this:

LEDs

Label	Status
Active	Ready for Operation
Ready	Waiting for Address server resp
Mod Status	Ready for Operation
Net Status	Not Ready for Operation
Net Status 1	Not Ready for Operation
Net Status 2	Fault detected

Services

Function	Status
DHCP Server	Enabled
FDR Server	Enabled
Access Control	Disabled
Embedded Router	Disabled
Scanner Status	Working properly
Network Time Service	Working properly
E-mail Service	At lease one connection is bad

To open this page:

Step	Action
1	Starting at the **Home page**, click the **Diagnostics** main menu item. **Result**: The **Diagnostics** page opens.
2	On the left side of the **Diagnostics** page, select **Ethernet** → **Status Summary**.
3	If necessary, type in the HTTP web access password. **NOTE:** The default password is **USER**.

Status Summary Data

The **LEDs** section of the page can present the following operational states:

LED	Color	Text Descriptions
Active	Green	Ready for operation
	Red	Detected error present
Ready	Green	Waiting for address server response
	Red	Duplicate IP address
		Waiting for served IP configuration
		Default IP address in use
		Detected configuration error
Module Status	Green	Ready for operation
	Red	Not configured
		Fault detected
		Recoverable fault detected
Network Status	Green	Ready for operation
	Red	Detected connection error
		Duplicate IP address
Network Status 1	Green	Ready for operation
	Gray	Detected connection error
		Duplicate IP address
Network Status 2	Green	Ready for operation
	Gray	Detected connection error
		Duplicate IP address

The **Services** section of the page can present the following functional conditions:

Function	Color	Text Descriptions
DHCP Server	—	Enabled
FDR Server		Enabled
Access Control		Disabled
IP Forwarding		Disabled
Scanner Status	Green	Working properly
	Red	At least one connection is bad
	Gray	Not configured
Network Time Service	—	Working properly
E-mail Service	—	At least one connection is bad

Rack Viewer

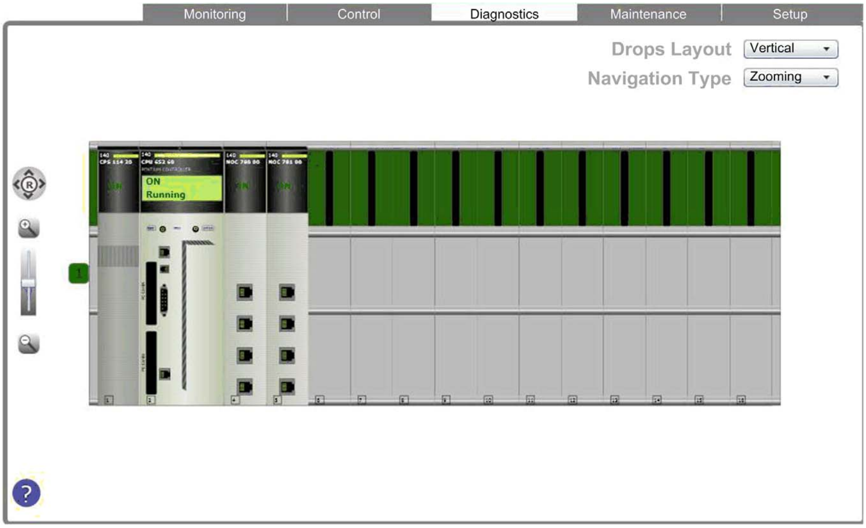
Introduction

Use the **Rack Viewer** to access web pages that describe the identity, placement, configuration, and operation of modules in the Quantum rack.

To view information describing a specific module, including the 140NOC78•00 head module, click the image of that module in the **Rack Viewer**.

Rack Display

The **Rack Viewer** looks like this, when it is first opened:



To open this page:

Step	Action
1	Starting at the **Home page**, click the **Diagnostics** main menu item. **Result**: The **Diagnostics** page opens.
2	On the left side of the **Diagnostics** page, select **Rack Viewer**.
3	If necessary, type in the HTTP web access password. **NOTE:** The default password is **USER**.
4	To open a page displaying configuration and operating data for the 140NOC78•00 module, click the module image in the rack. **Result**: The Rack Viewer parameter page opens.
5	To return to the main Rack Viewer page, click the **Back** arrow.

Processor Load

Introduction

Use the **Processor Load** web page to display dynamically generated data for the 140NOC78•00 head module's bandwidth usage.

Processor Load Display

The **Processor Load** page looks like this:

Monitoring | Control | Diagnostics | Maintenance | Setup

Processor Load

Communication Load

Function		Statistics		Units
I/O	Scanner	**Ethernet/IP Sent (writes)**	**0**	Packets per second
		Ethernet/IP Received (read)	**0**	Packets per second
		Modbus TCP Requests	**0**	Packets per second
		Modbus TCP Responses	**0**	Packets per second
	Adapter	**Ethernet/IP Sent (writes)**	**0**	Packets per second
		Ethernet/IP Received (read)	**0**	Packets per second
		Module Capacity	**5500**	Packets per second
		Module Utilization	**0.0**	Percent
	Client	**Ethernet/IP activity**	**0**	Messages per second
		Modbus TCP activity	**0**	Messages per second
	Server	**Ethernet/IP activity**	**0**	Messages per second
		Modbus TCP activity	**2**	Messages per second

NOTE: The background color for the **Processor Utilization** and **Module Utilization** values varies, depending upon the percentage of utilization. If utilization is:

- 90% to 100%: background color is RED
- 80% to 89.99%: background color is YELLOW
- 0% to 79.99%: background color is GRAY

To open this page:

Step	Action
1	Starting at the **Home page**, click the **Diagnostics** main menu item. **Result**: The **Diagnostics** page opens.
2	On the left side of the **Diagnostics** page, select **Ethernet** → **Processor Load**.
3	If necessary, type in the HTTP web access password. **NOTE:** The default password is **USER**.

Processor Load Parameters

The **Processor Load** page displays the following parameters for the communication module:

Parameter	Description
Module Load:	
Processor Utilization	The percent of Ethernet communication module processor capacity used by the present level of communication activity. The background color of the value changes, depending on the percentage utilization.
I/O Scanner:	
EtherNet/IP Sent (writes)	The number of EtherNet/IP packets the module has sent, since the last reset, in packets/second.
EtherNet/IP Received (read)	The number of EtherNet/IP packets the module has received, since the last reset, in packets/second.
Modbus TCP Requests	The number of Modbus TCP requests the module has sent, since the last reset, in packets/second.
Modbus TCP Responses	The number of Modbus TCP responses the module has received, since the last reset, in packets/second.
I/O Adapter:	
EtherNet/IP Sent (writes)	The number of EtherNet/IP packets the module has sent in the role of a local slave since the last reset, in packets/second.
EtherNet/IP Received (read)	The number of EtherNet/IP packets the module has received in the role of a local slave since the last reset, in packets/second.
I/O - Module	
Module Capacity	The maximum number of packets that the module can process, in packets per second.
Module Utilization	The percentage of communication module capacity being used by the application. The background color of the value changes, depending on the percentage utilization.
Messaging - Client:	
EtherNet/IP activity	The number of I/O messages sent by the module using the EtherNet/IP protocol since last reset, in packets per second.
Modbus TCP activity	The number of I/O messages sent by the module using the Modbus TCP protocol since last reset, in packets per second.
Messaging - Server:	
EtherNet/IP activity	The number of I/O messages received by the module using the EtherNet/IP protocol since last reset, in packets per second.
Modbus TCP activity	The number of I/O messages received by the module using the Modbus TCP protocol since last reset, in packets per second.

Scanner Status

Introduction

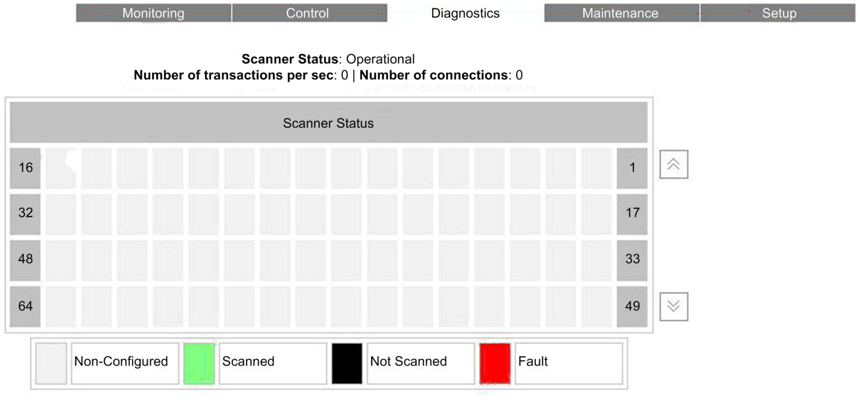
The **Scanner Status** web page displays read-only data describing the current state of the 140NOC78•00 head module in its role as I/O scanner.

Scanner Status Display

The top of the page displays the following general diagnostic information about the scanner:

- I/O scanning status
  - A value of **Operational** indicates that the values in the **Scanner Status** grid are reporting the state of scanned devices.
  - A value of **Stopped** indicates the local system is not scanning. In this case, any data that appears in the **Scanner Status** grid is meaningless.
- Number of transactions per second
- Number of connections

The **Scanner Status** web page looks like this:



In the **Scanner Status** grid, the colors that appear in each block indicate the following states for specific remote devices:

- GREEN indicates that a device is being scanned
- BLACK indicates that I/O scanning of the specific device has been intentionally disabled
- GRAY indicates an device that is not configured
- RED indicates a suspect device

NOTE: A green **Scanner Status** indicator in the grid can remain green for a remote scanned device after the Ethernet cable is detached from that device. This situation can occur if the health timeout value for that device is set to 0.
To avoid this result and to help promote the accurate reporting of I/O scanning health, configure an operational health timeout value in the range 1...65535 (in 1 ms increments).

The grid also indicates the protocol used to communicate with the remote device:
- MB: indicates a Modbus TCP connection
- EIP: indicates an EtherNet/IP connection

To open this page:

Step	Action
1	Starting at the **Home page**, click the **Diagnostics** main menu item. **Result**: The **Diagnostics** page opens.
2	On the left side of the **Diagnostics** page, select **Ethernet → Scanner Status**.
3	If necessary, type in the HTTP web access password. **NOTE:** The default password is **USER**.

Messaging

Introduction

The **Messaging** page provides current information on the open TCP connections on port 502.

Messaging Display

The top of the page displays the number of messages sent and received by local port 502.

The **Messaging** page looks like this:

Monitoring | Control | Maintenance | Setup

Number of Messages sent: 633 | **Number of Messages received**: 634

Conn.	Remote address	Remote port	Local port	Type	Mess. Sent	Mess. Received	Error Sent
1	127.0.0.1	51791	502	Modbus TCP	201	201	4
2	127.0.0.1	51892	502	Modbus TCP	140	140	0
3	172.168.30.199	1819	502	Modbus TCP	17	18	1

The display grid provides the following information about each active connection:

- **Conn.:** the connection number: 1 to 64
- **Remote address:** the IP address of the remote device
  NOTE: If the remote device includes an embedded web server, click the **Remote address** to open that server and view the remote device's web pages.
- **Remote port:** the TCP port for the connection on the remote device
- **Local port:** the TCP port for the connection on the Ethernet communication module
- **Type:** the connection type (EtherNet/IP or Modbus TCP)
- **Mess. Sent:** the number of messages transmitted over this connection
- **Mess. Received:** the number of messages received by this connection
- **Error Sent:** the number of events detected on this connection

NOTE:

- Following a request to close a connection, the PLC may hold the connection open in its memory for a few minutes, during which the display will reflect the open connection.
- The **Number of Messages received** is not reset after a port 502 connection is closed. Therefore, the count indicates the total number of messages that have been received since the module was started.

To open this page:

Step	Action
1	Starting at the **Home page**, click the **Diagnostics** main menu item. **Result**: The **Diagnostics** page opens.
2	On the left side of the **Diagnostics** page, select **Ethernet** → **Messaging**.
3	If necessary, type the HTTP web access password. **NOTE:** The default password is **USER**.

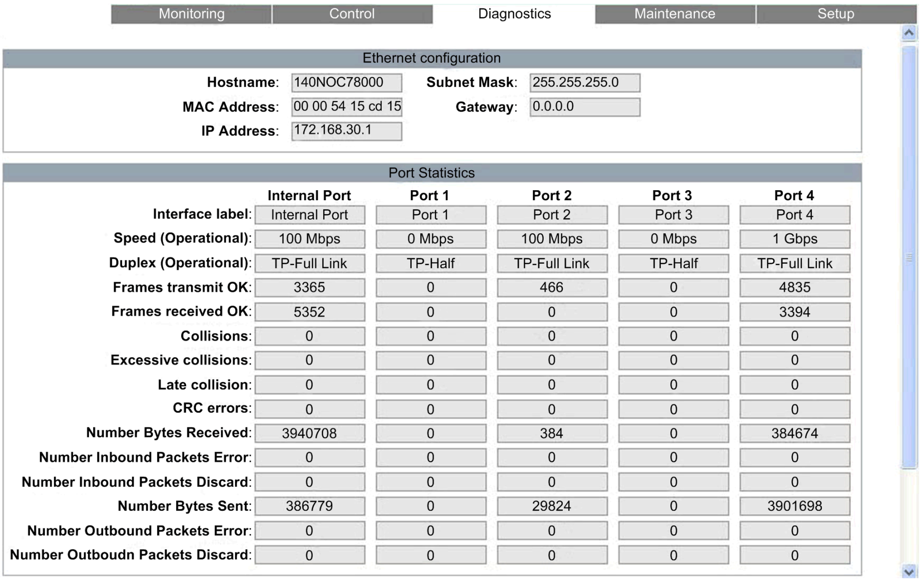
Ethernet Statistics

Introduction

The **Ethernet Statistics** page provides information about the status, transmit and receive statistics, and detected errors for the web server embedded in the 140NOC78•00 head module.

Ethernet Statistics Display

The **Ethernet Statistics** page looks like this:



	Internal Port	Port 1	Port 2	Port 3	Port 4
Interface label:	Internal Port	Port 1	Port 2	Port 3	Port 4
Speed (Operational):	100 Mbps	0 Mbps	100 Mbps	0 Mbps	1 Gbps
Duplex (Operational):	TP-Full Link	TP-Half	TP-Full Link	TP-Half	TP-Full Link
Frames transmit OK:	3365	0	466	0	4835
Frames received OK:	5352	0	0	0	3394
Collisions:	0	0	0	0	0
Excessive collisions:	0	0	0	0	0
Late collision:	0	0	0	0	0
CRC errors:	0	0	0	0	0
Number Bytes Received:	3940708	0	384	0	384674
Number Inbound Packets Error:	0	0	0	0	0
Number Inbound Packets Discard:	0	0	0	0	0
Number Bytes Sent:	386779	0	29824	0	3901698
Number Outbound Packets Error:	0	0	0	0	0
Number Outboudn Packets Discard:	0	0	0	0	0

Click the **Reset counters** button to reset the counting statistics to zero.

To open this page:

Step	Action
1	Starting at the **Home page**, click the **Diagnostics** main menu item. **Result**: The **Diagnostics** page opens.
2	On the left side of the **Diagnostics** page, select **Ethernet** → **Ethernet Statistics**.
3	If necessary, type in the HTTP web access password. **NOTE:** The default password is **USER**.

Ethernet Statistics

The **Ethernet Statistics** page displays the following data for the Ethernet communication module.

Ethernet configuration data:

Hostname	The name assigned to the communication module
MAC Address	The factory assigned Media Access Control (MAC) address, consisting of 6 hexidecimal octet values
IP Address	The Internet Protocol (IP) address *(see Quantum using EcoStruxure™ Control Expert, 140 NOC 771 01 Ethernet Communication Module, User Manual)* that has been assigned to the communication module
Subnet Mask	The subnet mask *(see Quantum using EcoStruxure™ Control Expert, 140 NOC 771 01 Ethernet Communication Module, User Manual)* that has been assigned to the communication module
Gateway Address	The IP address of the remote device *(see Quantum using EcoStruxure™ Control Expert, 140 NOC 771 01 Ethernet Communication Module, User Manual)*, if any, that serves as a gateway to the communication module

Port Statistics:

Speed (Operational)	Baud rate: 0, 10 or 100 Mbits/second
Duplex (Operational)	Twisted Pair—Full Duplex Link, or Twisted Pair—Half Duplex Link
Frames transmit OK	The number of frames that have been successfully transmitted
Frames received OK	The number of frames that have been successfully received
Collisions	The number of times a collision between two successfully transmitted packets was detected on the link
Excessive collisions	The number of times the transmitter has not succeeded after 16 attempts to transmit a frame, due to repeated collisions
Late collisions	The number of times a collision was detected after the slot time of the channel had elapsed
CRC errors	The number of times a CRC (FCS) error was detected on an incoming frame
Number Bytes Received	The number of inbound bytes received on the interface
Number Inbound Packets Error	The number of inbound packets that contain detected errors (not included in discards)
Number Inbound Packets Discard	The number of inbound packets received on the interface, but discarded
Number Bytes Sent	The number of outbound bytes transmitted on the interface
Number Outbound Packets Error	The number of outbound packets that contain detected errors (not included in discards)
Number Outbound Packets Discard	The number of outbound packets discarded while attempting to send them

QoS Configuration

Introduction

The 140NOC78•00 head module supports the OSI layer 3 Quality of Service (QoS) standard defined in RFC-2475. When the QoS is enabled, the module adds a *differentiated services code point* (DSCP) tag to each Ethernet packet it transmits, thereby indicating the priority of that packet.

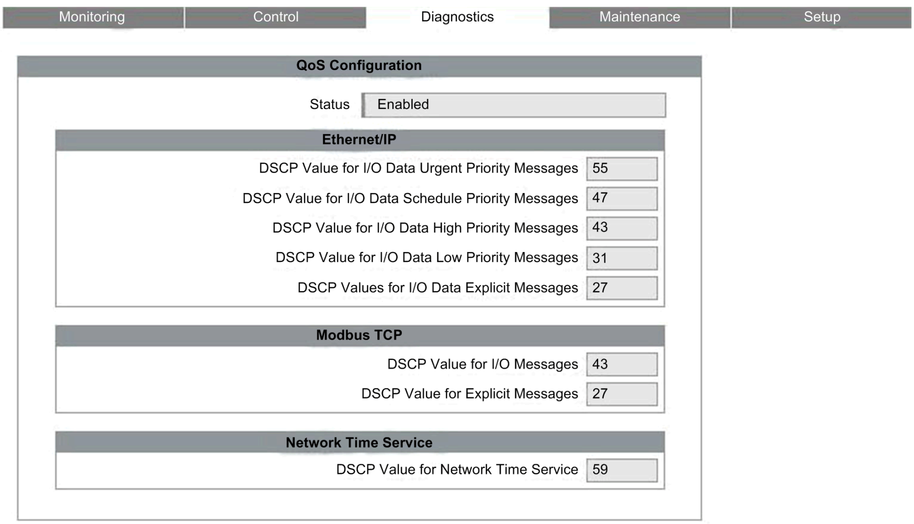
The **QoS Configuration** page displays the following:

- status of the QoS Ethernet packet tagging service, enabled or disabled
- the QoS service configuration settings

NOTE: The QoS service is enabled in the Services page *(see Quantum using EcoStruxure™ Control Expert, 140 NOC 771 01 Ethernet Communication Module, User Manual)*, and the configuration settings are input in the QoS page *(see page 139)*, of the Control Expert Ethernet Configuration Tool.

QoS Configuration Display

The QoS Configuration page looks like this:



This page is read-only.

To open this page:

Step	Action
1	Starting at the **Home page**, click the **Diagnostics** main menu item. **Result**: The **Diagnostics** page opens.
2	On the left side of the **Diagnostics** page, select **Ethernet** → **QoS Configuration**.
3	If necessary, type in the HTTP web access password. **NOTE:** The default password is **USER**.

Redundancy

Introduction

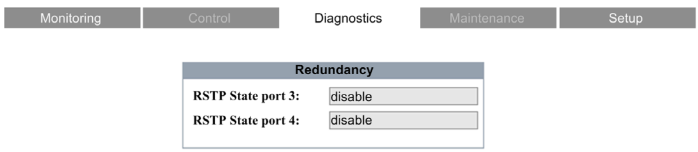
Use the **Redundancy** page to enable and disable the Rapid Spanning Tree Protocol (RSTP) for switch ports 3 and 4.

NOTE: Because only switch ports 3 and 4 support RSTP redundancy, use ports 3 and 4 to connect the 140NOC78•00 head module to the network, and ports 1 and 2 for connections to local devices.

The RSTP service creates a loop-free logical network path for Ethernet devices that are part of a topology that includes redundant physical paths, and automatically restores network communication by activating redundant links in the event the network experiences a service interruption.

Redundancy Display

The **Redundancy** page looks like this:



To open this page:

Step	Action
1	Starting at the **Home page**, click the **Diagnostics** main menu item. **Result**: The **Diagnostics** page opens.
2	On the left side of the **Diagnostics** page, select **Ethernet** → **Switch** → **Redundancy**.
3	If necessary, type in the HTTP web access password. **NOTE:** The default password is **USER**.

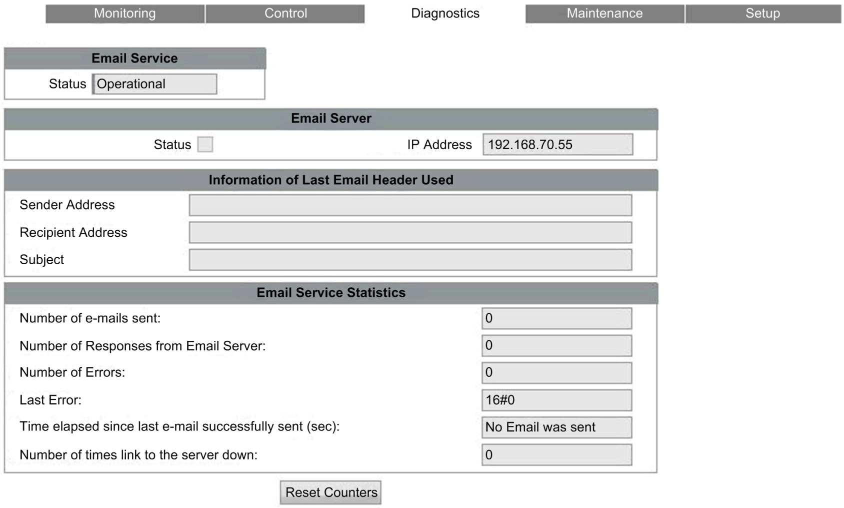
Email Diagnostics

Diagnosing SMTP Transmissions

Use the **SMTP Diagnostics** web page to display dynamically generated data describing the 140NOC78•00 head module Email transmissions.

NOTE: The Email service is enabled in the **Services** page, and the configuration settings are input in the **SMTP Configuration** page of the module DTM.

The **SMTP Diagnostics** web page looks like this:



Click the **Reset Counter** button to reset the **Email Service Statistics** to 0.

To open this page:

Step	Action
1	Starting at the **Home page**, click the **Diagnostics** main menu item. **Result**: The **Diagnostics** page opens.
2	On the left side of the **Diagnostics** page, select **Ethernet → SMTP Diagnostics**.
3	If necessary, type the HTTP web access password. **NOTE:** The default password is **USER**.

Email Diagnostic Parameters

Electronic mail notification service parameters include the following:

Parameter	Description
Email Service:	
Status	The status of this service in the Ethernet communication module: • Operational • Service Disabled
Email Server:	
Status	The connection status between Ethernet communication module and the SMTP server: • check mark = connected • no check mark = not connected **NOTE:** Status is checked at start-up and at least every 30 minutes after start-up.
IP Address	IP address of the SMTP server
Information of Last Email Header Used:	
Sender Address:	Content of the *From* field in the last used Email header
Recipient Address:	Content of the *To* field in the last used Email header
Subject:	Content of the *Subject* field in the last used Email header
Email Service Statistics:	
Number of Emails Sent	Total number of Emails sent and successfully acknowledged by the SMTP server.
Number of Responses from Email Server	Total number of responses received from the SMTP server
Number of Errors	Total number of Emails that either: • could not be sent • were sent but were not successfully acknowledged by the SMTP server
Last Error	Hexadecimal code describing the reason for the last unsuccessful Email transmission *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)*. The value “0” indicates no unsuccessful transmissions.
Time elapses since last Email successfully sent (sec)	Counts the number of seconds since the last Email was successfully sent.
Number of times link to the server down	Number of times the SMTP server could not be reached. (Link checked every 30 minutes.)

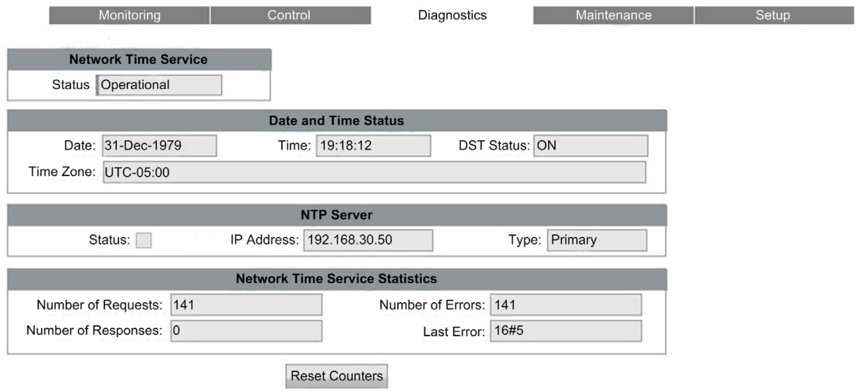
Network Time Service Diagnostics

Diagnosing the Network Time Service

Use the **Network Time Service Diagnostic** web page to display dynamically generated data describing the operation of the network time protocol (NTP) service that you configured in the Network Time Service page *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)* in Control Expert.

NOTE: The Network Time Service is enabled in the **Services** page, and the configuration settings are input in the **Network Time Service Configuration** page of the module DTM.

The **NTP Diagnostics** web page looks like t his:



Click the **Reset Counter** button to reset the **Network Time Service Statistics** to 0.

To open this page:

Step	Action
1	Starting at the **Home page**, click the **Diagnostics** main menu item. **Result**: The **Diagnostics** page opens.
2	On the left side of the **Diagnostics** page, select **Ethernet** → **NTP Diagnostics**.
3	If necessary, type in the HTTP web access password. **NOTE:** The default password is **USER**.

Network Time Service Diagnostic Parameters

Time synchronization service parameters are in the table:

Parameter	Description
Network Time Service:	
Status	Operational status of the service in the module: ● Operational ● Service Disabled
Date and Time Status:	
Date:	System date
Time:	System time **NOTE:** Red text indicates the network time server is not available.
DST Status	The actual working status of the automatic daylight savings service: ● ON = automatic adjustment of daylight savings is enabled and the current date and time reflect the daylight savings time adjustment ● OFF = automatic adjustment of daylight savings is disabled; or automatic adjustment of daylight savings is enabled, but the current date and time may not reflect the daylight savings time adjustment
Time Zone	Time zone plus or minus Universal Time, Coordinated (UTC)
NTP Server:	
Status	Connection status of the NTP server: ● check mark = the NTP server is reachable ● no check mark = the NTP server is not reachable
IP Address	The IP address of the NTP server
Type	The NTP server currently active: ● Primary ● Secondary

Parameter	Description
Network Time Service Statistics:	
Number of Requests:	Total number of client requests sent to the NTP server
Number of Responses:	Total number of server responses sent from the NTP server
Number of Errors:	Total number of unanswered NTP requests
Last Error	Last detected error code received from the NTP client: ● 0: good NTP configuration ● 1: late NTP server response (can be caused by excessive network traffic or server overload) ● 2: NTP not configured ● 3: invalid NTP parameter setting ● 4: NTP component disabled ● 5: NTP server is not synchronized (NTP server needs to be synchronized so that the NTP accesses behave as defined in the client NTP settings) ● 7: unrecoverable NTP transmission ● 9: invalid NTP server IP address ● 15: invalid syntax in the custom time zone rules file

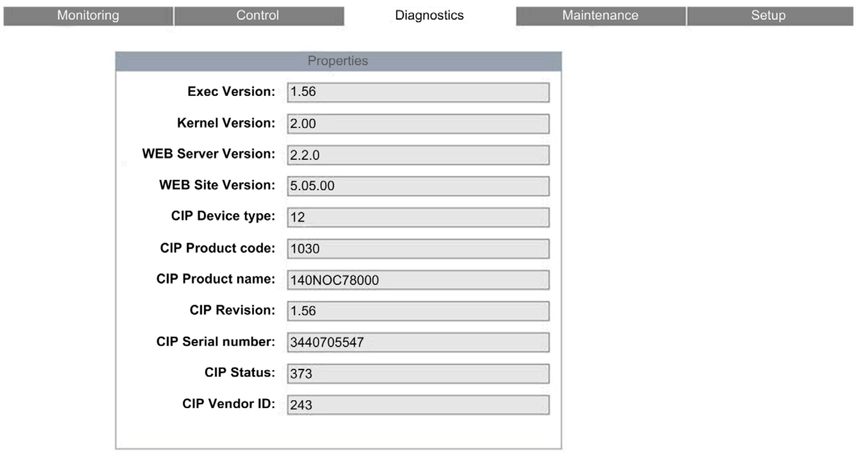
Properties

Introduction

The **Properties** web page displays read-only data describing the particular 140NOC78•00 head module installed in your system.

Properties Display

The **Properties** page looks like this:



To open this page:

Step	Action
1	Starting at the **Home page**, click the **Diagnostics** main menu item. **Result**: The **Diagnostics** page opens.
2	On the left side of the **Diagnostics** page, select **Properties**.
3	If necessary, type in the HTTP web access password. **NOTE:** The default password is **USER**.

Appendices

Appendix A
Detected Error Codes

Overview

This chapter contains a list of codes that describe the status of Ethernet communication module messages.

What Is in This Chapter?

This chapter contains the following topics:

TCP/IP Ethernet Detected Error Codes

TCP/IP Ethernet Detected Error Codes

An event in an MBP_MSTR routine via TCP/IP Ethernet may produce one of the following codes in the MBP_MSTR control block.

TCP/IP Ethernet Hexadecimal Detected Error Codes

TCP/IP Ethernet hexadecimal detected error codes include:

Code (hexadecimal)	Meaning
16#1001	Abort by user
16#2001	An operation type that is not supported has been specified in the control block
16#2002	One or more control block parameters were modified while the MSTR element was active (this only applies to operations which require several cycles for completion). Control block parameters my only be modified in inactive MSTR components.
16#2003	Invalid value in the length field of the control block
16#2004	Invalid value in the offset field of the control block
16#2005	Invalid value in the length and offset fields of the control block
16#2006	Unauthorized data field on slave
16#2007	Invalid slot number in the configuration routing register **Example**: 253 for 140 CRP 312 00 slot number
16#2008	Unauthorized network routing path on slave
16#200E	The control block is not assigned, or parts of the control block are located outside of the %MW (4x) range.
16#200F	The space allocated for the CIP response is too small.
16#3000	Generic Modbus exception response
16#3001	Slave does not support requested operation
16#3002	Non-existing slave registers were requested
16#3003	An unauthorized data value was requested
16#3005	Slave has accepted a lengthy program command
16#3006	Function cannot currently be carried out: lengthy command running
16#3007	Slave has rejected lengthy program command
16#4001	Inconsistent response by Modbus slave
16#F001	Module is resetting
16#F002	Component not fully initialized

Modbus TCP Explicit Messaging Detected Error Codes

Modbus TCP Detected Error Codes

An event in an MBP_MSTR routine via Modbus TCP may produce one of the following detected error codes in the MBP_MSTR control block.

Modbus TCP Hexadecimal Detected Error Codes

Modbus TCP hexadecimal detected error codes include:

Code (hexadecimal)	Meaning
16#5101	No resources
16#5102	Bad IP address
16#5103	Transaction timed out
16#5104	Concurrent connections or transactions limit reached
16#5105	Remote address not allowed
16#5106	No route to host
16#5107	Remote host is down
16#5108	Connection reset by peer
16#5109	Network is down
16#5301	• No resources available — or — • Module not ready or initializing
16#510A	Connection refused
16#510B	Connection timed out

EtherNet/IP Implicit or Explicit Messaging Detected Error Codes

Introduction

If an `MBP_MSTR` function block does not execute an EtherNet/IP explicit message, Control Expert displays a hexadecimal detected error code. This code can describe:

- an EtherNet/IP event
- a TCP/IP Ethernet event

Refer to the topic TCP/IP Ethernet detected error codes *(see page 374)* for a description of those codes.

EtherNet/IP Detected Error Codes

EtherNet/IP hexadecimal detected error codes include:

Code	Description
16#800D	Timeout on the explicit message request
16#8015	Either: ● Nor resources to handle the message, or ● Internal event: no buffer available, no link available, impossible to send to the TCP task
16#8018	Either: ● Another explicit message for this device is in progress, or ● TCP connection or encapsulation session in progress
16#8030	Timeout on the Forward_Open request
Note: The following 16#81xx events are Forward_Open response detected error codes that originate at the remote target and are received via the CIP connection.	
16#8100	Connection in use or duplicate Forward_Open
16#8103	Transport class and trigger combination not supported
16#8106	Ownership conflict
16#8107	Target connection not found
16#8108	Invalid network connection parameter
16#8109	Invalid connection size
16#8110	Target for connection not configured
16#8111	RPI not supported
16#8113	Out of connections
16#8114	Vendor ID or product code mismatch
16#8115	Product type mismatch
16#8116	Revision mismatch
16#8117	Invalid produced or consumed application path
16#8118	Invalid or inconsistent configuration application path

Code	Description
16#8119	Non-Listen Only connection not opened
16#811A	Target object out of connections
16#811B	RPI is smaller than the production inhibit time
16#8123	Connection timed out
16#8124	Unconnected request timed out
16#8125	Parameter event in unconnected request and service
16#8126	Message too large for unconnected_send service
16#8127	Unconnected acknowledge without reply
16#8131	No buffer memory available
16#8132	Network bandwidth not available for data
16#8133	No consumed connection ID filter available
16#8134	Not configured to send scheduled priority data
16#8135	Schedule signature mismatch
16#8136	Schedule signature validation not possible
16#8141	Port not available
16#8142	Link address not valid
16#8145	Invalid segment in connection path
16#8146	Event in Forward_Close service connection path
16#8147	Scheduling not specified
16#8148	Link address to self invalid
16#8149	Secondary resources unavailable
16#814A	Rack connection already established
16#814B	Module connection already established
16#814C	Miscellaneous
16#814D	Redundant connection mismatch
16#814E	No more user-configurable link consumer resources: the configured number of resources for a producing application has reached the limit
16#814F	No more user-configurable link consumer resources: there are no consumers configured for a producing application to use
16#8160	Vendor specific
16#8170	No target application data available
16#8171	No originator application data available
16#8173	Not configured for off-subnet multicast
16#81A0	Event in data assignment
16#81B0	Optional object state event

Code	Description
16#81C0	Optional device state event
Note: All 16#82xx events are register session response detected error codes.	
16#8200	Target device does not have sufficient resources
16#8208	Target device does not recognize message encapsulation header
16#820F	Reserved or unknown event from target

Glossary

A

adapter

The target of real-time I/O data connection requests from scanners. It cannot send or receive real-time I/O data unless it is configured to do so by a scanner, and it does not store or originate the data communications parameters necessary to establish the connection. An adapter accepts explicit message requests (connected and unconnected) from other devices.

advanced mode

A selection in Control Expert that displays expert-level configuration properties that help define Ethernet connections. To maintain system performance, confirm that advanced mode properties are configured only by persons with a solid understanding of communication protocols.

ARP

(*address resolution protocol*) A request and reply protocol used for resolution of network layer addresses into link layer addresses, a function in multiple-access networks.

C

CIP™

(*common industrial protocol*) A comprehensive suite of messages and services for the collection of manufacturing automation applications (control, safety, synchronization, motion, configuration and information). CIP allows users to integrate these manufacturing applications with enterprise-level Ethernet networks and the internet. CIP is the core protocol of EtherNet/IP.

control network

An Ethernet-based network containing PLCs, SCADA systems, an NTP server, PCs, AMS, switches, etc. Two kinds of topologies are supported:

- flat — Devices in this network belong to the same subnet.
- 2 levels — The network is split into an operation network and an inter-controller network. These 2 networks can be physically independent, but are generally linked by a routing device.

D

DDT

(*derived data type*) A set of elements with the same type (`array`) or with different types (structure).

determinism

For a defined application and architecture, the ability to predict that the delay between an event (change of an input value) and the corresponding change of an output state is a finite time *t*, smaller than the time required for your process to run correctly.

device network

An Ethernet-based network within a remote I/O network that contains both remote I/O and distributed I/O devices. Devices connected on this network follow specific rules to allow remote I/O determinism.

DHCP

(*dynamic host configuration protocol*) An extension of the BOOTP communications protocol that provides for the automatic assignment of IP addressing settings (ncluding IP address, subnet mask, gateway IP address, and DNS server names). DHCP does not require the maintenance of a table identifying each network device. The client identifies itself to the DHCP server using either its MAC address, or a uniquely assigned device identifier. The DHCP service utilizes UDP ports 67 and 68.

distributed I/O cloud

A group of distributed I/O devices connected either to a non-ring port on a DRS or to a distributed I/O communications module in the local rack. Distributed I/O clouds are single-point connections to the Ethernet I/O network and are not required to support RSTP.

distributed I/O device

Any Ethernet device (Schneider Electric device, PC, servers, or third-party devices) that supports I/O exchange with a PLC or other Ethernet communication service.

DRS

(*dual-ring switch*) A ConneXium extended managed switch with one of several possible predefined configurations downloaded to it so that it can participate in an Ethernet I/O network. A DRS provides 2 RSTP-enabled ring connections, one for the main ring and one for a sub-ring. It also manages QoS, which provides a predictable level of performance for both remote I/O and distributed I/O traffic on the same I/O network.

DRSs require a firmware version 6.0 or later.

DTM

(*device type manager*) A device driver running on the host PC. It provides a unified structure for accessing device parameters, configuring and operating the devices, and troubleshooting the network. DTMs can range from a simple graphical user interface (GUI) for setting device parameters to a highly sophisticated application capable of performing complex real-time calculations for diagnosis and maintenance purposes. In the context of a DTM, a device can be a communications module or a remote device on the network.

See *FDT*.

E

EDS

(*electronic data sheet*) Simple text files that describe the configuration capabilities of a device. EDS files are generated and maintained by the manufacturer of the device.

EtherNet/IP™

A network communication protocol for industrial automation applications that combines the standard internet transmission protocols of TCP/IP and UDP with the application layer common industrial protocol (CIP) to support both high speed data exchange and industrial control. EtherNet/IP employs electronic data sheets (EDS) to classify each network device and its functionality.

explicit messaging

TCP/IP-based messaging for Modbus TCP and EtherNet/IP. It is used for point-to-point, client/server messages that include both data (typically unscheduled information between a client and a server) and routing information. In EtherNet/IP, explicit messaging is considered class 3 type messaging, and can be connection-based or connectionless.

extended distributed I/O network

An Ethernet-based network containing distributed I/O devices located on an existing distributed I/O network that participate in an Ethernet remote I/O network through use of an *extended port* on a control network head module.

F

FDR

(*fast device replacement*) A service that uses configuration software to replace a device.

FDT

(*field device tool*) The technology that harmonizes communication between field devices and the system host.

H

high-capacity daisy chain loop

Often referred to as HCDCL, a high-capacity daisy chain loop uses DRSs to extend the distance between remote I/O drops or connect sub-rings (containing remote I/O drops or distributed I/O devices) and/or distributed I/O clouds to the Ethernet remote I/O network.

HMI

(*human machine interface*) An HMI is a device that displays process data to a human operator, who in turn uses the HMI to control the process.

An HMI is typically connected to a SCADA system to provide diagnostics and management data, such as scheduled maintenance procedures and detailed schematics for a particular machine or sensor.

Hot Standby

A high-availability control system with a second (standby) PLC that maintains up-to-date system status. If the primary PLC becomes inoperable, the standby PLC takes control of the system.

I

implicit messaging

UDP/IP-based class 1 connected messaging for EtherNet/IP. Implicit messaging maintains an open connection for the scheduled transfer of control data between a producer and consumer. Because an open connection is maintained, each message contains primarily data, without the overhead of object information, and a connection identifier.

independent distributed I/O network

An Ethernet-based network containing distributed I/O devices located on an existing distributed I/O network that participate in the control network only of an Ethernet remote I/O network.

interlink port

An Ethernet port on Ethernet remote I/O head modules allowing direct connection of distributed I/O modules to the remote I/O network and transparency between a control network and the Ethernet remote I/O network.

isolated distributed I/O network

An Ethernet-based network containing distributed I/O devices that do not participate in an Ethernet remote I/O network.

L

local rack

A Quantum rack containing the controller, a power supply, and an Ethernet remote I/O head module. A local rack consists of 1 or 2 racks, the main rack (containing the remote I/O head module) and an optional extended rack. A Quantum Ethernet remote I/O network requires 1 local rack on the main ring.

M

MAST

A master processor task that is run through its programming software. The MAST task has 2 sections:
- IN: Inputs are copied to the IN section before execution of the MAST task.
- OUT: Outputs are copied to the OUT section after execution of the MAST task.

N

NTP

(*network time protocol*) Protocol for synchronizing computer system clocks. The protocol uses a jitter buffer to resist the effects of variable latency.

P

PLC

programmable logic controller. The PLC is the brain of an industrial manufacturing process. It automates a process as opposed to relay control systems. PLCs are computers suited to survive the harsh conditions of the industrial environment.

R

remote I/O drop

One of the 3 types of remote I/O devices in an Ethernet remote I/O network. A remote I/O drop is a Quantum or an X80 rack of I/O modules that are connected to an Ethernet remote I/O network and managed by an Ethernet remote adapter module. A drop can be a single rack or a rack with an extension rack.

remote I/O main ring

The main ring of an Ethernet remote I/O network. The ring contains remote I/O devices and a local rack (containing a controller, a power supply module, and an Ethernet remote I/O head module).

remote I/O network

An Ethernet-based network that contains 1 standalone PLC or one Hot Standby system and remote I/O devices. There are 3 types of remote I/O devices: a local rack, a remote I/O drop, and a ConneXium extended dual-ring switch (DRS). Distributed I/O devices may also participate in a remote I/O network via connection to DRSs.

RPI

(requested packet interval) The time period between cyclic data transmissions requested by the scanner. EtherNet/IP devices publish data at the rate specified by the RPI assigned to them by the scanner, and they receive message requests from the scanner at each RPI.

RSTP

(*rapid spanning tree protocol*) A protocol that allows a network design to include spare (redundant) links to provide automatic backup paths if an active link stops working, without the need for loops or manual enabling/disabling of backup links.

S

SCADA

(*supervisory control and data acquisition*) SCADA systems are computer systems that control and monitor industrial, infrastructure, or facility-based processes (examples: transmitting electricity, transporting gas and oil in pipelines, and water distribution).

service port

A dedicated Ethernet port on the Quantum Ethernet remote I/O modules. The port may support 3 major functions (depending on the module type):

- port mirroring — for diagnostic use
- access — for connecting HMI/Control Expert/ConneXium Network Manager to the PLC
- extended — to extend the device network to another subnet
- disabled — disables the port, no traffic is forwarded in this mode

simple daisy chain loop

A daisy chain loop, often referred to as SDCL, that contains remote I/O devices only (no switches or distributed I/O devices). This topology consists of a local rack (containing a remote I/O head module), and 1 or more remote I/O drops (each drop containing a remote I/O adapter module).

SNMP

(*simple network management protocol*) Protocol used in network management systems to monitor network-attached devices for events. The protocol is part of the internet protocol suite (IP) as defined by the internet engineering task force (IETF), which consists of network management guidelines, including an application layer protocol, a database schema, and a set of data objects.

sub-ring

An Ethernet-based network with a loop attached to the main ring, via a DRS. A sub-ring may contain either remote I/O or distributed I/O devices.

T

TCP/IP

Also known as *internet protocol suite*, TCP/IP is a collection of protocols used to conduct transactions on a network. The suite takes its name from 2 commonly used protocols: transmission control protocol and internet protocol. TCP/IP is a connection-oriented protocol that is used by Modbus TCP and EtherNet/IP for explicit messaging.

U

UDP

(*user datagram protocol*) A transport layer protocol that supports connectionless communications. Applications running on networked nodes can use UDP to send datagrams to one another. UDP does not always deliver datagrams as reliable or ordered as those delivered by TCP. However, by avoiding the overhead required for TCP, UDP is faster. UDP may be the preferred protocol for time-sensitive applications, where dropped datagrams are preferable to delayed datagrams. UDP is the primary transport for implicit messaging in EtherNet/IP.

Index

E

F

G

H

I