Modicon M580

BMENOC0321 Control Network Module Installation and Configuration Guide

Original instructions

09/2020

NVE24232.06

www.schneider-electric.com

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

You agree not to reproduce, other than for your own personal, noncommercial use, all or part of this document on any medium whatsoever without permission of Schneider Electric, given in writing. You also agree not to establish any hypertext links to this document or its content. Schneider Electric does not grant any right or license for the personal and noncommercial use of the document or its content, except for a non-exclusive license to consult it on an "as is" basis, at your own risk. All other rights are reserved.

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

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

Validity Note

This document is valid for an M580 system when used with EcoStruxure™ Control Expert 15.0 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 described 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
Modicon M580 Standalone, System Planning Guide for Frequently Used Architectures	HRB62666 (English), HRB65318 (French), HRB65319 (German), HRB65320 (Italian), HRB65321 (Spanish), HRB65322 (Chinese)
Modicon M580, System Planning Guide for Complex Topologies	NHA58892 (English), NHA58893 (French), NHA58894 (German), NHA58895 (Italian), NHA58896 (Spanish), NHA58897 (Chjnese)
Modicon M580 Hot Standby, System Planning Guide for Frequently Used Architectures	NHA58880 (English), NHA58881 (French), NHA58882 (German), NHA58883 (Italian), NHA58884 (Spanish), NHA58885 (Chinese)
Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications	EIO0000002726 (English), EIO0000002727 (French), EIO0000002728 (German), EIO0000002730 (Italian), EIO0000002729 (Spanish), EIO0000002731 (Chinese)
M580 BMENOS0300, Network Option Switch, Installation and Configuration Guide	NHA89117 (English), NHA89119 (French), NHA89120 (German), NHA89121 (Italian), NHA89122 (Spanish), NHA89123 (Chinese)
Modicon M580, Hardware, Reference Manual	EIO0000001578 (English), EIO0000001579 (French), EIO0000001580 (German), EIO0000001582 (Italian), EIO0000001581 (Spanish), EIO0000001583 (Chinese)
Modicon M580, RIO Modules, Installation and Configuration Guide	EIO0000001584 (English), EIO0000001585 (French), EIO0000001586 (German), EIO0000001587 (Italian), EIO0000001588 (Spanish), EIO0000001589 (Chinese),
Modicon M580, Change Configuration on the Fly, User Guide	EIO0000001590 (English), EIO0000001591 (French), EIO0000001592 (German), EIO0000001594 (Italian), EIO0000001593 (Spanish), EIO0000001595 (Chinese)
Modicon X80, Discrete Input/Output Modules, User Manual	35012474 (English), 35012475 (German), 35012476 (French), 35012477 (Spanish), 35012478 (Italian), 35012479 (Chinese)
Modicon X80, BMXEHC0200 Counting Module, User Manual	35013355 (English), 35013356 (German), 35013357 (French), 35013358 (Spanish), 35013359 (Italian), 35013360 (Chinese)

Title of documentation	Reference number
Electrical installation guide	EIGED306001EN (English)
Control Panel Technical Guide, How to protect a machine from malfunctions due to electromagnetic disturbance	CPTG003_EN (English), CPTG003_FR (French)
EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual	35006144 (English), 35006145 (French), 35006146 (German), 35013361 (Italian), 35006147 (Spanish), 35013362 (Chinese)
EcoStruxure™ Control Expert, System Bits and Words, Reference Manual	EIO0000002135 (English), EIO0000002136 (French), EIO0000002137 (German), EIO0000002138 (Italian), EIO0000002139 (Spanish), EIO0000002140 (Chinese)
EcoStruxure™ Control Expert, Operating Modes	33003101 (English), 33003102 (French), 33003103 (German), 33003104 (Spanish), 33003696 (Italian), 33003697 (Chinese)
EcoStruxure™ Control Expert, Installation Manual	35014792 (English), 35014793 (French), 35014794 (German), 35014795 (Spanish), 35014796 (Italian), 35012191 (Chinese)
Web Designer for FactoryCast, User Manual	35016149 (English), 35016150 (French)
Modicon Controllers Platform Cyber Security, Reference Manual	EIO0000001999 (English), EIO0000002001 (French), EIO0000002000 (German), EIO0000002002 (Italian), EIO0000002003 (Spanish), EIO0000002004 (Chinese)

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

Chapter 1
BMENOC0321 Module Characteristics

Introduction

This chapter describes the BMENOC0321 Ethernet control network module.

This module is the preferred entry point from the control network to a device network (including RIO and distributed equipment) managed by a Modicon M580 PAC. The module provides network transparency and provides a direct Ethernet connection between the control room subnetwork and the automation devices subnetwork.

What Is in This Chapter?

This chapter contains the following sections:

Section 1.1
Introduction to the BMENOC0321 Module

What Is in This Section?

This section contains the following topics:

Module Description

Introduction

The BMENOC0321 control network module is installed on a local Ethernet backplane in a Modicon M580 system. With the Ethernet backplane enabled *(see page 88)*, the BMENOC0321 provides access to the Modicon M580 CPU's network (through the external ports of the CPU).

Schneider Electric recommends the installation of a maximum of two BMENOC0321 control network modules in a Modicon M580 system to provide Ethernet transparency between a control network (for example, a SCADA system) and an M580 device network. You can enable the IP forwarding service *(see page 111)* on only one BMENOC0321 module per local rack.

NOTE: Do not mount the BMENOC0321 module on a BMX (X Bus only) backplane. The module can power up only on a BME (Ethernet) backplane. Refer to the rack descriptions in the Modicon M580 Hardware Reference Manual.

Ruggedized Version

The BMENOC0321C (coated) equipment is the ruggedized version of the BMENOC0321 (standard) equipment. It can be used at standard temperatures and in harsh chemical environments.

For more information, refer to chapter *Installation in More Severe Environments (see Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications)*.

Altitude Operating Conditions

The characteristics apply to the modules BMENOC0321 and BMENOC0321C for use at altitude up to 2000 m (6560 ft). When the modules operate above 2000 m (6560 ft), apply additional derating.

For detailed information, refer to chapter *Operating and Storage Conditions (see Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications)*.

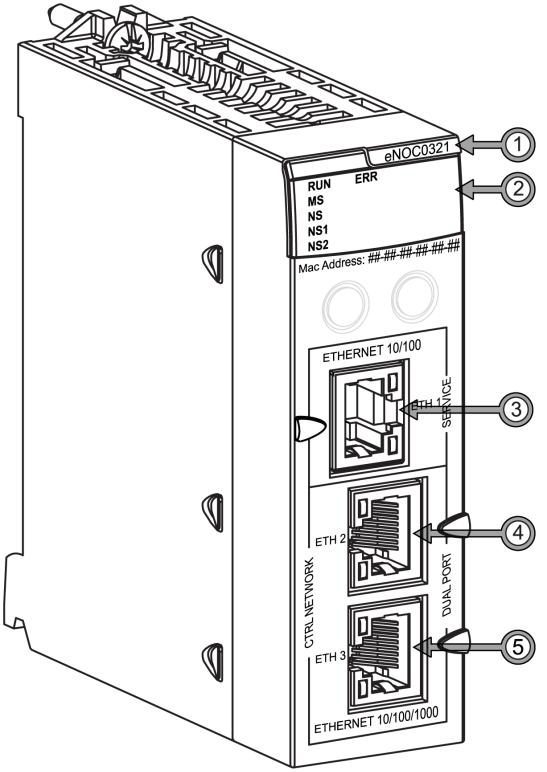
BMENOC0321 and PlantStruxure

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 Modicon M580 offer, in which an M580 local rack communicates with M580 RIO drops and distributed equipment in the same network.

Physical Description

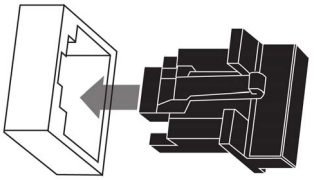
This figure shows the external features of the BMENOC0321 module:



Legend:

Item	Description	Function
1	module name	BMENOC0321
2	LED array	Observe the LED display *(see page 184)* to diagnose the module.
3	SERVICE port (ETH 1)	Use the RJ45 Ethernet connector for a service port. **NOTE:** Refer to the service port configuration *(see page 109)*.
4	control network port (ETH 2)	These RJ45 control network ports provide: ● Ethernet communications (10/100/1000 Mbps) ● connections for distributed device communications ● cable redundancy through a daisy chain loop architecture
5	control network port (ETH 3)	

To keep dust out of unused Ethernet ports, cover the ports with the stopper:



External Ports

The BMENOC0321 module monitors the functionality of network links depending on which links are connected to the network. The module has four external ports (up to three IP addresses).

Port	Type	Description
ETH 1	service	This port supports the diagnosis of Ethernet ports, provides access to external tools and devices (for example, Control Expert, ConneXium Network Manager, HMI, etc.), and provides a connection to a DIO network. The port supports these modes: ● Port mirroring *(see page 109)* ● Access port *(see page 109)* (default) **NOTE:** In access port mode, the IP address of the port is the same as that of the control network. ● Extended device network: In this mode the IP address of the port is in an extended DIO network. ● 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. ● 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. To configure this port, refer to the topic Configuring the Service Port *(see page 109)*.
ETH 2 ETH 3	control network	These two copper ports provide connections for: ● gigabit link for control network communications ● star, loop, or mesh topology **NOTE:** ● These ports support the RSTP redundancy protocol. ● By default, these ports are configured for connection to a control network to be used as a gateway to a device network by servers and workstations in the control room (like SCADA servers and clients).

Dual-Bus Backplane Connector

The dual-bus interface on the back of the BMENOC0321 module connects to the X Bus and Ethernet bus connectors on the Ethernet backplane when you mount the module in the rack *(see page 30)*. The module, therefore, supports both X Bus and Ethernet communication over the backplane:

Bus	Description
X Bus	The BMENOC0321 module uses X Bus communication on the Ethernet backplane to obtain and exchange these data through the CPU: ● configuration data for the BMENOC0321 module ● application and diagnostic data
Ethernet	The BMENOC0321 module uses the Ethernet bus on the Ethernet backplane to manage connectivity to the BMENOC0321 module: ● The BMENOC0321 module provides Ethernet connectivity to the CPU. ● The BMENOC0321 module communicates with Ethernet communication modules on the local rack that manage distributed equipment in the device network. ● The BMENOC0321 module communicates with network devices that are attached to the external ports of the CPU.

Key Module Features

Product Features

These tables describe the key features of the BMENOC0321 module

Diagnostics Features	
PAC Application	Some module diagnostics (I/O connection health, redundancy status, etc.) are available through the PAC application and are updated every CPU cycle.
Local Modbus Server *(see page 220)*	Some module diagnostics (I/O connection, extended health, redundancy status, FDR server, etc.) are available to Modbus clients that read the local Modbus server area with Modbus function code 3 when the unit ID is set to 100 or through Modbus function code 8/21, 8/22, or 43/14.
CIP Objects *(see page 223)*	Some module diagnostics (Ethernet interface, redundancy, Ethernet scanner, etc.) are available through CIP objects that EtherNet/IP devices such as SCADA or HMI can read.
Ethernet Ports	You can diagnose network issues by examining packets to and from Ethernet ports when the service port is configured for port mirroring *(see page 109)*.
Embedded Web Pages *(see page 335)*	Embedded web pages provide diagnostics data through a web browser.

Explicit Messaging: With the DATA_EXCH function block, the BMENOC0321 control network module supports explicit messaging *(see page 150)* through the EtherNet/IP and Modbus TCP protocols.

Firmware Upgrade: The firmware upgrade service *(see page 331)* allows the field upgrade of this module firmware using the Automation Device Maintenance or Unity Loader tool.

Features in Control Expert: You can enable and disable these Ethernet services *(see page 94)* in Control Expert.	Address Server (address and operation parameters)
	SNMP (agent)
	RSTP (cable redundancy)
	Network Time Service (NTP server)
	QoS (DSCP tagging)
	Service Port (control network connections)
	IP Forwarding (network transparency): The module uses the IP forwarding service to separate Ethernet traffic between the control network, the device network, and the embedded network.
	Email (SMTP): The simple mail transfer protocol (SMTP) provides mechanisms that allow controller-based projects to report alarms or events.

Section 1.2
Specifications

What Is in This Section?

This section contains the following topics:

Standards and Certifications

Download

Click the link that corresponds to your preferred language to download standards and certifications (PDF format) that apply to the modules in this product line:

Title	Languages
Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications	• English: *EIO0000002726* • French: *EIO0000002727* • German: *EIO0000002728* • Italian: *EIO0000002730* • Spanish: *EIO0000002729* • Chinese: *EIO0000002731*

Communication Specifications

Introduction

The BMENOC0321 control network module provides support for I/O scanning using EtherNet/IP and Modbus TCP.

These specifications describe the I/O communication and the implicit and explicit messaging capacities of the BMENOC0321 modules.

NOTE: The maximum I/O scanning capability for a BMENOC0321 module is 5,500 packets per second.

I/O Communication Specifications

These tables present the I/O communications features of the BMENOC0321 module.

EtherNet/IP (CIP Implicit Messaging):

Feature		Maximum Capacity
scanner	number of devices	128 (EtherNet/IP devices and local slaves)
	message size	input: 505 bytes (including header) output: 509 bytes (including header)
adapter	number of instances	12 adapter instances
	number of connections	2 connections per instance
	message size	511 bytes (including header)
	inputs	505 bytes (including header)
	outputs	509 bytes (including header)

Modbus TCP (Modbus I/O Scanner):

Feature		Maximum Capacity
registers	number of devices	128 devices shared with EtherNet/IP
	read	125 registers
	write	120 registers
message size	read	250 bytes (125 words) (excluding header)
	write	240 bytes (120 words) (excluding header)

Combined EtherNet/IP Scanner/Adapter and Modbus Scanner

I/O Data Exchange with the CPU		
Feature	**Maximum Capacity**	**Comment**
input data size	4 KB (2 K words)	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 4 KB, but more than 3.5 KB.
output data size	4 KB (2 K words)	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 4 KB, but more than 3.5 KB.

Explicit Messaging Specifications

These tables present the explicit messaging features of the BMENOC0321 module.

NOTE: These tables report the maximum capacity for a single BMENOC0321 module. For more information, refer to the performance characteristics and capacities for M580 CPUs *(see Modicon M580, Hardware, Reference Manual)* in the *Modicon M580 Hardware Reference Manual*.

EtherNet/IP (CIP explicit messaging):

Feature		Maximum Capacity
client	simultaneous requests	16
	message size	1024 bytes
server	simultaneous requests	32
	message size	1024 bytes

Modbus TCP (Modbus explicit messaging):

Feature		Maximum Capacity
client	simultaneous requests	16
	message size	1024 bytes
server	simultaneous requests	32
	message size	1024 bytes

Chapter 2
Installing the BMENOC0321 Module

Introduction

This chapter describes the installation process of the BMENOC0321 Ethernet communications module within a Modicon M580 system.

What Is in This Chapter?

This chapter contains the following topics:

Mounting an Ethernet Communications Module on the Modicon M580 Rack

Introduction

Use these instructions to install an Ethernet communications module in a single slot on the Ethernet backplane.

NOTE: Fitting operations (installation, assembly, and disassembly) are described below.

Before Installing a Module

Before installing the Ethernet communications module, remove the protective cap from the module connector on the rack.

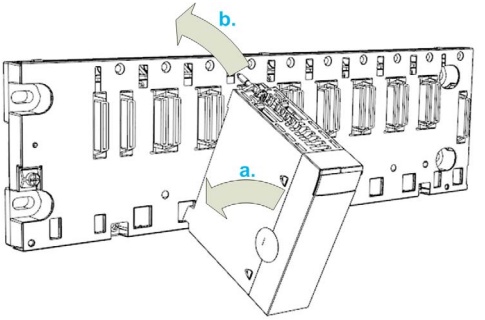
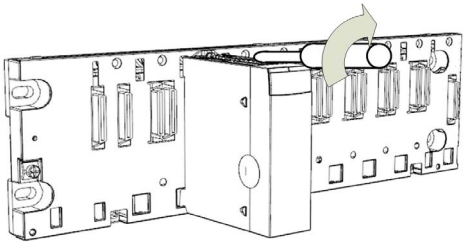
Selecting a Backplane

Install the Ethernet communications module in a single slot on one of these Ethernet backplanes:

Backplane	Description
BMEXBP0400[1]	4-slot Ethernet backplane
BMEXBP0400(H)[1]	4-slot hardened Ethernet backplane
BMEXBP0800[1]	8-slot Ethernet backplane
BMEXBP0800(H)[1]	8-slot hardened Ethernet backplane
BMEXBP1200[1, 2]	12-slot Ethernet backplane
BMEXBP1200(H)[1, 2]	12-slot hardened Ethernet backplane
BMEXBP0602 (H)	10-slot hardened Ethernet and X Bus backplane
BMEXBP1002 (H)	6-slot hardened Ethernet and X Bus backplane
1. In a local rack, slots 0 and 1 are reserved for the CPU. 2. In the 12-slot Ethernet backplane, slots 2, 8, 10, and 11 are X Bus only slots. You can install an Ethernet communication module in any other rack slot.	

Installing the Module on the Rack

Mount the module in a single slot on the backplane:

Step	Action
1	Turn off the power supply to the rack.
2	Remove the protective cover from the module interface on the rack.
3	*a*: Insert the locating pins on the bottom of the module into the corresponding slots in the rack.  *b*: Use the locating pins as a hinge and pivot the module until it is flush with the rack. (The twin connector on the back of the module inserts into the connectors on the rack.)
4	Tighten the retaining screw to hold the module in place on the rack:  Tightening torque: 0.4...1.5 N•m (0.30...1.10 lbf-ft).

Grounding Considerations

⚠⚠ DANGER

ELECTRICAL SHOCK HAZARD

- Switch off the power supply at both ends of the PAC connection, and lock out and tag out both the power sources.
- In case lock out and tag out are not available, ensure that the power sources cannot be inadvertently switched on.
- 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.

Do not apply power to the Ethernet communications module until connections are made at both ends of the Ethernet cable. For example, connect the cable to both the module and another device (adapter module) or a DRS before you turn on the power.

Refer to your system hardware reference manual for details about the DRSs.

Use fiber-optic cable to establish a communications link when it is not possible to equalize the potential between the two grounds.

NOTE: Refer to the ground protection information provided in the *Electrical installation guide* and *Control Panel Technical Guide, How to protect a machine from malfunctions due to electromagnetic disturbance*.

Replacing a Module

Any Ethernet communications module on the rack can be replaced at any time with 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.

Cable Installation

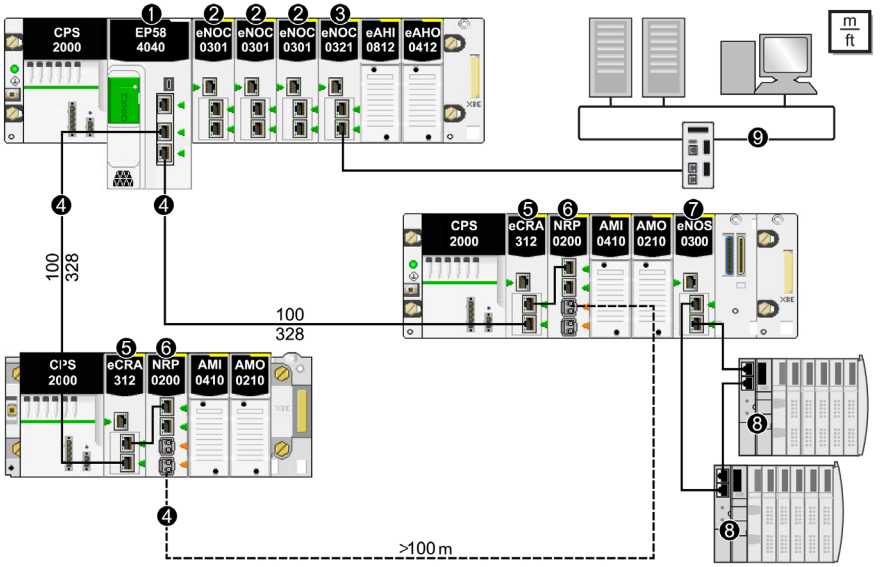
Cable Recommendations

To connect a BMENOC0321 control network module to a control network in a Modicon M580 system, Schneider Electric recommends the use of these cables:

- *10/100 Mbps*: For a communications link less than or equal to 100 Mbps, use CAT5e or CAT6 copper shielded twisted four-pair cables.
- *1000 Mbps*: For a communications link less than or equal to 1000 Mbps, use only CAT6 copper shielded twisted four-pair cables.

Connections Between Devices

This example shows the maximum cable lengths between RIO and DIO devices in an M580 device network. Use copper cable for distances less than or equal to 100 m. User fiber cable for distances greater than 100 m:



1 BME•58•••• CPU connecting the local rack to the main ring
2 BMENOC0301/BMENOC0311 Ethernet communication module on the local rack managing distributed equipment on the device network
3 BMENOC0321 control network module with IP forwarding service enabled
4 RIO main ring
5 BM•CRA312•0 (e)X80 EIO adapter module on an RIO drop on the main ring
6 BMXNRP020• fiber converter module connecting portions of the RIO main ring greater than 100 m.
7 BMENOS0300 network option switch module connecting a DIO sub-ring to the main ring
8 STB island in a DIO sub-ring connected to the main ring via a BMENOS0300 module on an RIO drop
9 control network

Chapter 3
Control Network Interconnectivity

Plan and Design Control Network Interconnectivity

In an M580 system, a single RIO network can include both RIO and distributed equipment. Install a BMENOC0321 control network module on the local rack to connect a new or existing Ethernet control network to a device network that contains RIO modules and distributed equipment.

Configure the BMENOC0321 module in Control Expert to communicate with these devices:

- BME•58•••• CPU
- BMENOC0301/BMENOC0311 Ethernet communication module
- RIO and distributed equipment
- HMI devices
- SCADA programs

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 topics:

How the Control Network Works within an M580 System

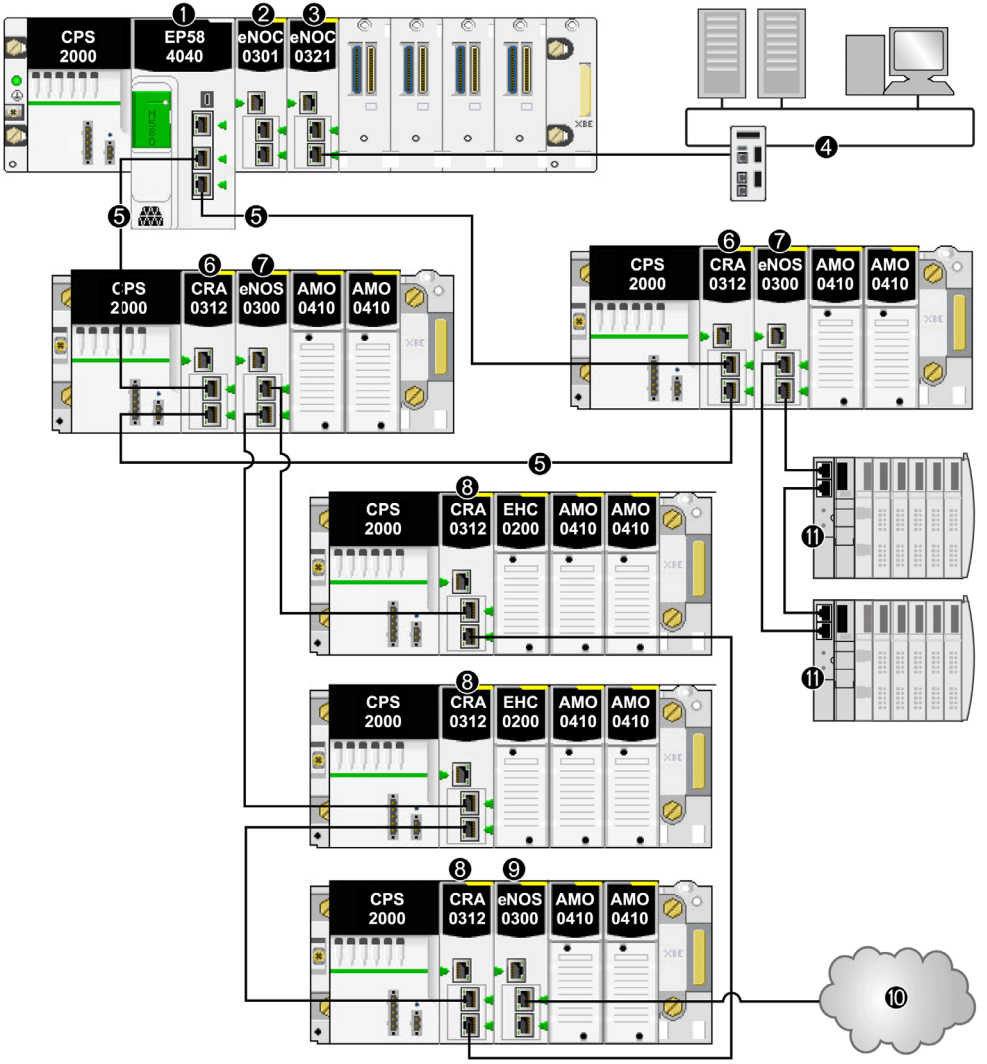
Introduction

The main functionality of the BMENOC0321 control network module is to provide network transparency between a device network (including RIO and distributed equipment) and a control network.

Other functionality of the BMENOC0321 module:

- The module operates in a network that uses the RSTP protocol.
- The module configures IP parameters and device configuration files for I/O devices in the control network.
- The module supports Hot Standby functionality.
- The module scans I/O devices in the control network.
- The module operates in a network that is connected to the control network through a 1000 Mbps connection.
- The module supports IP secure communications (IPsec *(see page 119)*).

In this M580 architecture, a BMENOC0321 control network module connects a control network to a local rack that includes a CPU and a BMENOC0301/BMENOC0311 Ethernet communication module. The BMENOC0321 module provides network transparency between the control network and the device network:



1 BME•58•••• CPU connecting the local rack to the main ring
2 BMENOC0301 Ethernet communication module on the local rack managing distributed equipment on the device network
3 BMENOC0321 control network module
4 control network

5 RIO main ring
6 BM•CRA312•0 (e)X80 EIO adapter module on an RIO drop on the main ring
7 BMENOS0300 network option switch module on an RIO drop
8 BM•CRA312•0 module on an RIO drop on an RIO sub-ring
9 BMENOS0300 module on a drop in an RIO sub-ring
10 DIO cloud connected to an RIO sub-ring via a BMENOS0300 network option switch module
11 STB island in a DIO sub-ring connected to the main ring via a BMENOS0300 module on an RIO drop

Network Characteristics

- There are two subnets, one for the control network and one for the device network.
- The monitoring workstation on the control network can communicate with equipment on the device network through the BMENOC0321 control network module.
- The IP forward feature of the BMENOC0321 module manages Ethernet transparency between the control network and the device network.

How is the BMENOC0321 Module's Default Gateway Used?

As previously stated, if a datagram is targeted outside of a network, the datagram is sent to the default gateway. In an M580 system, the default gateway is the BMENOC0321 module. If the datagram is not targeted to a device in one of the three networks known by the BMENOC0321 module, the datagram is sent to the BMENOC0321 module's default gateway.

NOTE: Double-click the BMENOC0321 module in the Control Expert **PLC bus** to configure the default gateway.

Rules for Connectivity

Introduction

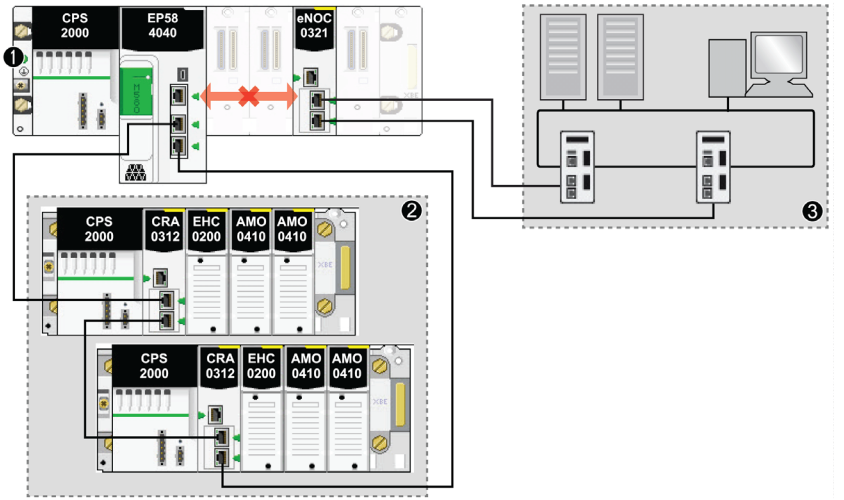
The local rack within an M580 system can have different combinations of Ethernet communication modules. This topic describes the networks that are created when the BMENOC0321 control network module is configured in different ways.

Network Types

A local rack contains one CPU and up to six communication modules, only one of which can have the IP forwarding service enabled *(see page 111)*. The BMENOC0321 module can communicate with other modules and devices in the system for various network combinations depending on the status of the backplane connection (enabled or disabled):

- *disabled:* You cannot enable the IP forwarding service for the BMENOC0321 module when the backplane connection is disabled.
- *enabled:* You can enable the IP forwarding service for the BMENOC0321 module when the backplane connection is enabled.

In this illustration, the backplane connection on the local rack is disabled (red arrow). Therefore, the IP forwarding service is disabled. In this example of isolation, there is no transparency *(see page 42)* between the device network and the control network:

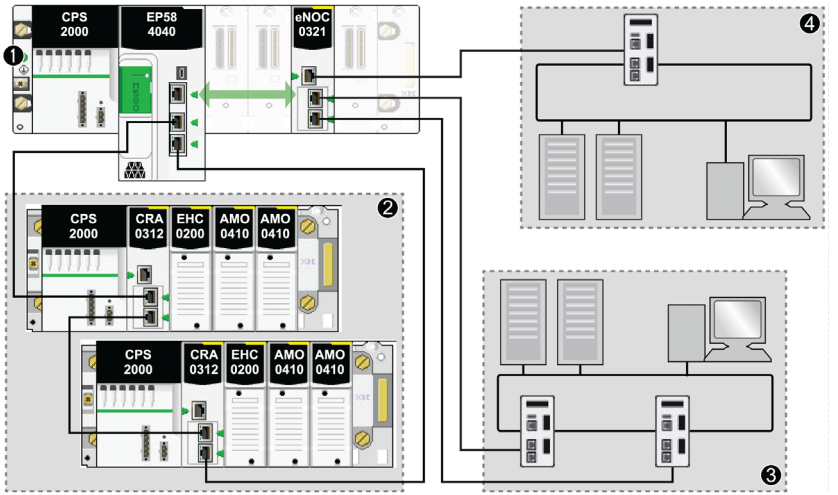


1 The local rack has no Ethernet backplane connection between the M580 CPU (and other devices in the device network) and the BMENOC0321 module with the IP forwarding service disabled.
2 The device network is connected to the CPU on the local rack.
3 The BMENOC0321 module is connected to a control network with redundant links.

In the next illustration, you can enable the IP forwarding service among three configured networks because the backplane connection on the local rack is enabled (green arrow):

Network	Ports
device network	Ethernet backplane port
extended device network	service port (ETH 1)
control network	ETH 2, ETH 3

In this example, the IP forwarding service is enabled to allow network transparency. The BMENOC0321 module uses the backplane connection (green arrow) to communicate with the device network:



1 The local rack includes a backplane connection (green arrow) between the M580 CPU and the BMENOC0321 module with the IP forwarding service enabled.
2 The device network is connected to the CPU on the local rack.
3 The BMENOC0321 module is connected to a control network.
4 The service port on the BMENOC0321 module is connected to an extended device network.

Using Different Services and Protocols

The status (enabled or disabled) of one service or protocol can affect the use of other services and protocols. This table shows the possible combinations of services when they are enabled or disabled for the BMENOC0321 module:

IPsec protocol *(see page 119)*	**IP Forwarding Service** *(see page 111)*	**Ethernet Backplane Port** *(see page 88)*
on	off	off
off	on	on
off	off	on/off

Transparency Functionality

Introduction to Transparency

You can segregate a network into multiple subnetworks to limit user access and increase performance. This usually means that devices in different subnetworks are not able to communicate directly.

You can, however, use the IP forwarding functionality *(see page 36)* of the BMENOC0321 control network module to enable Ethernet network transparency to facilitate seamless communications between devices in different subnetworks. In M580 systems, use a BMENOC0321 module to achieve transparency between different types of devices in the control network, the device network, and the extended device network.

For example, you can run the Control Expert DTM software on a PC that is located in the control network to access configuration data, diagnostic data, and I/O data from devices in the device network (M580 CPU, ATV, TeSys, STB, etc.).

Use Control Expert to configure the IP forwarding service *(see page 111)*.

NOTE: The recommended maximum throughput for a BMENOC0321 module that uses the IP forwarding service is 1,350 packets per second.

Before You Begin

Before you start this example, change your Control Expert configuration to facilitate the use of the IP forwarding service:

Step	Action
1	Enable the IP forwarding service *(see page 111)*.
2	Configure the service port *(see page 109)* as an extended network port.

NOTE: If you download the application via a BMENOC0301/11 module, the module resets at the end of the download, which resets the connection between Control Expert and the module. If you download the application via the USB port on the M580 CPU, the connection is sustained.

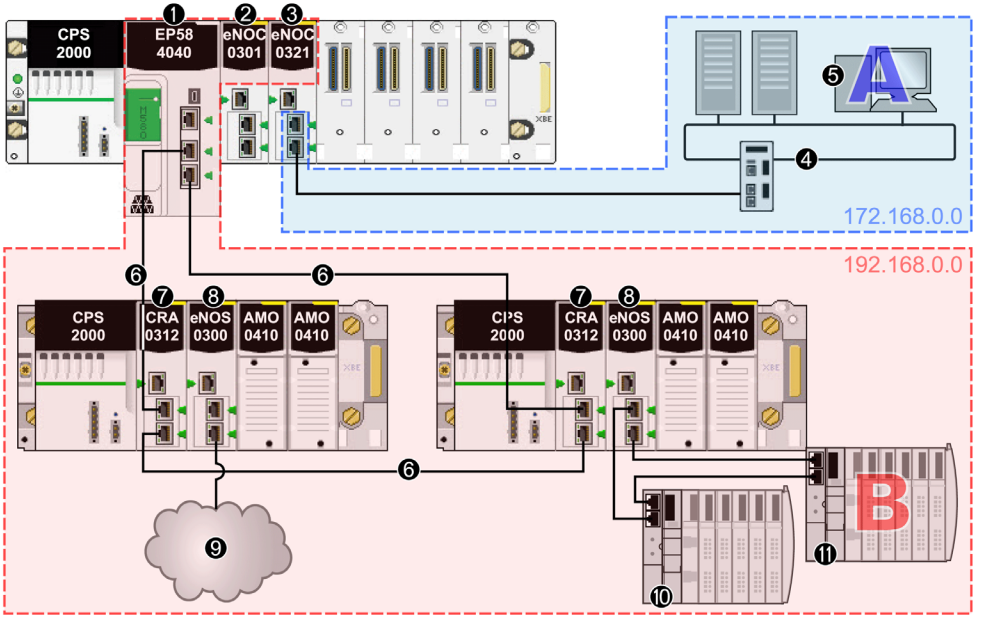
IP Forwarding Example

Suppose you want to provide transparency between the control network and the device network:

- On the control network, host **A** (a PC) use the IP address 172.168.100.1 in subnetwork 172.168.0.0.
- On the device network, host **B** (an Advantys STB module) uses the IP address 192.168.10.200 in subnetwork 192.168.0.0.

To facilitate communications between hosts **A** and **B**, connect the control network and device network physically, as well as logically. The IP forwarding service in the BMENOC0321 module is the interface for this network connection.

In the sample architecture, the IP forwarding service in the BMENOC0321 module provides transparency between the device network and the control network. Host **A** in subnetwork 172.168.00 (blue) can communicate with host **B** in subnet 192.168.0.0 (red) because the BMENOC0321 module has an address in both subnetworks:



1 A BME•58•••• CPU connects the local rack to the main ring.
2 A BMENOC0301 Ethernet communication module is connected to the CPU over the Ethernet backplane (so it is on the same network as the CPU).
3 The IP forwarding service on the BMENOC0321 module has IP addresses in three subnetworks (172.168.0.0 and 192.168.0.0).
4 A control network is in subnetwork 172.168.0.0.
5 A PC (host **A**) is in the control network.
6 The RIO main ring is connected to the CPU.
7 A BM•CRA312•0 (e)X80 EIO adapter module is on an RIO drop on the main ring.
8 BMENOS0300 network option switch modules are on RIO drops.
9 A DIO cloud connects to the main ring through the BMENOS0300 network option switch module.
10 An STB island in a DIO sub-ring connects to the main ring through the BMENOS0300 module.
11 Another STB island in the same DIO sub-ring includes the STB module that is host **B** in the device network.

In this example, the IP forwarding service of the BMENOC0321 module has three interfaces with different IP addresses in three subnetworks:

Network	IP Forwarding Service			
	IP Address	Sub-Network Mask	Network Address	Ethernet Interface
control network	172.168.30.1	255.255.0.0	172.168.0.0	ETH 2, ETH 3
device network	192.168.13.1	255.255.0.0	192.168.0.0	Ethernet backplane port
extended device network	10.20.1.1	255.255.0.0	10.20.0.0	ETH 1

Now that you have established the IP forwarding service, add the IP address forwarding information to the PC (host **A**) and the STB module (host **B**), which allows the hosts to send packets beyond their own subnetworks by utilizing the IP forwarding service of the BMENOC0321 module.

Configure the STB module to forward all traffic that is destined for outside its subnetwork to the BMENOC0321 module. That is, confirm that all traffic for networks other than 192.168.0.0 is forwarded to the appropriate interface of the BMENOC0321 module.

In this example, the appropriate interface of the BMENOC0321 module is its device network interface, which is at IP address of 192.168.13.1 in the same network as the STB module. This configuration is accomplished by setting the default gateway address of the STB module to be 192.168.13.1.

Configure the PC in a similar way. However, in a PC environment, it is possible to configure distinct rules about communications. To facilitate communications between the example PC in the control network and the devices in the device network, set the IP address of the BMENOC0321 module in the control network as the route for traffic that is destined for the device network.

Set a Static Route

The PC (host **A**) resides in the control network and is able to communicate with the BMENOC0321 module in the local rack via the module's control network IP address. For the PC to communicate with devices in the device network, add a static route to the PC, as in this example: `c:\route ADD 192.168.0.0 mask 255.255.0.0 172.16.30.1`

Where:

- `192.168.0.0` is the device network.
- `172.16.30.1` is the IP address of the BMENOC0321 module in the control network.

Use the optional `"-p"` option to create a persistent route across system boots.

With this configuration, the PC sends all traffic destined for the device network (192.168.0.0) to the BMENOC0321 module (at IP address 172.16.30.1). The BMENOC0321 module then forwards the traffic to the appropriate device (and vice versa).

Connecting a Control Network to an M580 System

Introduction

A BMENOC0321 control network module provides multiple network connectivity options, while preserving network determinism:

- *non-redundant (single attachment)*:
  - Use a single connection from the *control network port* on the BMENOC0321 module on the local rack to an Ethernet port on a switch on the control network
  - This connection does not provide redundancy.
  - Use copper shielded twisted 4-pair CAT6 (10/100/1000 Mbps) cable to connect the BMENOC0321 module to the switch on the control network. Confirm that the distance to the switch is less than or equal to 100 m.
- *redundant* (*RSTP*):
  - Implement cable redundancy in a daisy chain loop topology from the *control network port* on the BMENOC0321 module to a port on an Ethernet managed Ethernet switch on the control network. This DRS is linked to a second DRS, which completes the daisy chain loop by connecting back to the BMENOC0321 module.
  - Use copper shielded twisted 4-pair CAT6 (10/100/1000 Mbps) cable between the BMENOC0321 module and the two Ethernet switches and between the DRSs as well. Confirm that the distance to the switch and between the switches is less than or equal to 100 m.

NOTE: The switch used in a non-redundant control network type does not have to be a managed dual-ring switch (DRS).

NOTE:
To connect a BMENOC0321 module to a control network in a Modicon M580 system, Schneider Electric recommends the use of these cables:

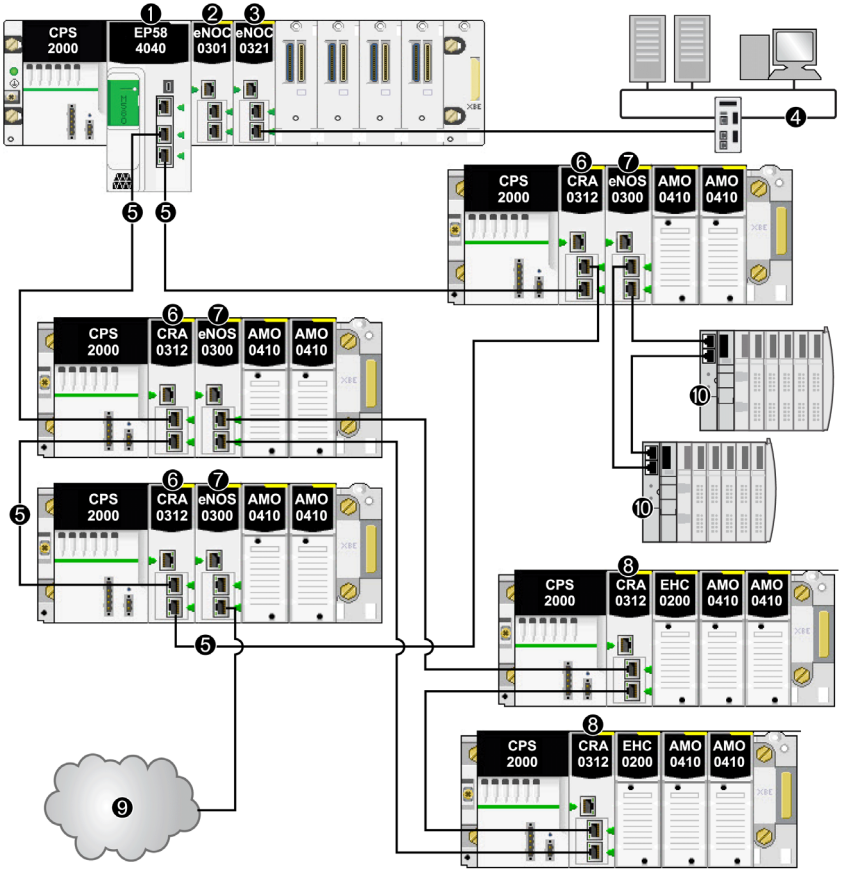
- *10/100 Mbps*: For a communications link less than or equal to 100 Mbps, use CAT5e or CAT6 copper shielded twisted four-pair cables.
- *1000 Mbps*: For a communications link less than or equal to 1000 Mbps, use only CAT6 copper shielded twisted four-pair cables.

Connecting a Non-redundant Control Network

For control networks that do not require redundancy, you can provide network transparency between the control network and your desired network(s):

Step	Action
1	Configure these modules on the local rack: • M580 CPU • BMENOC0321 control network module • BMENOC0301/BMENOC0311 Ethernet communication module(s) **NOTE:** The number of BMENOC0301/BMENOC0311 Ethernet communication modules depends upon your specific network design. You have the option to use these modules to extend the DIO scanning capabilities beyond the capacity of the selected M580 CPU.
2	Confirm that the Ethernet backplane ports are enabled.
3	Install an Ethernet switch on the control network a distance equal to or less than 100 m from the BMENOC0321 module on the local rack. **NOTE:** The switch does not have to be a managed dual-ring switch (DRS).
4	Connect the control network port of the BMENOC0321 module (ETH 2 or ETH 3) to an Ethernet port on the switch on the control network. **NOTE:** Refer to the Schneider Electric recommendations for cable types *(see page 33)*.

This sample architecture shows an Ethernet RIO network that is connected to a control network. The BMENOC0321 control network module on the local rack is connected to a control network to provide network transparency between the RIO network and the control network:



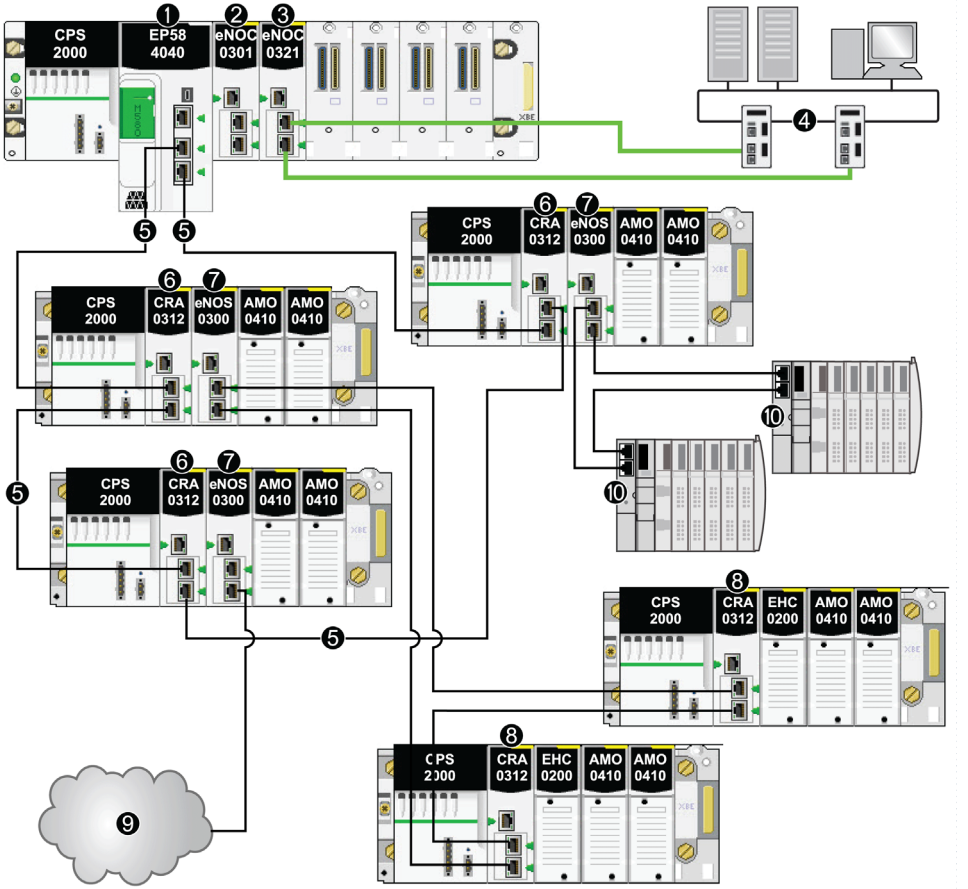
1 BME•58•••• CPU connecting the local rack to the main ring
2 BMENOC0301/BMENOC0311 Ethernet communication module on the local rack managing a number of distributed equipment nodes on the device network beyond the DIO scanning capacity of the M580 CPU
3 BMENOC0321 control network module
4 control network
5 RIO main ring
6 BM•CRA312•0 (e)X80 EIO adapter module on an RIO drop on the main ring
7 BMENOS0300 network option switch module on an RIO drop
8 BM•CRA312•0 module on an RIO drop on an RIO sub-ring
9 DIO cloud connected to the main ring via a BMENOS0300 network option switch module on an RIO drop
10 STB island in a DIO sub-ring connected to the main ring via a BMENOS0300 module on an RIO drop

Connecting a Redundant Control Network

If your control network requires redundant links, provide network transparency between the control network and your desired network(s):

Step	Action
1	Configure these modules on the local rack: ● M580 CPU ● BMENOC0321 control network module ● BMENOC0301/BMENOC0311 Ethernet communication module(s) **NOTE:** The number of BMENOC0301/BMENOC0311 Ethernet communication modules depends upon your specific network design. You have the option to use these modules to extend the DIO scanning capabilities beyond the capacity of the selected M580 CPU.
2	Confirm that the Ethernet backplane ports are enabled.
3	Install and connect two managed Ethernet switches via copper shielded twisted 4-pair CAT5e (10/100 Mbps) cable on the control network a distance equal to or less than 100 m from each other and from the BMENOC0321 module on the local rack.
4	Use copper shielded twisted 4-pair CAT5e (10/100 Mbps) cables to make these connections: ● Connect a control network port on the BMENOC0321 module (ETH 2 or ETH 3) to a port on the DRS (optional). ● Connect the other control network port on the BMENOC0321 module (ETH 2 or ETH 3) to a different RSTP port (in the same domain) in the DRS.

This graphic shows an Ethernet RIO network that is connected to a control network with redundant links. The BMENOC0321 control network module on the local rack is connected to two separate DRSs on the control network to provide redundancy and network transparency between the RIO network and the control network:



1 BME•58•••• CPU connecting the local rack to the main ring
2 BMENOC0301/BMENOC0311 Ethernet communication module on the local rack managing distributed equipment on the device network
3 BMENOC0321 control network module (connected to control network via RSTP protocol in Gb dual ports)
4 control network
5 RIO main ring
6 BM•CRA312•0 (e)X80 EIO adapter module on an RIO drop on the main ring
7 BMENOS0300 network option switch module on an RIO drop on the main ring
8 BM•CRA312•0 module on an RIO drop on an RIO sub-ring
9 DIO cloud connected to the main ring via a BMENOS0300 network option switch module on an RIO drop
10 STB island in a DIO sub-ring connected to the main ring via a BMENOS0300 module on an RIO drop

Chapter 4
Creating a Control Expert Project

What Is in This Chapter?

This chapter contains the following topics:

New Modicon M580 Project

Introduction

Use these steps to create a new Modicon M580 Control Expert project and add these components to the **PLC bus**:

- CPU
- power supply
- BMENOC0321 module

NOTE: If you already have a Control Expert project with an installed a power supply and a CPU, skip to the procedure for adding a BMENOC0321 module (below).

Create a Project

Create and save a new Control Expert project:

Step	Action
1	Open Control Expert.
2	Open the **New Project** window in the menu (**File → New...**).
3	Expand (+) the Modicon M580 menu.
4	In the **PLC** list, select the Modicon M580 PAC (PLC) for your project.
5	In the **Rack** list, select the Modicon M580 rack for your project.
6	Press **OK**. **NOTE:** Control Expert processes your request and opens the **Project Browser**.

Schneider Electric recommends that you periodically save changes to the project:

Step	Action
1	Open the **Save As** dialog (**File → Save**).
2	Enter a **File name** for the new Control Expert project.
3	Click **Save** to save your project to the path indicated in the **Save in** field.

You can change the location to which you save your project file:

Step	Action
1	Open the **Options Management** window (**Tools → Options...**).
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 these other paths: ● Import/Export file path ● XVM path ● Project settings templates path
4	Click **OK** to confirm your path selections and close the window.

View the Hardware Rack

Use these steps to see a graphical view of the Modicon M580 rack:

Step	Action
1	Expand (+) the **Project Browser** to see the **PLC bus** (**Project → Configuration → PLC bus**).
2	Double-click the **PLC bus** to see the M580 rack and open the **Hardware Catalog**. **NOTE:** The rack contains a power supply module and the CPU that you selected previously.
3	Save the project (**File → Save**).

Add the BMENOC0321 Module

Add a BMENOC0321 Ethernet communications module to the Control Expert project:

Step	Action
1	View the available communication modules (**Hardware Catalog → Modicon M580 local drop → Communication**).
2	Drag the BMENOC0321 Ethernet communications module to an open slot in the rack to see the **New Device** window.
3	Note the topological address for the module in the **New Device** window and press **OK** to see the **General** tab of the **Properties of device** window. **NOTE:** The **General** tab in the **Properties of device** is the only tab that contains configurable information. The other tabs contain read-only information.

Step	Action
4	Note the provided **Alias name** for the module and press **OK**. You can use this field to configure a different **Alias name**: ● When you change the **Alias name**, Control Expert changes the base input and output type and variable names to match the new **Alias name**. ● Assign a unique **Alias name** to each communication module to distinguish between modules of the same type. ● The **Alias name** is used elsewhere in Control Expert: ❍ It is the **Network name** when you view the module properties. ❍ It is the module name in the **DTM Browser** under the **Host PC**.
5	Confirm that the **PLC bus** displays the BMENOC0321 and save the project (**File → Save**).

Communication Module and Remote Device Node Commands

In the Control Expert **PLC bus**, right-click on the BMENOC0321 module to access these commands:

Name	Description	
Cut[1]	Cut the selected module to the clipboard.	**NOTE:** These commands do not apply to the BMENOC0321 module.
Copy[1]	Copy the selected module to the clipboard.	
Paste[1]	Paste the module on the clipboard to a selected rack slot.	
Export	This allows you to export the module configuration and all devices configured behind the NOC master DTM *(see page 56)*. **NOTE:** This function is disabled if the PLC is connected to Control Expert.	
Delete Module[1]	● Delete the selected module from the rack. ● Delete the selected module from the **DTM Browser**. ● Delete the corresponding DTM and its sub-node DTMs from the DTM connectivity tree.	
Open Module[1]	See a description of the selected communications module.	
Move Module[1]	Move the selected module to the rack slot that you designate.	
Power Supply and IO Budget[2]	View these tabs: ● **Power supply**: power consumption for the module ● **I/O**: number of networks used by module	

1. This command also appears in the **Edit** menu.
2. This command also appears in the **Services** menu.

Power Supply and IO Budget

Open the **Power Supply and IO Budget** window to monitor the budget for the application-specific channels for each module on the local rack. A bar chart indicates the state of the budget according to this color scheme:

Color	Description
green	This is the number of configured channels.
white	This is the number of available channels.
red	This is the number of channels that are not managed by the BMENOC0321 module. (In this case, a message reports the excess of unmanaged channels.)

Open the **Power Supply and IO Budget** window to update the budget for modules that are added or removed.

NOTE: Close the **Power Supply and IO Budget** window to delete or add a module.

Use the **Power Supply and IO Budget** tabs to monitor the module:

- **Power supply**: This tab shows the power discharged in the module for each voltage it uses as well as the total power.
- **I/O**: This tab shows the number of application-specific channels configured in the module.

Export the BMENOC0321 Module Configuration

At a Glance

You can access this function from the Control Expert PLC bus configuration window. The function allows you to export the BMENOC0321 module configuration and all the devices configured behind the NOC master DTM.

The **entire configuration** is copied to a `.ZHW` file.

Exporting

To export the module configuration, perform the following operations:

Step	Action
1	Expand (+) the **Project Browser** to see the **PLC bus** (**Project** → **Configuration** → **PLC bus**).
2	Double-click the **PLC bus** to see the M580 rack.
3	Select the BMENOC0321 module from which you want to export the configuration. Activate the **Export** command from the context-sensitive menu (accessible with a right-click). **Result**: A dialog box appears on the screen.
4	Select the destination directory for the export (directory tree).
5	Enter the file name.
6	Select the **Export** button. **Result**: A progress indicator lets you know how the export is proceeding.
7	A message in the **output window** tells you that export is complete.

Import a BMENOC0321 or a BMENOC0301/11 Module Configuration

At a Glance

You can access this function from an empty slot of the Control Expert PLC bus configuration window. The function allows you to import a BMENOC0321 or a BMENOC0301/11 module configuration and all the devices configured behind the NOC master DTM.

File type to import: `.ZHW`

Restrictions

You cannot import twice (or more) the same exported file (.ZHW) in the same application.

Before importing again an exported file (.ZHW), use the **M580ApplicationUpdate.exe** tool to make a file conversion. This tool is located in the same program directory as Control Expert software.

The **Import** command is enabled if:

- the PLC is not connected to Control Expert.
- you select an empty slot of the M580 BMEXBP*xxxx* main rack.

Importing

To import a module configuration, perform the following operations:

Step	Action
1	Expand (+) the **Project Browser** to see the **PLC bus** (**Project** → **Configuration** → **PLC bus**).
2	Double-click the **PLC bus** to see the M580 rack.
3	Select an empty slot of the M580 main rack. Select the **Import** command from the context-sensitive menu (accessible with a right-click). **Result**: A dialog box appears on the screen.
4	Choose the source directory for the import (directory tree).
5	Select the file to import. A dedicated tooltip indicates the type of content for the `.ZHW` file. **Result**: The name of the file appears in the **File name** field.
6	Select the **Import** button. **Result**: A progress indicator lets you know how the export is proceeding.
7	A message tells you that the import is complete.

Helping Secure a Project in Control Expert

Creating an Application Password

In Control Expert, create a password to help protect your application from unwanted modifications. The password is encrypted and stored in the PAC. Any time the application is modified, the password is required.

Step	Action
1	In the **Project Browser** window, right-click **Project → Properties**.
2	In the **Properties of Project** window, click the **Protection** tab.
3	In the **Application** field, click **Change password**.
4	In the **Modify Password** window, enter a password in the **Entry** and **Confirmation** fields.
5	Click **OK**.
6	In the **Application** field, select the **Auto-lock** check box if you want to require the password to resume the application display. You may also click the up/down arrows to set the number of minutes at which time the application would auto-lock.
7	To save the changes: ● Click **Apply** to leave the **Properties of Project** window open. – or – ● Click **OK** to close the window.
8	Click **File → Save** to save your application.
9	If you wish to change the password at a later time, follow the preceding steps.

NOTE:

- To help ensure cyber security, confirm that you change the password with modules that have firmware V1.05 or later.
- You cannot reset the module to factory settings if you lose the password.

More information about application password is given in Application Protection *(see EcoStruxure™ Control Expert, Operating Modes)* page.

NOTE: When exporting a project to a .XEF or a .ZEF file, the application password is cleared.

Using Memory Protect

In Control Expert, select the **Memory Protect** option to help protect your application from unwanted modifications.

Step	Action
1	In the **Project Browser** window, expand the **Configuration** folder to display the CPU.
2	To open the CPU configuration window: • Double-click the CPU. – or – • Right-click **BME P58 •0•0 → Open**.
3	In the CPU window, click the **Configuration** tab.
4	Select the **Memory protect** check box, and enter an input address of your choice.
5	Click **File → Save** to save your application.

Chapter 5
Configuring the BMENOC0321 Module

Introduction

This chapter shows you how to use Control Expert programming software to select and configure the BMENOC0321 Ethernet communications module on the local rack.

NOTE: The device configuration procedure is valid when configuring a project with Control Expert Classic. When you configure your device from a system project, some commands are disabled in Control Expert editor. In this case, you need to configure these parameters at system level by using the Topology Manager.

What Is in This Chapter?

This chapter contains the following sections:

Section 5.1
Configuration with the Control Expert DTM

Introduction

Use the instruction in this section to configure an Ethernet communications module with the Control Expert DTM.

What Is in This Section?

This section contains the following topics:

About the Control Expert DTM Browser

Introduction to FDT/DTM

Control Expert incorporates the Field Device Tool (FDT) / Device Type Manager (DTM) approach to integrate distributed devices with your process control application. Control Expert includes an FDT container that interfaces with the DTMs of EtherNet/IP and Modbus TCP devices.

An EtherNet/IP device or Modbus TCP device is defined by a collection of properties in its DTM. For each device in your configuration, add the corresponding DTM to the Control Expert **DTM Browser**. From the **DTM Browser** you can open the device's properties and configure the parameters presented by the DTM.

Device manufacturers may provide a DTM for each of its EtherNet/IP or Modbus TCP devices. However, if you use an EtherNet/IP or Modbus TCP device that has no DTM, configure the device with one of these methods:

- Configure a generic DTM that is provided in Control Expert.
- Import the EDS file for the device. Control Expert populates the DTM parameters based on the content of the imported EDS file.

NOTE: The DTM for a BMENOC0321 module is automatically added to the **DTM Browser** when the module is added to the **PLC bus**.

Automatic DTM Creation

In a Unity Pro 11.0 or later application, DTMs for some Ethernet communication modules and other pre-configured devices (see the following list) are created automatically when added to an Ethernet rack on the main local or main remote drops. A default DTM name is assigned in the DTM topology, but you may modify the name:

- Right-click the desired DTM name in the **DTM Browser** and select **Properties**.
- Click the **General** tab, and edit the DTM name in the **Alias name** field.
- Click Apply to save the changes and leave the window open.
  – or –
  Click **OK** to save the changes and close the window.
  NOTE: The **OK** button is valid to press only when Control Expert has confirmed that the DTM is unique.

DTMs are automatically created when you add a BMENOC0321 Ethernet communication modules to an Ethernet rack.

Windows Compatibility

M580 DTMs are compatible with the following operating systems:

- Microsoft Windows 7® 32/64 bits Professional edition
- Microsoft Windows 8
- Microsoft Windows Server 2008

NOTE: Unity Pro 10.0 no longer supports Microsoft Windows XP.

The following table describes the minimum and recommended PC configuration to run M580 DTMs inside Control Expert:

Parameter	Description
processor	minimum: Pentium 2.4 GHz or later recommended: 3.0 GHz
RAM memory	minimum: 2 GB recommended: 3 GB **NOTE:** Use a PC with 4 GB of RAM memory if more than 20 DTMs are used in your application. **NOTE**: For applications using FDT / DTM: • minimum: 2 GB • recommended: 4 GB
hard disk	minimum: 8 GB free space recommended: 20 GB free space
operating system	Microsoft Windows 7® 32/64 bits Professional edition or later
drive	minimum: DVD drive recommended: DVD writer drive
display	minimum: VGA (800 x 600) recommended: SVGA (1024 x 768) or later with high color 24 bits
peripherals	Microsoft mouse or compatible pointing device
web access	Web registration requires Microsoft Internet Explorer V8 or later.
other	USB port on the PC

Open the DTM Browser

View the configuration options for the BMENOC0321 Ethernet communications module in the Control Expert **DTM Browser**:

Step	Action
1	Open a Control Expert project that includes a BMENOC0321 module.
2	Open the Control Expert **DTM Browser** (**Tools → DTM Browser**).
3	In the **DTM Browser**, find the name that you assigned to the BMENOC0321 module *(see page 53)*.
4	Double-click the name of the BMENOC0321 to open the configuration window.
5	View the DTM configuration parameters for the Ethernet communications module in the open dialog: ● Channel Properties *(see page 84)* ● Services *(see page 93)* ● Security *(see page 118)* ● EtherNet/IP Local Slaves *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* ● Device List *(see page 136)* ● Logging *(see page 144)*

DTM Types

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

There are two kinds of DTMs:

- *master (communication) DTMs*: This DTM is both a device DTM and a communication DTM. The master DTM is a pre-installed component of Control Expert.
- *generic DTMs*: The Control Expert FDT container is the integration interface for any device's communication DTM.

This list contains these node types:

DTM Type	Description
communication (master)	Communication DTMs appear under the root node (host PC). A communication DTM can support gateway DTMs or device DTMs as children if their protocols are compatible.
gateway	A gateway DTM supports other gateway DTMs or device DTMs as children if their protocols are compatible.
device	A device DTM does not support any child DTMs.

Node Names

Each DTM node has a default name when it is inserted into the browser. The default name for gateway and device DTMs is in the format *<protocol:address> device name*. (For example, < EtherNet IP:192.168.20.3 > BMENOC0321.)

This table describes the components of the default node name:

Element	Description
channel	This is the name of the channel communication medium into which the device is plugged. This name is read from the DTM and is set by the device vendor. **Example**: EtherNet/IP, Modbus
address	This is the bus address of the device that defines the connection point on its parent gateway network (for example, the device IP address).
device name	The default name is determined by the vendor in the device DTM, but the user can edit the name.

Node Status

The **DTM Browser** contains graphics to indicate the status of each DTM node in the connectivity tree:

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

Command	Description
Delete	Remove the selected node (and its sub-nodes) from the **DTM Browser**.
Properties	Open the **Properties of ...** dialog box to identify the name of the missing DTM.

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

DTM Browser Menu Commands

Introduction

The Control Expert **DTM Browser** includes these commands for the selected DTM associated with a module:

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

Host PC Node Commands

Right-click **Host PC** to access these commands in the Control Expert **DTM Browser**:

Name	Description
Add...[1]	Open the **Add** window (a subset of the **Hardware Catalog**). Select a device DTM to add to the **DTM Browser**.
Check DTM devices[1]	Check the current project for invalid DTMs or DTMs that are not installed on the PC. If the results of the check include invalid or not-installed DTMs, they appear in the **User errors** tab in the information window and a red **X** is superimposed over their icons in the **DTM Browser**.
DTM services	Display the communication DTMs and the device topology along with their respective IP addresses and connection states. For each device, you can connect, disconnect, load data from devices, or store data to devices. You can also choose to stop communications or continue an activity when errors are detected.
DTM hardware catalog	Display the **DTM catalog** tab in the **Hardware Catalog**.
Expand all[2]	Display and expand every DTM in the project in the **DTM Browser**.
Collapse all[2]	Display only the communication DTMs in the project.
1. This command also appears in the **Edit** menu. 2. This command also appears in the **View** menu.	

Communication Module and Device Commands

Right-click the desired module or device in the **DTM Browser** and scroll to these commands:

Name	Description
Open[1]	View the configuration options for the selected module or device. **NOTE:** You can also double-click the DTM in the **DTM Browser** to open this window.
Add[1]	Open the **Add** dialog box to view a subset of available DTMs in the **Hardware Catalog**. **NOTE:** Control Expert filters the content of the **Add** dialog to display 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.
Field Bus Discovery	This scans the connected physical devices to create the corresponding field bus topology. Refer to the *Field Bus Discovery Service topic (see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.
Sort by Address	Sort the DTMs according to their IP addresses.
Connect[1]	This connects the DTM to its physical device on the network. This connection does not depend on the PAC 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 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	Copy the selected device DTM.
Paste	Paste the selected device DTM.
Go to module or device	Use this feature to delete a pre-configured module DTM: • Right-click the desired DTM node. • Select **Go to module or device**. • Right-click the module, and select **Delete**. **NOTE:** You cannot use this feature if you manually open the window that displays the module/device you wish to delete.
Device menu	This command opens a sub-menu that contains device-specific commands, as determined by the device vendor.
Properties[1]	Open the Ethernet communications module's **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 in[2]	Make this selection to display only the selected module in the connectivity tree of the **DTM Browser**.
Zoom out[2]	This returns to the display of the entire DTM connectivity tree.
Expand all[2]	Display the DTMs below the selected DTM.
Collapse all[2]	Display 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 displays that contains these commands:

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	Open 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.
	Export EDS library	Opens the **Export EDS library** wizard, which you can use to archive EDS device library.
	Import EDS library	Opens the **Import EDS library** wizard, which you can use to import archived EDS 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** window, which you can use to send explicit messages to EtherNet/IP remote devices.
	Modbus TCP Explicit Message	Opens the **Modbus TCP Explicit Message** window, which you can use to send explicit messages to Modbus TCP remote devices.
	Store Device Conf to FDR	Transfers the configuration settings of the device DTMs to the FDR server, either online or offline.
	About	
	Advanced Mode	Displays or hides expert-level properties that help define Ethernet connections.

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 this icon:

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

Enable and disable **Advanced Mode**:

Step	Action
1	Close configuration windows associated with the Ethernet communication module.
2	In the **DTM Browser**, right-click the Ethernet communication module.
3	Scroll to **Additional functions** (**Device menu → Additional functions**) to see the status of the **Advanced Mode**: ● *checked*: The **Advanced Mode** is enabled. ● *unchecked*: The **Advanced Mode** is disabled. **NOTE:** If any configuration or properties windows that are associated with the device or module are open, the **Advanced Mode** is not available (grayed out).
4	Select **Advanced Mode** to toggle its status. For example, if **Advanced Mode** is checked (enabled), select it to disable it.

In **Advanced Mode** you can configure these items:

- EtherNet/IP features *(see page 116)* (timeout parameters and DIO scanner behavior)
- RSTP parameters *(see page 102)* (bridge parameters and port parameters
- Online Action *(see page 216)* (refresh data and reset devices)

Managing DTM Connections

Introduction

Use these instructions to connect or disconnect a device of module DTM to or from a physical device or module.

Connecting and Disconnecting

Connect or disconnect a DTM and the associated device or module through the contextual pop-up menu in the Control Expert **DTM Browser**:

Step	Action
1	In the Control Expert **DTM Browser**, locate the DTM that you want to connect to or disconnect from.
2	Click the right mouse button to view a pop-up menu.
3	Select **Connect** or **Disconnect** from the pull-down menu (or access the **Connect** and **Disconnect** commands in the Control Expert **Edit** menu): ● **Connect**: Perform these tasks with a connection: ❍ Configure Ethernet communication modules, distributed devices, and their common Ethernet connections. ❍ Monitor and diagnose the real-time operation of the device or module. ● **Disconnect**: Perform these tasks without a connection: ❍ Configure an Ethernet communication module or distributed device by editing its properties. ❍ A disconnected DTM appears in normal text (not **bold**). (The **Connect** command is available only for disconnected DTMs.)

The **DTM Browser** indicates the relationship between the DTM and the remote module or device:

- A connected DTM appears in **bold** text. (The **Disconnect** command is available only for connected DTMs.)
- A disconnected DTM appears in regular (not **bold**) text. The **Connect** command is available only for disconnected DTMs.

To connect to BMENOC0321, set the **Source IP Address** in the channel properties configuration *(see page 84)* to the same network as the communications module.

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 network. 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.

NOTE: To use the field bus discovery service, connect the workstation directly to the device network. If you connect to the device network through a BMENOC0321 control network module, the IP Forwarding service blocks the broadcast messages that are required to discover the network devices.

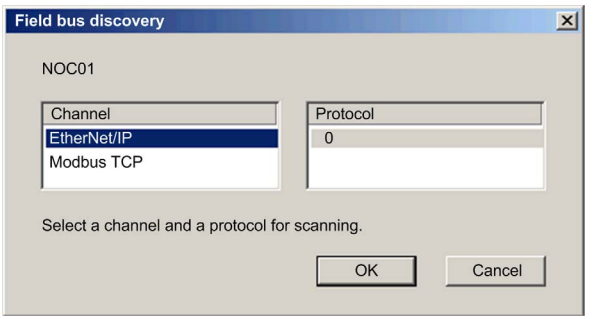
Performing Field Bus Discovery

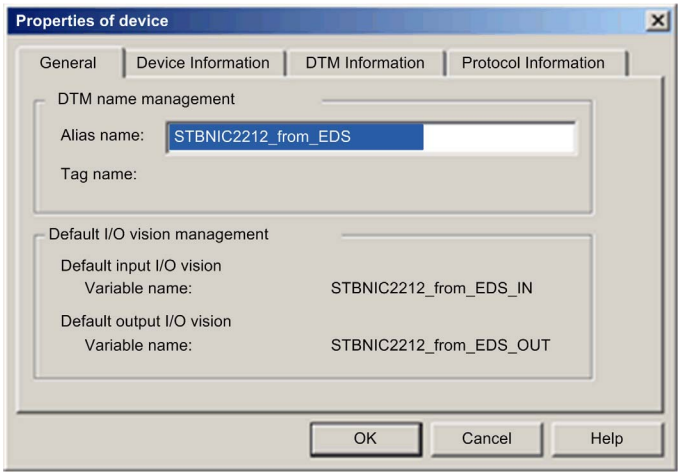
The results of the scanning process is compared to the registered DTMs in the DTM catalog of the computer. If a match is found in the DTM catalog for a scanned device, the results are accompanied with a matching type that gives the accuracy of the match.

These are the available matching types:

- *Exact match*: Every identification attribute matches. The correct device type was found.
- *Generic match*: At least the **Vendor** and device **Type ID** attributes match. The support level of the DTM is “Generic Support.”
- *Uncertain match*: At least the **Vendor** and device **Type ID** attributes match. The support level of the DTM is *not* “Generic Support.”

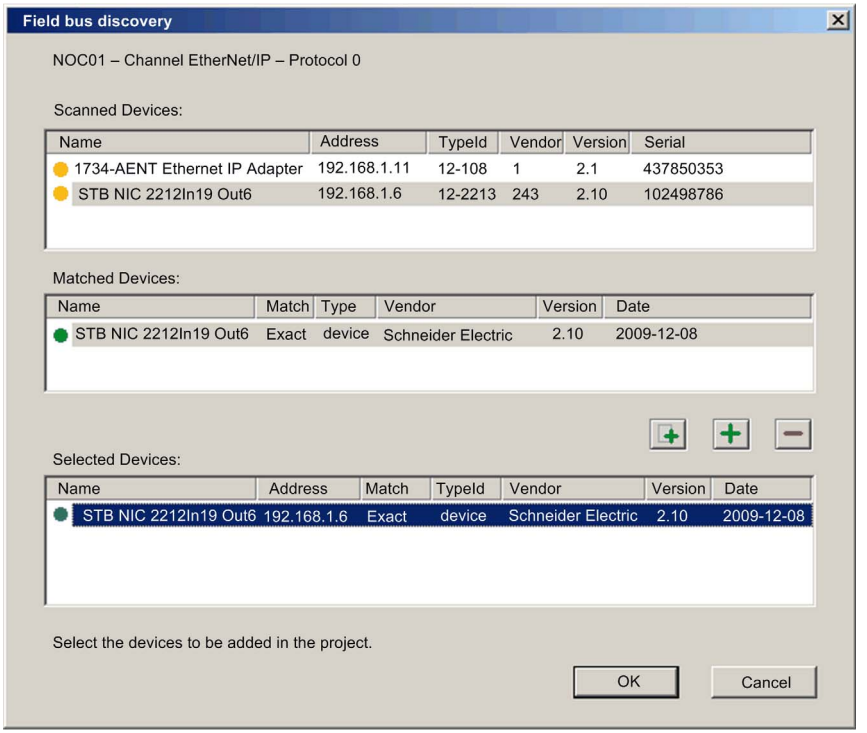
Use the field bus discovery service:

Step	Action
1	In the **DTM Browser**, select an appropriate DTM. **NOTE:** The field bus discovery service limits its search to the range of IP addresses that is pre-configured for the selected channel in the **Channel Properties** page *(see page 84)*.
2	Right-click the DTM and scroll to **Field bus discovery** to open the dialog box: 

Step	Action
3	Under these conditions, select a channel and a protocol: ● The DTM has more than one channel. ● The channel supports more than one protocol.
4	Click on **OK**. The service starts to detect devices on the selected channel.
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 the devices you want to add in the **Field Bus Discovery** dialog, click **OK**.
8	If the field bus discovery process finds at least one device with an IP address that is already used in the project, you are asked if you want to continue and replace the existing project device(s): ● **Yes**: Proceed to the next step. ● **No**: Cancel automatic field bus discovery.
9	The device properties dialog (below) 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**. The dialog closes, then re-opens if there is another device to be added to the application.
10	Repeat the above step for each additional discovered device.
11	After you finish adding devices to the application, configure each device for operation as part of the application: ● 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 these lists:

List	Description
Scanned Devices	The devices (matched and unmatched) found during the scan.
Matched Devices	The matched DTMs found in the workstation DTM catalog 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:

Color	Meaning
Green	The device has been selected.
Yellow	The device has been matched.
Red	The device has **not** been matched.
Black	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
NOTE: An icon can consist of two 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	

This dialog has five buttons:

Button	Use this button to...
Add All 	Automatically add the most closely 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**, 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. Information in the three lists is discarded.

Configuring DTM Properties

Introduction

You can edit and view parameters in the **Device List** that is associated with the M580 DTM.

Open the Device List

View the **Device List**:

Step	Action
1	Open the **DTM Browser** in Control Expert (**Tools → DTM Browser**).
2	Double-click the M580 DTM in the **DTM Browser**.
3	In the configuration tree associated with the M580 DTM, click **Device List**.

Configuring Properties

Configure the **Device Editor** properties:

Step	Action	
1	While you edit a parameter, Control Expert displays an icon next to the field you are editing and in the navigation tree. These icons refer to value of the parameter that is being edited:	
2		The entered value is not valid. The **Apply** button does not work until a valid value is entered.
		This parameter has changed. The **Apply** button does not work until the value is corrected.
3	Click one of these buttons: • **Apply**: Save your changes and keep the page open. • **OK**: Save your changes and close the page. • **Cancel**: Cancel changes. **NOTE:** Your changes do 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.	

Uploading and Downloading DTM-Based Applications

Introduction

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

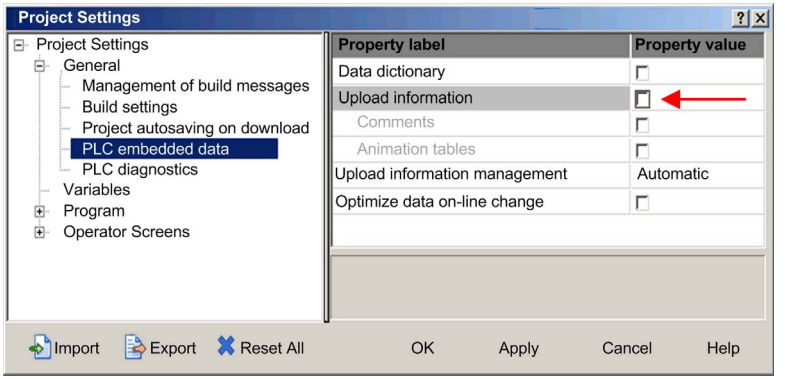
To perform a successful upload, confirm that the application file includes specific upload-related information as part of the application.

Downloading DTM-Based Applications

Control Expert applications that include DTM files require more memory than traditional Control Expert applications. 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 displays a message during the build process, before the application is downloaded to the PAC.

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, change the Control Expert configuration:

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 the **Upload information** check box: 
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 PAC.

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

Uploading DTM-Based Applications

DTM-based applications that were successfully downloaded to the CPU (with the project's **Upload information** setting enabled) can later be uploaded from the PAC to the PC if the target PC has these files installed:

- a version of Control Expert that is equal to or later than the version used to create the application
- the DTMs for the modules included in the configuration
- the device DTMs for the DTM-based devices attached to the network (confirm that the DTMs are of the same or later revision as each device DTM used in the configuration)
- the device EDS files for any EtherNet/IP device used in the configuration (confirm that the EDS files are of the same or later revision as each device EDS file used in the configuration)

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

NOTE: Confirm that each of the above DTM components is installed on the target PC *before* attempting the upload.

Input and Output Items

Introduction

Create input and output items to support peer-to-peer data transfers between and among scanners. Use the Control Expert DTM to create input and output items and to define the name and data type of each item.

NOTE: The BMENOC0321 control network module performs the function of a network scanner. However, you can enable its local slaves *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* to make the BMENOC0321 perform the role of an EtherNet/IP adapter. In that case, network EtherNet/IP scanners can read from and write to CPU data through the enable local slaves.

Create input and output items in these groups:

- one or more single bits
- 8-bit bytes
- 16-bit words
- 32-bit dwords
- 32-bit IEEE floating values

The number of items you create depends upon the data type and size of each item.

Accessing Items

View the **Items** configuration tabs:

Step	Action
1	Open an M580 project in Control Expert.
2	Open the **DTM Browser** (**Tools → DTM Browser**).
3	In the **DTM Browser**, double-click the DTM that corresponds to the Ethernet communication module.
4	*device connections*: Expand **Device List** and select **Items** for the appropriate connection. *local slaves*: Expand **EtherNet/IP Local Slaves** and select **Items** for the appropriate local slave.

Creating Input Items

Follow these steps to create sample input items:

Step	Action
1	Select the **Input** tab.
2	In the **Default Item Name Root** field, enter a context-sensitive name.
3	Select the first two table rows (0 and 1).
4	Click **Define Item(s)** to open the **Item Name Definition** dialog box.

Step	Action
5	In the **New Item(s) Data Type** field, scroll to **Word** for this example. **NOTE:** The number of selected rows conforms to the data type: • **Byte**: Select a single row. • **WORD**: Select two rows beginning with the next available whole word.
6	Click **OK** to see the new item on the **Input** tab.
7	Click **Apply** to save the new items and leave the page open.
8	Repeat these steps to create additional input items that consume the next available row(s) in the table.
9	Save your changes (**File → Save**).

Creating Input Bit Items

Follow these steps to create sample input bit items:

Step	Action
1	Select the **Input (bit)** tab.
2	In the **Default Item Name Root** field, enter a context-sensitive name to monitor the device status.
3	Press the **Define Items** button.
4	Enter a name in the **Item Name** (or accept the default name).
5	Click **OK** to see the new bit item on the **Input** tab.
6	Click **Apply** to save the new items and leave the page open.
7	Repeat these steps to create additional input bit items.
8	Save your changes (**File → Save**).

Creating Output Items

Follow these steps to create sample output items:

Step	Action
1	Select the **Output** tab.
2	In the **Default Item Name Root** field, enter a context-sensitive name.
3	Select the first two table rows (0 and 1). **NOTE:** The number of selected rows conforms to the data type: • **Byte**: Select a single row. • **WORD**: Select two rows beginning with the next available whole word.
4	Click **Define Item(s)** to open the **Item Name Definition** dialog box.
5	In the **New Item(s) Data Type** field, scroll to **Word** for this example.
6	Click **OK** to see the new item on the **Output** tab.

Step	Action
7	Click **OK** to close the **Items** window.
8	Save your changes (**File → Save**).

Creating Output Bit Items

Follow these steps to create sample output bit items:

Step	Action
1	Select the **Output (bit)** tab.
2	In the **Default Item Name Root** field, enter a context-sensitive name to monitor the device status.
3	Press the **Define Items** button.
4	Enter a name in the **Item Name** (or accept the default name).
5	Click **OK** to see the new bit item on the **Input** tab.
6	Click **Apply** to save the new items and leave the page open.
7	Repeat these steps to create additional input bit items.
8	Click **OK** to save the new items and close the page.

Section 5.2
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:

Accessing Channel Properties

Introduction

On the Control Expert **Channel Properties** page, you can select a **Source IP Address** (PC) from a pull-down menu.

The **Source IP Address** (PC) menu is a list of IP addresses that are configured for a PC that has the Control Expert DTM installed.

To make the connection, choose a **Source IP Address** (PC) that is in the same network as the BMENOC0321 module.

You can execute these tasks through this connection:

- Perform fieldbus discovery.
- Execute Online Actions.
- Send an explicit message to an EtherNet/IP device.
- Send an explicit message to a Modbus TCP device.
- Diagnose modules.

NOTE: Refer to the *Modicon M580 Standalone System Planning Guide for Frequently Used Architectures* to establish transparency between a USB connection and a device network.

Open the Page

View the **Channel Properties** for the Ethernet communications module:

Step	Action
1	Open a Control Expert project that includes a BMENOC0321 module *(see page 52)*.
2	Open the **DTM Browser** (**Tools → DTM Browser**).
3	In the **DTM Browser**, find the name that you assigned to the BMENOC0321 module *(see page 53)*.
4	Double-click (or right-click **Open**) the name of the BMENOC0321 to open the configuration window. **NOTE:** You can also right-click the module and scroll to **Open** to view the configuration window.
5	Select **Channel Properties** in the navigation pane.

Property Descriptions

Select **Channel Properties** in the navigation tree to configure these properties:

Field	Parameter	Description
Source Address	Source IP Address (PC)	A list of IP addresses assigned to network interface cards installed on your PC. **NOTE:** If the configured main IP address of the CPU is not in the subnet of any of the IP configured on the interface cards of the PC, then the first interface card IP is suggested by default.
	Sub-Network Mask (read-only)	The subnet mask that is associated with the selected source IP address (PC) .
EtherNet/IP Network Detection (see note)	Begin detection range address	The first IP address in the address range for automatic fieldbus discovery of EtherNet/IP devices.
	End detection range address	The last IP address in the address range for automatic fieldbus discovery of EtherNet/IP devices.
Modbus Network Detection	Begin detection range address	The first IP address in the address range for automatic fieldbus discovery of Modbus TCP devices.
	End detection range address	The last IP address in the address range for automatic fieldbus discovery of Modbus TCP devices.
NOTE: To use the fieldbus discovery service, connect the workstation directly to the device network. If you connect to the device network through a BMENOC0321 control network module, the IP Forwarding service blocks the broadcast messages that are required to discover the network devices.		

Make the Connection

Connect to the **Source IP Address** (PC) :

Step	Action
1	Select an IP address from the **Source IP Address** (PC) pull-down menu.
2	Press the **Apply** button.
3	In the **DTM Browser**, find the name that you assigned to the BMENOC0321 module.
4	Right-click on the name of the CPU and scroll to **Connect**.

TCP/IP Monitoring

Expand (+) the **Channel Properties** heading in the configuration tree and select the **TCP/IP** item at level 1.

The read-only information on this page monitors the IP parameters that were configured in Control Expert.

Managing Source IP Addresses 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.

Switch Properties

Introduction

Use the **Switch** properties to perform these tasks:

- Enable or disable the Ethernet ports on the BMENOC0321 Ethernet communication module.
- View and edit the baud rate for each port, including the transmission speed and duplex mode.

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

Access the Switch Properties

View the **Switch** properties for the BMENOC0321 module:

Step	Action
1	Open the DTM Browser *(see page 65)* and view the **Channel Properties** for the module.
2	Expand (+) the **Channel Properties** to see the **Switch** page.
3	Select the **Switch** page to see the configurable properties.

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

Properties

Configure **Switch** properties to conform to your application:

Column	Description
Port	This read-only column shows the Ethernet ports that are connected to the module's internal switch (ETH 1, ETH 2, etc.) and the backplane port.
Enabled	Scroll to enable (**Yes**) or disable (**No**) a port. **NOTE:** • The port is disabled by default. • When you enable IPsec, the DTM automatically disables the backplane Ethernet port on the BMENOC0321. This isolates the IPsec network (control room network) from the device network. (Refer to the table for using different services and protocols *(see page 41)*.)
Baud Rate	Select a baud rate for the enabled port (see below).

The baud rate for the enabled backplane port is **100 Mbits/sec Full duplex**.

Select a baud rate for an enabled Ethernet port:

Port	Available Baud Rates
ETH 1	Auto 10/100Mbits/sec (default)
	100 Mbits/sec Half duplex
	100 Mbits/sec Full duplex
	10 Mbits/sec Half duplex
	10 Mbits/sec Full duplex
ETH 2, ETH 3	Auto 10/100/1000Mbits/sec (default)
	1000 Mbits/sec Half duplex
	1000 Mbits/sec Full duplex
	100 Mbits/sec Half duplex
	100 Mbits/sec Full duplex
	10 Mbits/sec Half duplex
	10 Mbits/sec Full duplex
Backplane Port	100 Mbits/sec Full duplex

NOTE: Schneider Electric recommends this default baud rate. With this setting, connected devices perform auto-negotiation and thereby determine the fastest common transmission rate and duplex mode.

NOTE: Cable recommendations:
To connect a BMENOC0321 control network module to a control network in a Modicon M580 system, Schneider Electric recommends the use of these cables:

- *10/100 Mbps*: For a communications link less than or equal to 100 Mbps, use CAT5e or CAT6 copper shielded twisted four-pair cables.
- *1000 Mbps*: For a communications link less than or equal to 1000 Mbps, use only CAT6 copper shielded twisted four-pair cables.

TCP/IP Properties

Introduction

The read-only information on the **TCP/IP** page monitors the IP parameters that were configured in Control Expert.

Use the channel **Configuration** tab of the module to edit the IP addressing settings to use in **Static** configuration mode.

Access the Configuration Tab

Access the channel **Configuration** tab for the BMENOC0321 control network module:

Step	Action
1	In the **Project Browser**, double-click **Project → Configuration → PLC bus**.
2	In the **PLC Bus** dialog box, right-click the BMENOC0321 control network module and click **Open**. **Result**: The configuration window for the module is displayed.
3	Select **Channel 0** to display the **Configuration** tab.

Configured Addresses

The BMENOC0321 control network module uses the scanner IP address, gateway IP address, and subnetwork mask configured in this **Configuration** tab:

IP Parameter	Description
Module IP Address	This 32-bit identifier consists of a network address and a host address that are assigned to a device that is connected to a TCP/IP Internet network through the Internet Protocol (IP).
Sub-Network Mask	This 32-bit value hides (or masks) the host portion of the IP address to set the network address of the module.
Gateway IP Address	When necessary, this device address serves as a gateway to other parts of the network.

Hot Standby Considerations

In a Hot Standby system *(see Modicon M580 Hot Standby, System Planning Guide for, Frequently Used Architectures)*, distributed equipment uses the **Main IP address** setting of the CPU to communicate over an Ethernet network with the primary CPU.

NOTE: Configure the **Main IP address** in the **IP Config** tab *(see Modicon M580, Hardware, Reference Manual)* for the M580 CPU.

On switchover, the **Main IP address** setting is automatically transferred from the former primary CPU to the former standby CPU (now the new primary CPU). Similarly, on switchover, the **Main IP address + 1** setting is automatically transferred from the former standby CPU to the new standby CPU.

In this way, the configured links between the distributed equipment and the primary CPU do not require editing in the event of a switchover.

In an M580 Hot Standby system, the standby BMENOC0321 module uses the same IP+1 address as the BMENOC0301/11 module on the local rack. Confirm that you configure the IP address used in the BMENOC0301/11 module differently than the IP address of the BMENOC0321 module (for the control network and the fieldbus network when IP forwarding is enabled). Use an Ethernet network manager tool to verify the system is working.

NOTE:

- BMENOC0301/BMENOC0311: A switchover does not affect the assignment of **IP address A** or **IP address B**. These assignments are made exclusively by means of the rotary switch on the back of the CPU and are not affected by a change in primary or standby Hot Standby status.
- BMENOC0321: IP address A and B are not defined.

Default Address Configurations

The BMENOC0321 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 these default address configurations:

Default Address	Default Address	
Main IP address (see note)	control network:	169.254.10.MAC
	Ethernet RIO (device) network:	169.254.20.MAC
	extended DIO network:	169.254.30.MAC
Subnetwork mask	255.255.255.0.	
Gateway address	The default gateway address is not identical to the default IP address.	
NOTE: These addresses use the right-most octet of the MAC Address and use the subnet mask 255.255.255.0.		

The Ethernet communication module provides these basic services when it uses the default IP address (and the services are enabled in the configuration):

- FTP server (used for firmware download)
- HTTP/Web server
- Modbus TCP server
- EtherNet/IP explicit message server
- SNMP agent
- IP forwarding
- RSTP

Duplicate Address Checking

NOTICE
UNEXPECTED EQUIPMENT BEHAVIOR Confirm that each module has a unique IP address. Duplicate IP addresses can cause unpredictable module/network behavior. **Failure to follow these instructions can result in equipment damage.**

The module checks for duplicate IP addresses before it applies the configured IP address:

Response	Meaning
yes	Another network device is using the proposed IP address.
	The module does not use the proposed IP address. It uses the default IP address.
no	The module uses the proposed IP address (along with the associated network parameters).

To improve performance during a network power-up operation, power up the network switches before you power up any system component (Ethernet communications module, Modicon M580 rack, PACs, etc.).

NOTE: When the entire network powers up at once, some switches may be slower to complete the process. The relatively slow switch response can cause some ARP messages to be dropped, resulting in an incomplete detection of duplicate IP addresses.

Section 5.3
Ethernet Services

What Is in This Section?

This section contains the following topics:

Enabling and Disabling Ethernet Services

Introduction

The BMENOC0321 control network module provides several Ethernet services. Use the **Services** page in the Control Expert DTM to enable and disable those services.

Enabling/Disabling Ethernet Services

View the **Services** for the BMENOC0321 module:

Step	Action
1	Open a Control Expert project that includes a BMENOC0321 module *(see page 52)*.
2	Open the **DTM Browser** (**Tools → DTM Browser**).
3	In the **DTM Browser**, find the name that you assigned to the BMENOC0321 module *(see page 53)*.
4	Double-click the name of the BMENOC0321 to open the configuration window. **NOTE:** You can also right-click on the module and scroll to **Open** to open the configuration window.
5	Select **Services** in the navigation tree.
6	Enable or disable each feature: ● **Enabled:** Scroll to **Enabled** to enable the service. ● **Disabled:** Scroll to **Disabled** to disable the service.
7	Click a button: ● **Apply**: Save changes with the window open. ● **OK**: Save changes and close the window.
8	Expand (+) **Services** in the navigation tree to view the enabled services.

NOTE:

- Services that are enabled appear in the expanded **Services** tree.
- You can configure the settings for any enabled service. If you enable a service that you do not configure, the Control Expert DTM applies the default settings.

Available Services

These Ethernet services are provided by the BMENOC0321 Ethernet communications module:

Service	Description	Default
Address Server *(see page 96)*	Provides IP addressing parameters and operating parameters to other Ethernet devices.	enabled
SNMP *(see page 99)*	• Serves as an SNMP v1 agent. • Provide trap information to up to two devices configured as SNMP managers. **NOTE:** The SNMP service is enabled by default and cannot be disabled.	enabled
RSTP *(see page 101)*	Supports RSTP in combination with other similarly-configured network devices to manage redundant physical connections and create a loop-free logical path that connects network devices.	enabled
Network Time Service *(see page 104)*	Provides the source time synchronization signal for the BMENOC0321 module.	disabled
QoS *(see page 107)*	Adds DSCP tags to Ethernet packets so that network switches and routers can prioritize the transmission and forwarding of IP frames. **NOTE:** Before enabling QoS tagging, confirm that devices connected to the Ethernet communication module support DSCP tags.	enabled
Service Port *(see page 109)*	Supports connection to a control network through the service port.	enabled
IP Forwarding *(see page 111)*	Performs IP Forwarding of Ethernet packets, separating traffic between the control network, the device network, and the embedded network.	disabled
SMTP *(see page 113)*	The simple mail transfer protocol (SMTP) provides mechanisms that allow controller-based projects to report alarms or events.	disabled

Configuring the FDR Address Server

About the FDR Service

The Ethernet communications module includes a fast device replacement (FDR) server. That server provides operating parameter settings to replacement Ethernet devices that are equipped with FDR client functionality.

Any networked Ethernet device that is equipped with FDR client functionality can subscribe to the Ethernet communications 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 cannot store any additional client FDR files.

The Ethernet communications 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.

In an M580 Hot Standby system, the PRM files managed by the FDR server in both modules are synchronized when the applications in both PACs are the same. Refer to the discussion of FDR in Hot Standby systems in the *Modicon M580 Hot Standby System Planning Guide for Frequently Used Architectures*.

FDR Address Server Configuration

Configure the address server service with the Control Expert DTM to set IP parameters for an Ethernet device that is based on a unique name (device name) or the MAC address of the device:

Step	Action
1	Enable the **Address Server** on the **Services** page *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.
2	Expand (+) **Services** and select **Address Server**.
3	In the **FDR Server** menu, scroll to **Enabled** to enable the FDR server.
4	View these tables: • **Automatically Added Devices:** This table shows the devices (and the corresponding IP addresses) that are automatically included in the module configuration. • **Manually Added Devices:** This table shows the devices (and the corresponding IP addresses) that you add to the module configuration. **NOTE:** • The automatic and manual addition of devices are described below. • The same IP address cannot appear in both the **Manually Added Devices** table and the **Automatically Added Devices** table.
5	Press a button to finish: • **Apply**: Save changes with the window open. • **OK**: Save changes and close the window.

This service also allows a device to store the configuration of the communications module in local non-volatile memory. The address server automatically provides correct network and device parameters for replacement devices without stopping the process.

Manually Adding Remote Devices to the DHCP Service

You can manually add a device DTM that corresponds to a device in the **Device List** to the address server service of the Ethernet communications module. Devices that are equipped with DHCP or BOOTP client software can be added.

Add devices to the **Manually Added Devices** list:

Step	Action
1	In the **Address Server** page, click the **Add** button to add a new row to the list of **Manually Added Devices**.
2	In the new row, configure these parameters for the client device: ● *IP Address*: Double-click the cell in the **IP address** column and enter an IP address for the client device. ● *Identifier Type*: Scroll to the type of value that the client device uses to identify itself to the FDR server: ❍ MAC Address ❍ Device Name ● *Identifier*: Depending upon the identifier type, enter the client device setting for the MAC address or name. ● *Mask*: Enter the client device subnet mask. ● *Gateway*: Enter the gateway address that remote devices can use to communicate with devices located on other networks. Use 0.0.0.0 if remote devices do not communicate with devices on other networks.

Viewing the Auto-Generated Client List

The **Automatically Added Devices** table automatically displays a list of devices that fit these criteria:

- The devices correspond to a device in the **Device List**.
- The devices subscribe to the Ethernet communications module's IP 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.

These columns appear in the **Automatically Added Devices** list:

Column	Description
Device No	This number is assigned to the device in the Control Expert configuration.
IP Address	This address corresponds to the client device.
DHCP	**TRUE** indicates that the device subscribes to the DHCP service.

Column	Description
Identifier Type	*Identifier Type*: This is the type of value that the client device uses to identify itself to the FDR server: • MAC address • Device Name
Identifier	This is the MAC address or device name.
Netmask	This is the subnet mask of the client device.
Gateway	This is the IP address of the network device that a DHCP client device uses 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.

Example: DHCP Server Providing IP Addresses for Local and Remote Subnets

Refer to the appendix *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* for an example of configuring a DHCP server to provide IP addresses to devices in local and remote subnets.

Configuring the SNMP Agent

Introduction

The BMENOC0321 control network 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 two 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 BMENOC0321 module. The SNMP agent can communicate with up to two SNMP managers as part of an SNMP service.

View the Page

Display the **SNMP** page:

Step	Action
1	Open a Control Expert project that includes a BMENOC0321 module *(see page 52)*.
2	Open the **DTM Browser** (**Tools → DTM Browser**).
3	In the **DTM Browser**, find the name that you assigned to the BMENOC0321 module *(see page 53)*.
4	Double-click the name of the BMENOC0321 to open the configuration window. **NOTE:** You can also right-click on the module and scroll to **Open** to open the configuration window.
5	Expand (+) **Services** in the navigation tree.
6	Select **SNMP** to see the configuration options.

NOTE: You cannot disable the SNMP service.

Viewing and Configuring SNMP Properties

View and edit these properties on 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 one: ● **Disabled**: You can edit the Location and Contact settings on this page. ● **Enabled**: You cannot edit the Location and Contact settings on this page. (Those settings are managed by the SNMP Manager.)
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 = **private**).
	Trap	Password an SNMP manager requires from the SNMP agent before the manager will accept trap notices from the agent (default = **alert**).
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 = **Disabled**).

Apply the configuration by clicking a button:

- **Apply**: Save changes.
- **OK**: Save changes and close the window.

NOTE:

- To help ensure cyber security, confirm that you change the password with modules that have firmware V1.05 or later.
- You cannot reset the module to factory settings if you lose the password.

Configuring the Rapid Spanning Tree Protocol

Introduction

The Ethernet control network ports (**ETH 2**, **ETH 3**) on the front of the BMENOC0321 control network module support the *Rapid Spanning Tree Protocol*. RSTP is an OSI layer 2 protocol defined by IEEE 802.1D 2004. The protocol performs these services:

- RSTP creates a loop-free logical network path for Ethernet devices that are part of a topology that includes redundant physical paths. When either device network port (**ETH 2** or **ETH 3**) on the BMENOC0321 module is connected to a daisy-chain loop topology, the RSTP service directs network traffic to the other port.
- RSTP automatically restores network communications by activating redundant links when a network event causes an interruption in service.

NOTE:

- When an RSTP link is connected, the RSTP service acts on an event and forwards traffic through the correct port. During this re-connect time, some packets may be lost.
- The reconnection time is 50ms maximum when all devices in the RSTP domain have the same behavior.

RSTP software, operating simultaneously in all network switches, obtains information from each neighboring 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.

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

View the Page

Display the **RSTP** page:

Step	Action
1	Open a Control Expert project that includes a BMENOC0321 module *(see page 52)*.
2	Enable RSTP *(see page 94)* for the module on the **Services** page.
3	Open the **DTM Browser** (**Tools → DTM Browser**).
4	In the **DTM Browser**, find the name that you assigned to the BMENOC0321 module *(see page 53)*.
5	Double-click the name of the BMENOC0321 to open the configuration window. **NOTE:** You can also right-click on the module and scroll to **Open** to open the configuration window.
6	Expand (+) **Services** in the navigation tree.
7	Select **RSTP** to see the two configuration tabs, **General** and **Advanced**. **NOTE:** The **Advanced** tab appears only when you enable the DTM's Advanced Mode *(see page 72)*

Assign the Bridge Priority

The bridge priority is a 2-byte value for the switch. The range for valid values is 0 ... 65535, with a default of 32768 (the midpoint).

Select the **General** tab to configure the Bridge Priority:

Step	Action
1	Select a Bridge Priority from the drop-down list in the **RSTP Operational State** area: • **Root (0)** • **Backup Root (4096)** • **Participant (32768)** (default)
2	Finish the configuration: • **OK**: Assign the Bridge Priority and close the window. • **Apply**: Assign the Bridge Priority and keep the window open.

NOTE: The Bridge Priority value is used to establish the relative position of the switch in the RSTP hierarchy.

Advanced Configuration

Select the **Advanced** tab to configure these parameters when **Advanced Mode** is enabled *(see page 72)*:

Field	Property	Description
Bridge Parameters	**Maximum Age Time**	The switch waits this length of time (6 ... 40 sec) for receipt of the next hello message before it initiates a change to the RSTP topology. (Default = 40 sec.)
	Transmit Hold Count	The maximum number of BPDUs (1 ... 40) that the switch can transmit per second. (Default = 40.)
	Hello Time	The embedded switch sends heartbeat BPDUs at this (read-only) frequency (2 sec).

Field	Property	Description
Port Parameters (ETH 2, ETH 3)	**RSTP**	This (read-only) 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. 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
	RSTP Cost	Select a method to determine the RSTP cost of the path through the embedded switch: ● **Auto**: The RSTP protocol automatically assigns a value to the switch by operation of the RSTP algorithm. ● **Manual**: Input the RSTP cost integer (1 ... 200000000) in the **Value** field.
	Edge Port	Set to a fixed (read-only) 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 Network Time Service

Introduction

The network time protocol (NTP) service synchronizes the clock in the BMENOC0321 module with the clock of a time server. The synchronized value is used to update the clock in the module. Time service configurations typically use redundant servers and diverse network paths to achieve high accuracy and reliability.

Considerations:

- When the BMENOC0321 module acts as an NTP client in an M580 Hot Standby system, the module polls the server in both primary and standby states. In this instance, the module does not act as an NTP server.
- The BMENOC0321 module does not maintain the time through a power cycle. After a power cycle, the module gets a time at the next NTP synchronization.
- This service does not update the time in the CPU. The updated time for the BMENOC0321 module is independent of the CPU time.

Refer to the System Time Stamping User Guide *(see System Time Stamping, User Guide)* for detailed time synchronization information.

Time Synchronization Service Features

These are some of the features of the time synchronization service:

- The periodic time correction is obtained from the reference-standard time server.
- The functionality automatically switches to a backup time server when errors are detected with the primary time server system.
- The local time zone is configurable (including daylight savings time).

Time Synchronization Process

The NTP client sends requests to the NTP server in the network to get the reference time for synchronizing the local time of the Ethernet communications module:

Stage	Description
1	Through an Ethernet network, an NTP client requests a time synchronization signal from an NTP server.
2	The NTP client calculates the correct time and stores the value.

Power Up

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

- The Ethernet communications module powers up.
- The Ethernet communications module obtains the time from the NTP server.
- The service requires the requests to be sent periodically to obtain and maintain accurate time. Your **Polling Period** configuration (below) partially determines the accuracy of the time.

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

The Ethernet communications module does not maintain the time. Upon power up or power cycle, the clock value of the module is 0, which is equivalent to January 1st 1980 00:00:00:00.

Stop or Run PAC

- Stop and run have no effect on the accuracy of the clock.
- Stop and run have no effect on the update of the clock.

Configuring the Service

Configure the network time synchronization service in the Control Expert DTM:

Step	Action
1	Enable the **Network Time Service** *(see page 94)* on the **Services** page.
2	In the navigation tree, expand (+) **Services**.
3	Select the **Network Time Service** node to see the configurable parameters.
4	Enter changes in the appropriate fields on the **Network Time Service** configuration page. (The following table describes the configuration page parameters.)
5	Press a button to finish: ● **Apply**: Save changes and leave the window open. ● **OK**: Save changes and close the window.

Configurable Parameters

Configure these time synchronization parameters:

Field	Parameter	Description
NTP Server Configuration	**Primary NTP Server IP Address**	Enter a valid IP address for each.
	Secondary NTP Server IP Address	
	Polling Period	The polling period is number of seconds (1 ... 120, default = 20) between updates from the NTP server. A smaller polling period results in better accuracy.
Time Zone	pull-down menu	Select the desired time zone relative to UTC. (The default value is the time zone associated with the PC of your operating system.)
	Time Zone Offset	The offset value (minutes) is the difference in your configured time zone and UTC.
	NOTE: When you select a specific time zone, you cannot modify the **Daylight Saving** parameters (below).	
Daylight Saving	**Automatically adjust clock ...**	**Disabled**: The local time is not subject to daylight saving adjustment.
		Enabled: The Ethernet communications module automatically corrects the local time to account for daylight saving time. The **Start Daylight Saving** and **End Daylight Saving** fields are disabled because the dates are a part of the standard time zone info.
	Start Daylight Saving, **End Daylight Saving**	**Month**: January ... December
		Day of Week: Sunday ... Saturday
		Occurrence: 1 ... 5 (Some months can have five occurrences of the same day. A selection of *5* uses the last occurrence in any month.)
		Hour: Select the hour (0 ... 23) to change the time.
	NOTE: To manually configure the **Daylight Saving** parameters, perform these steps: ● Select **Custom Time Zone** in the **Time Zone** pull-down menu. ● Select **Enabled** in the menu for **Automatically adjust clock for daylight saving**.	

Configuring DSCP Values for QoS

Description

The BMENOC0321 control network module can be configured to use the Different Service Code Point (DSCP) service in the IP packets. When you enable QoS, the module adds a DSCP value to the IP header of the Ethernet frame to indicate the frame priority.

NOTE: The BMENOC0321 module supports the OSI layer 3 Quality of Service (QoS) standard defined in IEEE RFC 2475.

Use the **QoS** page to view or edit the QoS DSCP prioritization values.

Configuration

Configure the QoS service:

Step	Action
1	Enable the **QoS Tagging** field *(see page 94)* on the **Services** page.
2	Expand (+) the **Services** page to see **QoS** in navigation tree.
3	Select the **QoS** node to see the configurable parameters.
4	Enter changes in the appropriate fields on the **QoS** configuration page. (The table below describes the traffic settings.)
5	Press a button to finish: ● **Apply**: Save changes with the window open. ● **OK**: Save changes and close the window.

QoS Settings

Use these guidelines to effectively implement QoS settings in your Ethernet network:

- Use network switches and routers 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.

Schneider Electric recommends that these QoS values be set in the configuration.

Use the Control Expert DTM to set default values for EtherNet/IP traffic, Modbus TCP Traffic, and the Network Time Protocol traffic:

Field	Traffic	Default
EtherNet/IP Traffic	DSCP Value for I/O Data Scheduled Priority Messages	43
	DSCP Value for Explicit Messages	27
	DSCP Value for I/O Data Urgent Priority Messages[1]	55
	DSCP Value for I/O Data High Priority Messages[1]	43
	DSCP Value for I/O Data Low Priority Messages[1]	31

Field	Traffic	Default
Modbus TCP Traffic	DSCP Value for I/O Messages	43
	DSCP Value for Explicit Messages	27
Network Time Protocol Traffic	DSCP Value for Network Time Protocol Messages	59
[1]Enable **Advanced Mode** *(see page 72)* to access these fields.		

Configuring the Service Port

Introduction

Follow these steps to configure the service port (ETH 1 on the front of the BMENOC0321 control network module *(see page 19)*) as an access port, a port mirroring port, or an extended DIO network port.

View the Page

Enable service port configuration:

Step	Action
1	Enable the **Service Port** *(see page 94)* on the **Services** page *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.
2	Select **Service Port** in the navigation tree.
3	In the **Service Port Mode** list, select: ● **Access Port** (default) ● **Port Mirroring** ● **Extended Network**
4	Press a button to finish: ● **Apply**: Save changes with the window open. ● **OK**: Save changes and close the window.

Access Port Mode

In **Access Port** mode, the service port is **Enabled** and cannot be changed. Connect these types of devices to the service port in this mode:

- HMI
- a PC with Control Expert software
- a PC with ConneXium Network Manager software

You can communicate with the CPU/PAC or the BMENOC0321 module itself. You can also access other devices that are connected to the network.

Port Mirroring Mode

Select **Port Mirroring** mode to configure the port to monitor and capture traffic to support a network analyzer (like Wireshark). In this mode, the service port is a read-only port. That is, you cannot communicate with Ethernet devices through the service port.

In the **Port Mirroring** area, use the **Source Port** property to enable specific ports:

- **Yes**: Traffic to and from this port is mirrored to the service port.
- **No**: Traffic to and from this port is not monitored by the service port.

The service port monitors traffic to the enabled ports:

Source Port	Description
Internal Port	Monitor Ethernet traffic to and from the module via the service port.
ETH 2	Ethernet traffic to and from port ETH 2 is sent to the service port.
ETH 3	Ethernet traffic to and from port ETH 3 is sent to the service port.
Backplane Port	Ethernet traffic to and from the backplane port is sent to the service port.

NOTE: If a device that is connected to the service port, ETH 2 port, or ETH 3 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.

Extended Network Mode

Select this mode to extend the device network. Add an extended DIO network *(see Modicon M580 Standalone, System Planning Guide for, Frequently Used Architectures)* to an M580 system by following the directions in the *Modicon M580 Standalone System Planning Guide for Frequently Used Architectures*.

Online Configuration

Configure the service port online with Control Expert using CIP explicit messaging *(see page 27)*, but this configuration may be lost when the BMENOC0321 Ethernet communications module is reset.

Configure the service port online with Control Expert using CIP explicit messaging. Refer to the description of the service port control object *(see page 261)* The CIP object configuration is stored in volatile memory. When the BMENOC0321 Ethernet communications module is reset it reverts to the service port configuration in the DTM (above)

Configuring the IP Forwarding Service

Introduction

The BMENOC0321 control network module includes an IP forwarding service. The IP forwarding service provides transparency between networks in a PlantStruxure system and can route packets among a maximum of three subnetworks that each has its own distinct broadcast domain.

NOTE: You cannot enable the IPsec protocol *(see page 119)* and the IP Forwarding service at the same time. (You cannot build a Control Expert project when both are enabled.)

Use the Control Expert DTM to configure the IP forwarding service by assigning unique IP address parameters (including the IP address and subnetwork mask) for the BMENOC0321 control network module to facilitate communications among these networks:

- control network
- device network
- extended network

You can also identify the default gateway for the BMENOC0321 control network module. (Refer to the description of the role of the default gateway *(see page 38)*.)

NOTE: The default gateway is the IP address of the control network router. Usually this router is a device that connects the control network to other networks higher up in the Ethernet infrastructure.

Mapping BMENOC0321 Module Ports to Subnets

When the IP Forwarding service is enabled, these IP ranges are assigned to the ports on the BMENOC0321 control network module:

Port	Configured with the IP Address for the...	Typical Usage
ETH 1	Extended network	The service port can be connected to a BMENOC0301 or BMENOC0311 module that communicates with an extended DIO network when the service port is configured for **Extended Network Mode**.
ETH 2, ETH 3	Control network	Use one or both of these ports to connect to the control network. Each port is assigned the IP address settings input into the **Control Network** area.
Backplane	Device network	Use the backplane port to communicate with the CPU, and through it to modules on the RIO main ring, to modules on RIO sub-rings, and to distributed equipment on DIO sub-rings.

Displaying the IP Forwarding Service Parameters

To display the **IP Forwarding** page and access the parameters:

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 **IP Forwarding** field to **Enabled**. Then click **Apply**. **Result**: The **IP Forwarding** service appears in the navigation tree.
3	Click **IP Forwarding** in the navigation tree.
4	Enter IP addressing parameters for the **IP Forwarding** service.
5	Click **Apply** to save changes and leave the window open, or click **OK** to save changes and close the window.

Configuring Electronic Mail Notification

Introduction

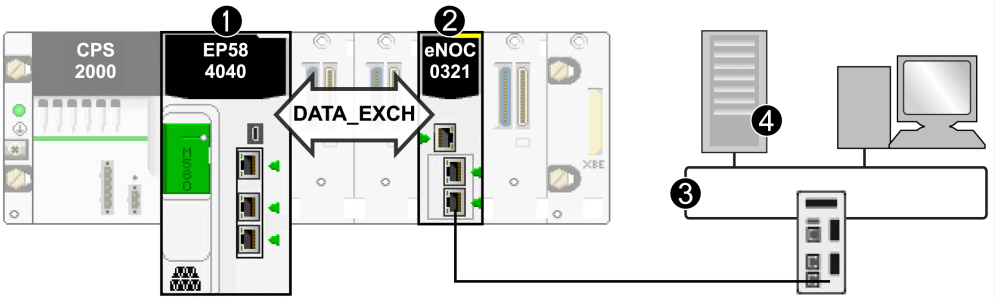
The electronic mail notification service allows controller-based projects to report alarms or events. The controller monitors the system, and can automatically create an electronic mail message alert with data, alarms, and/or events. Mail recipients can be either local or remote.

- Based on predefined events or conditions, messages are created using the DATA_EXCH function block *(see page 151)*.
- The email message is constructed from predefined headers, plus variables and text (a maximum of 238 bytes). This message is sent directly from the automation system to the local email server.
- Mail headers contain common predefined items — recipient list, sender name, and subject. These items can be updated by an authorized administrator.

NOTE: Test the email block before using it in an application. If you improperly configure an email DATA_EXCH block to receive an email when a detected problem occurs, the email may not be sent as expected.

Mail Service Client

The BMENOC0321 control network module includes an SMTP client. When the module receives a specific DATA_EXCH request over X Bus from the Control Expert project, it sends an email message to the SMTP mail server:



1 M580 CPU
2 BMENOC0321 control network module
3 control network
4 SMTP server

Displaying the SMTP Page

To display the **SMTP** 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 **SMTP** field to **Enabled**. Then click **Apply**. **Result**: **SMTP** appears in the navigation tree.
3	Select **SMTP** 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.

Configuring the Mail Service

A user-defined event or condition triggers the DATA_EXCH block to create a message. Each message uses one of 3 user-defined headers. Each message sent from the controller may contain text and variable information (with a maximum of 238 bytes).

The project selects the appropriate header. Each header contains:

- sender's name
- list of recipients
- subject

View and edit these properties in the **SMTP** page:

Property	Description
SMTP Server IP Address	Enter the IP address of the mail server.
SMTP Server Port	The default TCP port number for **SMTP** is 25. Configure the port as specified by your local mail server.
Password Authentication	If security is needed, enable **Password Authentication** by selecting the check box. Enter values for: ● Login ❍ Any printable character allowed ❍ 64-character maximum ● Password ❍ Any printable character allowed ❍ 64-character maximum **NOTE:** You can use an optional login (system ID) and password to authenticate the connection to the SMTP mail server. The SMTP-supported authentication method is LOGIN.

Property	Description
Email Header	Each header contains: ● Sender's ID in the **From** field ❍ 32-character maximum (no spaces) **NOTE:** The minimum length of the local part of a valid email address (before the @ symbol) is three characters. ● List of recipients in the **To** field ❍ Separate each email address with a comma. ❍ 128-character maximum ● Fixed part of message in the **Subject** field[1] ❍ 32-character maximum [1] The **Subject** field consists of 2 parts: **1.** Fixed (32-character maximum) **2.** Dynamic (206-character maximum) An authorized administrator can define and update the text and variable information. Define the 3 mail headers to indicate different levels of importance. For example: ● Header 1 could be `Detected problem reported by PLC 10.` ● Header 2 could be `Notification from substation 10.` ● Header 3 could be `Info message from water system.` Listing different recipients in each of the 3 headers allows the right information to flow quickly to the correct recipients. The project adds pertinent information such as the specific device, process, or location. This pertinent information is added to the body of the mail message. Then, the complete message is sent to an electronic mail server for distribution to recipients. These recipients could be engineers, managers, or process owners.

Operating Modes and Sending Requests

Because the controller program sends the email request, a controller cannot send an email message either while in the stopped mode or while downloading a project. As soon as the controller is in run mode, the function block sends a request during the first project scan.

Diagnostic counters are reset to 0 after a power-up, a project download, or a reconfiguration of the mail service.

Error Codes

The codes that correspond to errors that are detected during the execution of this function are included in an appendix *(see page 388)*.

Advanced Settings Tab

Introduction

The EtherNet/IP **Advanced** tab is available for Ethernet communication modules that support the DIO scanner service.

Accessing the Advanced Tab

View the EtherNet/IP **Advanced** tab:

Step	Action
1	Find the Ethernet communication module in the Control Expert **DTM Browser**.
2	Right-click the module and scroll to **Device menu → Additional functions → Advanced Mode**.
3	Double-click the module in the **DTM Browser** to view the **Channel Properties**.
4	Expand (+) **Channel Properties**.
5	Select **EtherNet/IP** to view the items in the **Group/Parameter** column: • **Timeout**: EtherNet/IP timeout settings • **Behavior**: EtherNet/IP scanner behavior

Timeout Settings

These timeout settings are in the EtherNet/IP **Timeout** field:

Parameter	Value	Comment
FW_Open I/O Connection Timeout (msec)	4960	Specifies the amount of time the scanner waits for FW_Open response of an I/O connection.
FW_Open EM Connection Timeout (msec)	3000	Specifies the amount of time the scanner waits for FW_Open response of an EM connection.
EM Connection RPI (msec)	10000	Sets T->O and O->T RPI for all EM (explicit messaging) connections.
EM Request Timeout (sec)	10	Specifies the amount of time the scanner will wait between the request and the response of an explicit message.

Scanner Behavior

Configure the behavior of the DIO scanner in the EtherNet/IP **Behavior** field:

Parameter	Value	Comment
Allow RESET via explicit message	**False**	(Default.) The scanner ignores the Identity object reset service request.
	True	The scanner will reset if an Identity object reset service request is received.
Behavior when CPU state is STOP	**Idle**	(Default.) The EtherNet/IP I/O connection stays open, but the **Run/Idle** flag is set to Idle.
	Stop	The EtherNet/IP IO connection is closed.

Section 5.4
Security

What Is in This Section?

This section contains the following topics:

Configuring IP Secure Communications

Introduction to IPsec

The Internet Engineering Task Force (IETF) developed and designed Internet Protocol Security (IPsec) as an open set of protocol standards that make IP communication sessions private and secure. The IPsec functionality of the BMENOC0321 module supports the data integrity and origin authentication of IP packets.

Follow the steps below to create a specific IPsec configuration on a Windows 7 PC. For more information about IPsec, refer to the Internet Engineering Task Force website (www.IETF.org).

Client-initiated communications are not supported from the BMENOC0321 Ethernet communication module when IPsec is enabled. For example, peer-to-peer (BMENOC0321-to-BMENOC0321) communications are not supported when IPsec is enabled.

NOTE:

- You cannot enable the IPsec protocol and the IP Forwarding service *(see page 111)* at the same time. (You cannot build a Control Expert project when both are enabled. Refer to the table for using different services and protocols *(see page 41)*.)
- Use Unity Pro 11.1 with DTM v3.6.**x** (and later) to run IPsec.

Process Overview

Configure IPsec communications in these stages:

Stage	Name	Description
1	Policy	Create an IPsec policy *(see page 120)*.
2	Rule	**Tunnel Endpoint**: no tunnel (transport mode) *(see page 121)*
		Connection type: network connections or local area network *(see page 121)*

		IP Filter List *(see page 121)*:
		• IP filter 1:
		  ❍ *address*: IP address of the first BMENOC0321 module.
		  ❍ *protocol*: **Any**
		  ❍ *description*: BMENOC0321 module 1
		• IP filter 2:
		  ❍ *address*: IP address of the second BMENOC0321 module.
		  ❍ *protocol*: **Any**
		  ❍ *description*: BMENOC0321 module 2
		NOTE: Repeat these steps for each BMENOC0321 module in your configuration.

		IP Filter Actions *(see page 122)*: • *action*: block, permit, negotiate • *method*: SHA-1 (no encryption) • *key expiration*: 86400
		Authentication Method *(see page 123)*: pre-shared key

Stage	Name	Description
3	General Properties *(see page 124)*	Security policy name and description
		Policy change timeout
		Key exchange settings: ● PFS ● authentication timeout (2879 min.) ● Internet Key Exchange (IKE) security methods ❍ *key exchange encryption*: 3DES ❍ *Integrity*: SHA1 ❍ *Diffie-Hellman group*: 1024 - medium (2)
4	Enable/Disable	Enable or disable the IPsec policy *(see page 124)*.
5	DTM	Configure the pre-shared key in the Control Expert DTM *(see page 124)*.

Before You Begin

Configure IPsec manually for each PC that supports IPsec:

- These directions are for PCs that run Windows 7.
- Confirm that you have administrative privileges to configure IPsec.
- Harden the PC that hosts the IPsec client to decrease the attack surface and observe the defense-in-depth concept. Refer to *Schneider Electric's guidelines* to harden your PC to reduce the surface of vulnerability.

IP Security Policy

Create an IPsec policy to define the rules for secure communications within the IPsec protocol:

Step	Action
1	On a Windows 7 PC, open the **Administrative Tools** from the Control Panel. **NOTE:** Consult your Windows 7 documentation to access the **Administrative Tools**.
2	Double-click **Local Security Policy** to open the **Local Security Policy** window.
3	In the left pane, expand **Security Settings** and double-click **IP Security Policies on Local Computer**.
4	In the right pane, right-click and scroll to **Create IP Security Policy ...** to open the **Policy Wizard**.
5	In the **IP Security Policy Wizard**, press the **Next** button: **a.** Assign a name to a new **Security Policy** in the **Name** field. **b** Provide a description of the new policy in the **Description** field. (This step is optional).
6	Press the **Next** button to proceed to the **Requests for Secure Communication** window.
7	Uncheck the check box (**Activate the default ...**) and press **Next** to open the **Completing the IP Security Policy Wizard**.
8	Uncheck the **Edit properties** check box and press **Finish**.

NOTE: The new security policy appears in the right pane of the **IP Security Policies on Local Computer** window. You can double-click on the security policy at any time to access its **Properties** window.

IP Security Rule

Configure an IPsec rule to enable an IPsec configuration to monitor traffic between the application layer and the network layer:

Step	Action
1	In Windows 7, double-click the policy to open the **Properties** window.
2	Select the **Rules** tab.
3	Press **Add...** to open the **Create IP Security Rule Wizard**.
4	Press **Next** to configure the **Tunnel Endpoint**.
5	Select **This rule does not specify a tunnel** to use the **Transport** mode within the IPsec protocol.
6	Press **Next** to configure the **Network Type**.
7	Select the **All network connections** option button to apply the policy to local and remote connections.
8	Press **Next** to access the **IP Filter List** configuration. **NOTE:** The **IP Filter List** identifies the traffic that is processed through the IPsec rule.

IP Filter List

IPsec uses packet filters to evaluate communication packets according to their connections to various services. Packet filters are located between the endpoints of a peer-to-peer connection to verify that the packets adhere to the established administrative rules for communications.

Every IP filter in a single IP filter list has the IP address of the same source of the communications packets. The IP addresses for the destinations of communications packets (BMENOC0321 modules) are different.

Create a filter list that contains the IP addresses for the BMENOC0321 modules that can communicate with the source (PC):

Step	Action
1	In Windows 7, in the **IP filter lists** table of the **Security Rule Wizard**, click **Add** to create a new **IP filter list**: **a.** Assign a name to a new **Filter List** in the **Name** field. **b.** Provide a description of the new **Filter List** in the **Description** field. (This step is optional.)
2	Press **Add** to open the **IP Filter Wizard** and press **Next**.
3	Provide an optional description of the new **IP Filter** in the **Description** field.
4	Check the **Mirrored** check box to communicate in both directions (source and destination).
5	Press **Next** to configure the **IP Traffic Source**.

Step	Action
6	Scroll to **My IP Address** to designate the PC at one endpoint of the secure communications.
7	Press **Next** to configure the **IP Traffic Destination**.
8	Scroll to **a specific IP Address or Subnet** and enter the IP address of a BMENOC0321 module in your configuration. (The BMENOC0321 module is the only destination for this traffic.)
9	Press **Next** to configure the **IP Protocol Type** and select **Any** to allow traffic from the trusted IP address.
10	Press **Next** to view the **Completing the IP Filter Wizard** window.
11	Uncheck the **Edit properties** check box and press **Finish** to return to the **IP Filter List**.
12	Press **OK** to exit the **IP Filter List**.

IP Filter Actions

Configure filter actions:

Step	Action
1	In Windows 7, in the **Name** column of the **IP Filter List**, select the option button for the newly created IP filter list and click **Next** to configure the **Filter Action**.
2	Check the **Use Add Wizard** check box.
3	Press **Add** to open the **Filter Action Wizard**.
4	Press **Next** to configure the **Filter Action Name**: **a.** Enter a name for the **Filter Action** in the **Name** field. **b.** Provide an optional description of the new **Filter Action Name** in the **Description** field and press **Next**.
5	Select **Negotiate security** and press **Next**. **NOTE:** The source and destination addresses agree on a method for secure communication before packets are sent.
6	Select **Do not allow unsecure communication** and press **Next**.
7	Select **Custom** in the **IP Traffic Security** window and press **Settings** to customize the settings: **a.** Select **Data and Address integrity without encryption** and select **SHA1** in the pull-down menu to use secure hash algorithm 1. **b.** De-select **Data integrity with encryption** to disable the Encapsulating Security Payload (ESP).. **c.** Check the **Generate a new key every** check box and enter 86400 in the **seconds** field to specify that the IKE expires in 86400 seconds. **d.** Press **OK** to return to the **IP Traffic Security** configuration.
8	Press **Next**.
9	Check the **Edit properties** check box and press **Finish**.
10	Do not check the **Use session key perfect forward secrecy (PFS)** check box.
11	Press **OK**.

Authentication Method

Source and destination devices can agree to use a secret text string before communications begin. In this case, the string is called a pre-shared key.

Configure the authentication method to use a pre-shared key:

Step	Action
1	In Windows 7, in the **Name** column of the **Filter Actions**, select the option button for the newly created IP filter list and click **Next** to configure the **Authentication Method**.
2	Check the **Use this string to protect the key exchange (preshared key)** check box.
3	In the text field, use any 16 ASCII characters to create a case-sensitive name for the pre-shared key. **NOTE:** At the end of this process, you will configure an identical pre-shared key in the Control Expert DTM *(see page 124)* to create a connection between a specific IP address and the BMENOC0321 module.
4	Press **Next**.
5	Uncheck the **Edit properties** check box and press **Finish**.

IP Security Policy General Properties

Configure the general properties:

Step	Action
1	In Windows 7, in the **Properties** window, select the **General** tab.
2	Click **Settings** to open the **Key Exchange Settings** window.
3	Do not check the **Master key perfect forward secrecy (PFS)** check box.
4	In the **minutes** field, enter 2879 to set the key lifetime to 2879 minutes (47 hours and 59 minutes).
5	Click **Methods...** to open the **Key Exchange Security Methods** window.
6	Click **Edit** to open the **IKE Security Algorithms** window.
7	In the three pull-down menus, make these selections: • **Integrity algorithm**: **SHA1** (Secure Hash Algorithm 1) • **Encryption algorithm**: **3DES** (Triple Data Encryption Algorithm) • **Diffie-Hellman group**: **Medium (2)** (Generate 1024 bits of master key material.)
8	Press **OK** to return to the **Key Exchange Security Methods** window.
9	Press **OK** to return to the **Key Exchange Settings** window.
10	Press **OK** to return to the **Properties** window.
11	Press **OK** to close the **Properties** window.

Enable and Disable the Policy

Assign or un-assign a local security policy to enable and disable secure communications:

Step	Action
1	In Windows 7, open **Local Security Policy** in **Administrative Tools**.
2	Right-click the name of the new local security policy in the **Name** column and make a selection: ● **Assign**: Assign the local security policy to enable communications to the IPsec-enabled PC. ● **Un-assign**: Un-assign the local security policy to disable communications to the PC.

The IPsec policy agent does not run if you see this message: "The service cannot be started ...." In that case, configure the service to start automatically:

Step	Action
1	In Windows 7, expand (+) **Administrative Tools**.
2	Double-click **Services** to access the local services.
3	Double-click **IPsec Policy Agent** to open its properties.
4	Select the **General** tab.
5	In the **Startup type** pull-down menu, scroll to **Automatic**.
6	In the **Service status**, press **Start**. **NOTE:** When **Start** is grayed out, the service is already running.
7	Press **OK** to apply the changes and close the window.

NOTE: When you enable IPsec, the DTM automatically disables the backplane Ethernet port on the BMENOC0321. This isolates the IPsec network (control room network) from the device network. (Refer to the table for using different services and protocols *(see page 41)*.)

Configure IPsec in the Control Expert DTM

Enable IPsec and set the pre-shared key in the Control Expert DTM:

Step	Action
1	Open your Control Expert project.
2	Open the **DTM Browser** (**Tools → DTM Browser**).
3	In the **DTM Browser**, double-click the name that you assigned to the BMENOC0321 module *(see page 53)* to open the configuration window. **NOTE:** You can also right-click the module and select **Open** to open the configuration window.
4	Select **Security** in the navigation tree to view the configuration options.
5	In the **IPsec** menu, select **Enabled**.

Step	Action
6	In the **Pre-Shared Key** field, enter the 16-character name of the pre-shared key. **NOTE:** The ASCII characters in the case-sensitive pre-shared key match the 16-character pre-shared key that you defined earlier *(see page 123)*.
7	Press the **Apply** button to save the configuration.
8	Rebuild the project and download the application to apply these settings to the BMENOC0321 module.

Troubleshooting IPsec Communications

Use the standard Windows 7 IPsec diagnostic tools to troubleshoot IPsec communications. For example, these steps use the Microsoft Management Console (MMC) service for management applications:

Step	Action
1	In Windows 7, create a console that includes an IP Security Monitor.
2	Click a server name.
3	Double-click **Quick Mode**.
4	Double-click **Statistics** to see the number of authenticated bytes that are sent and received.

NOTE:

- You cannot reset the values. To refresh the count values, relaunch the Microsoft Management Console.
- Disable **IP Forwarding** *(see page 112)* before you enable IPsec. IPsec applies to a single IP address.

Use a Wireshark network analyzer to confirm that IPsec communications have started for an established IKE session. IPsec packets have an authentication header instead of the normal protocol header. This table shows an example of a network trace of a successful IKE session that was established by a ping request between a Windows 7 PC (source) and BMENOC0321 module (destination):

Number	Time	Source	Destination	Protocol	Length	Information
1	0	192.168.20.201	192.168.20.1	ISAKMP	342	Identity Protection (Main Mode)
2	0.00477	192.168.20.1	192.168.20.201	ISAKMP	126	Identity Protection (Main Mode)
3	0.012426	192.168.20.201	192.168.20.1	ISAKMP	254	Identity Protection (Main Mode)
4	1.594495	192.168.20.1	192.168.20.201	ISAKMP	270	Identity Protection (Main Mode)
5	1.598533	192.168.20.201	192.168.20.1	ISAKMP	110	Identity Protection (Main Mode)

Number	Time	Source	Destination	Protocol	Length	Information
6	1.603296	192.168.20.1	192.168.20.201	ISAKMP	110	Identity Protection (Maine mode)
7	1.612634	192.168.20.201	192.168.20.1	ISAKMP	366	Quick Mode
8	3.202976	192.168.20.1	192.168.20.201	ISAKMP	374	Quick Mode
9	3.207794	192.168.20.201	192.168.20.1	ISAKMP	102	Quick Mode

Use these solutions to facilitate communications when IPsec is enabled:

Behavior	Explanation
There is no communication with the BMENOC0321 when IPsec is enabled on the Windows PC.	**Explanation:** The IPsec policy agent is not running. **Solution:** Configure IPsec to start automatically *(see page 124)*.
	Explanation: IPsec is not enabled on the BMENOC0321. **Solution:** Enable IPsec *(see page 124)* on the **Security** tab of the BMENOC0321 DTM.
	Explanation: IPsec is not configured properly in Windows. **Solution:** See **NOTE 1** (below).
Control Expert cannot connect to the BMENOC0321 via Ethernet.	**Explanation:** IPsec is not enabled on both the BMENOC0321 and the Windows PC. **Solution:** See **NOTE 2** (below).
	Explanation: IPsec is not configured properly in Windows. **Solution:** See **NOTE 1** (below).
	Explanation: The power to the BMENOC0321 was recently cycled. **Solution:** See **NOTE 3** (below).
The firmware update tool is not able to connect to the BMENOC0321 via Ethernet.	**Explanation:** IPsec is not enabled on both the BMENOC0321 and the Windows PC. **Solution:** See **NOTE 2** (below).
	Explanation: IPsec is not configured properly in Windows. **Solution:** See **NOTE 1** (below).
	Explanation: The power to the BMENOC0321 was recently cycled. **Solution:** See **NOTE 3** (below).
	Explanation: The IKE and IPsec ports may be blocked by a firewall or another program associated with antivirus applications. **Solution:** See **NOTE 4** (below).
NOTE 1: Confirm that the parameters in the Windows configuration match those in the IPsec implementation: ● Double-check the pre-shared key *(see page 123)*. ● Double-check the IP address of the BMENOC0321 in the in the DTM *(see page 121)*. ● Disable **Perfect Forward Secrecy** for both communication endpoints in Windows *(see page 123)*.	
NOTE 2: Verify that the DTM configuration and the Windows **Local Security Policy** *(see page 124)* are enabled for IPsec.	

Behavior	Explanation
NOTE 3: Choose a solution: ● Wait 5 minutes for the Windows security associations to timeout. ● Unassign then reassign the local security policy *(see page 124)* in Windows to force the security associations to be reset.	
NOTE 4: Verify that the IKE port (UDP 500) and IPsec Authentication Header port (51) are open on any firewall between the PC application and the PAC, including the firewalls associated with antivirus applications (like McAfee or Symantec).	

Configuring Security Services

Introduction

The Control Expert DTM provides security services to the BMENOC0321 control network module. Enable and disable these services on the **Security** tab in the Control Expert DTM.

Access the Security Tab

View the **Security** configuration options:

Step	Action
1	Open your Control Expert project.
2	Open the **DTM Browser** (**Tools → DTM Browser**).
3	In the **DTM Browser**, double-click the name that you assigned to the BMENOC0321 module. *(see page 53)* to open the configuration window. **NOTE:** You can also right-click the module, and select **Open**.
4	Select **Security** in the navigation tree to view the configuration options.

NOTE: For general safety information, refer to the cyber security manual.

Service Selection

Enable and disable these services in the **Security** tab:

Service	Description
FTP	Enable or disable (default) these items: ● firmware upgrade ● device configuration management using the FDR service **NOTE:** Local data storage remains operational, but remote access to data storage is disabled.
TFTP	Enable or disable (default) the ability to read X80 I/O module configuration files using the FDR service. **NOTE:** In M580 Hot Standby systems, you can disable TFTP services in the Ethernet screen for the BMENOC0321 module. (You might do this if connected DIO modules either do not push their configuration to the FDR server *(see page 96)* in the module, or if they use only FTP to transfer their configuration to this server.) However, if TFTP is disabled, Hot Standby synchronization cannot be performed because it is based on TFTP.
HTTP	Enable or disable (default) the web access service.
Access Control	● **Enabled** (default): Deny Ethernet access to the Modbus and EtherNet/IP server by unauthorized network devices. ● **Disabled**: There is no restriction on which network devices can access the Modbus and EtherNet/IP server.

Service	Description
IPsec	Enable or disable (default) secure communications for traffic between the IP address that corresponds to a BMENOC0321 module and another IP address using IPsec *(see page 119)*.
Pre-Shared Key	This field is associated with IPsec, and is empty by default. If you enable IPsec, enter 16 characters. Select a value that is difficult to guess (combination of upper and lower case letters, numbers, and special characters).
DHCP / BOOTP	Enable or disable (default) the automatic assignment of IP addressing settings. Your DHCP selection also enables/disables automatic assignment of subnet mask, gateway IP address, and DNS server names.
SNMP	Enable or disable (default) the protocol used to monitor network-attached devices.
EIP	Enable or disable (default) access to the EtherNet/IP server and its electronic data sheets (EDS), which classify each network device and its functionality.

NOTE:

- The default settings represent the maximum security level. The increased security reduces the communication capabilities and the access to communication ports.
- Services that are selected online (through Control Expert or ETH_PORT_CTRL *(see page 132)*) apply only to the rack on which the EF runs.
- Refer to the discussion of the ETH_PORT_CTRL function block *(see page 132)* to enable/disable the FTP, TFTP, HTTP, and DHCP/BOOTP protocols.

Enabling Security

Set the **Security** tab parameters before you download the application to the CPU. When they are disabled, security services can be enabled only when you download a new application.

Use these steps to set the security level quickly:

Step	Action
1	In a respective service, select **Enabled** in the associated pull-down menu. **NOTE:** When you enable or disable a service, the pencil symbol appears to indicate that you are editing the security settings.
2	Click **Enforce Security** to reset services to their default states (above) and implement the highest level of security.
3	Click **Unlock Security** to use the lowest level security settings (opposite of default settings).
4	Click **Apply** to enable the service. **NOTE:** The pencil symbol disappears.
5	Save your project (**File → Save**).

Using Access Control for Authorized Addresses

Use the **Access Control** page to restrict device access to the BMENOC0321 module or the CPU communication server service via the BMENOC0321 module in its role as either a Modbus TCP, EtherNet/IP, FTP, TFTP, HTTP, or SNMP server. When you enable access control in the **Security** dialog, add to the list of **Authorized Addresses** the IP address of each device that is permitted to communicate with the BMENOC0321 module, to the list of **Authorized Addresses**:

- By default, the IP address of the BMENOC0321 module or the CPU communication server service via the BMENOC0321 module with **Subnet** set to **Yes** allows any device in the subnet to communicate with the BMENOC0321 module using EtherNet/IP and Modbus TCP.
- Add the IP address of any client device that may send a request to the BMENOC0321 module or the CPU communication server service via the BMENOC0321 module, which, in this case, acts as a Modbus TCP or EtherNet/IP server.
- Add the IP address of your maintenance PC to communicate with the PAC through the BMENOC0321 module or the CPU communication server service via the BMENOC0321 module (using Control Expert to configure and diagnose your application).
- A service column is grayed out in the **Authorized Addresses** grid if the respective service is disabled in the **Services** field.

You can enter a maximum of 128 authorized IP addresses.

Adding Devices to the Authorized Addresses List

To add devices to the **Authorized Addresses** list:

Step	Action
1	Set **Access Control** to **Enabled**.
2	In the **IP Address** column of the **Authorized Addresses** list, double-click the default IP address (0.0.0.0) to enter an IP address.
3	Enter the address of the device to access the BMENOC0321 module or the CPU communication server service via the BMENOC0321 module with either of these methods: ● *Add a single IP address*: Enter the IP address of the device and select **No** in the **Subnet** column. ● *Add a subnet*: Enter a subnet address in the **IP Address** column. Select **Yes** in the **Subnet** column. Enter a subnet mask in the **Subnet Mask** column. **NOTE:** A red exclamation point (!) indicates a detected error in the entry. You can save the configuration only after the detected error is fixed.
4	Repeat these steps for each additional device or subnet to which you want to grant access to the BMENOC0321 module or the CPU communication server service via the BMENOC0321 module. **NOTE:** You can enter up to 128 authorized IP addresses or subnets.
5	Click **Apply**.

Removing Devices from the Authorized Addresses List

To remove devices from the **Authorized Addresses** list:

Step	Action
1	In the **Authorized Addresses** list, select the IP address of the device to delete.
2	Press the **Delete** button.
3	Click **Apply**.

Finishing the Configuration

Click a button to finish:

- **OK**: Save changes and close the window.
- **Apply**: Save changes and leave the window open.
- **Cancel**: Cancel changes.

ETH_PORT_CTRL: Executing a Security Command in an Application

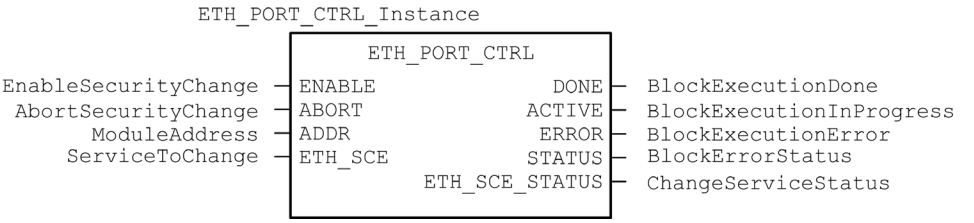
Function Description

Use the ETH_PORT_CTRL function block to control the FTP, TFTP, HTTP, and DHCP/BOOTP protocols when they are enabled in the **Security** screen *(see page 129)* of the Control Expert DTM. (By default, these protocols are disabled.)

The additional parameters EN and ENO may also be configured.

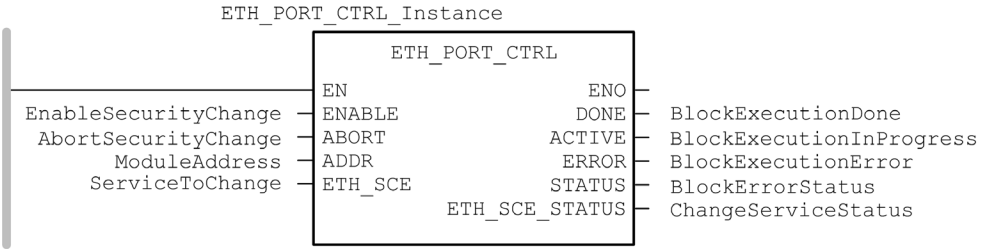
FBD Representation

Representation:



LD Representation

Representation:



IL Representation

```
CAL ETH_PORT_CTRL_Instance (ENABLE := EnableSecurityChange,  ABORT :=
AbortSecurityChange, ADDR := ModuleAddress, ETH_SCE := ServiceToChange,
DONE => BlockExecutionDone,  ACTIVE => BlockExecutionInProgress,  ERROR
=> BlockExecutionError,  STATUS => BlockErrorStatus,  ETH_SCE_STATUS =>
ChangeServiceStatus)
```

ST Representation

```
ETH_PORT_CTRL_Instance (ENABLE := EnableSecurityChange, ABORT :=
AbortSecurityChange, ADDR := ModuleAddress, ETH_SCE := ServiceToChange,
DONE => BlockExecutionDone, ACTIVE => BlockExecutionInProgress, ERROR
=> BlockExecutionError, STATUS => BlockErrorStatus, ETH_SCE_STATUS =>
ChangeServiceStatus);
```

Description of Parameters

The following table describes the input parameters:

Parameter	Type	Comment
ENABLE	BOOL	Set to 1 to enable the block operation.
ABORT	BOOL	Set to 1 to abort the currently active operation.
ADDR	ANY_ARRAY_INT	This array contains the address of the device for which you want to change the security state, which is the result of the ADDMX *(see EcoStruxure™ Control Expert, Communication, Block Library)* or ADDM *(see EcoStruxure™ Control Expert, Communication, Block Library)* function in this format: (*rack#*, *slot#*, *channel#*). For example: ● ADDM('0.0.3') for a M580 CPU ● ADDM('0.3.0') for a BMENOC in slot 3 of the main rack ● ADDMX('0.0.3{192.168.10.2}SYS) for a BMXCRA with the IP address 192.168.10.2 **NOTE:** ● To address a module in the local rack, enter 0.0.3 (CPU main server address). ● In M580 Hot Standby systems, ADDR represents the address of the primary controller. If you disable TFTP you disable the synchronization of the FDR service *(see page 96)*.
ETH_SCE	WORD	For each protocol, use these binary values to control the protocol: ● 00: The protocol is unchanged. ● 01: Enable the protocol. ● 10: Disable the protocol. ● 11: reserved **NOTE:** A value of 11 reports a detected error in ETH_SCE_STATUS. These bits are used for the different protocols: ● 0, 1: FTP ● 2, 3: TFTP (Only available for Modicon M580) ● 4, 5: HTTP ● 6, 7: DHCP / BOOTP ● 8...15: reserved (value = 0)

The following table describes the output parameters:

Parameter	Type	Comment
DONE	BOOL	Operation completed indication. Set to 1 when the execution of the operation is completed successfully.
ACTIVE	BOOL	Operation in progress indication. Set to 1 when the execution of the operation is in progress.
ERROR	BOOL	Set to 1 if an error is detected by the function block.
STATUS	WORD	Code providing the detected error identification *(see EcoStruxure™ Control Expert, Communication, Block Library)*.
ETH_SCE_ STATUS	WORD	For each protocol, these values contain the response to any attempt to enable or disable the FTP, TFTP, HTTP, or DHCP / BOOTP protocols: ● 0: command executed ● 1: command not executed Reasons for not executing the command can be: ● The communication service has been disabled by the configuration. ● The communication service is already in the state requested by the command (**Enabled** or **Disabled**). ● The communication service (x) is not supported by the module or is a non-existing service. These bits are used for the different protocols: ● 0: FTP ● 1: TFTP ● 2: HTTP ● 3: DHCP / BOOTP ● 4 ... 15: reserved (value = 0)

Execution Type

When used on a BMENOC0321 module, the ETH_PORT_CTRL function block is executed ***asynchronously*** and may take several cycles until the DONE output turns **ON**. Therefore, the ACTIVE output is set to **ON** until the completion of the ETH_PORT_CTRL function block.

How to Use the ETH_PORT_CTRL EFB

Follow these steps to use the ETH_PORT_CTRL EFB.

Step	Action
1	Set the bits of the services you want to activate in ETH_SCE.
2	Set ENABLE input to activate the EFB.
3	Reset ENABLE input as soon as the ACTIVE output is reset by the EFB.
4	Check STATUS output value: • STATUS<>0: There is a communication status code. • STATUS = 0: Check ETH_SCE_STATUS. The services for which the bits are set haven't been modified as they should be.

Section 5.5
Device List

What Is in This Section?

This section contains the following topics:

Device List Configuration and Connection Summary

Introduction

The **Device List** is part of the DTM data structure for the BMENOC0321 module. The **Device List** contains read-only properties that summarize these items:

- configuration data:
  - input data image
  - output data image
  - maximum and actual numbers of devices, connections, and packets
- Modbus request and EtherNet/IP connection summary

Open the Page

Open the **Device List** page:

Step	Action
1	Open your Control Expert Pro project.
2	Open the **DTM Browser** (**Tools → DTM Browser**).
3	In the **DTM Browser**, find the name that you assigned to the BMENOC0321 module *(see page 53)*.
4	Double-click the name of the BMENOC0321 to open the configuration window. **NOTE:** You can also right-click on the module and scroll to **Open** to open the configuration window.
5	Select **Device List** in the navigation tree.

Configuration Summary Data

Select **Device List** and view the **Configuration Summary** table on the **Summary** tab to see values for these items:

- **Input**
- **Output**
- **Configuration Size**

Expand (+) the **Input** row to view the **Input Current Size** values:

Description	Source
This value is the sum of all Modbus request and EtherNet/IP connection sizes.	This value is configured in the **General** page for a selected distributed device and connection.

Expand (+) the **Output** row to view the **Output Current Size** values:

Description	Source
This value is the sum of all Modbus request and EtherNet/IP connection sizes.	This value is configured in the **General** page for a selected distributed device and connection.

The maximum size of the X Bus input or output memory variable is 4 KB (2048 words). The variable contains a 16-byte descriptor followed by a value that represents the number of input or output data objects. Each data object contains a 3-byte object header followed by the input or output data. The number of data objects and the size of the input or output data depend on the configuration. The maximum overhead in the variable is 403 bytes (16 + 387), where 16 is the number of bytes in the descriptor and 387 is the product of 3 X 129, where 3 is the number of bytes in the header and 129 is the number of input or output objects (128 maximum scanned devices or local slaves that the BMENOC0321 module supports plus one (1) input or output object for the scanner DDDT). Therefore, at least 3.6 KB of the 4-KB variable is available for the input or output current size.

NOTE: The input current size also includes 28 words of scanner DDT input data. The output current size also includes 24 words of scanner DDT output data.

Expand (+) the **Configuration Size** row in the **Connection Summary** table to view these values:

Name	Description	Source
Maximum Number of DIO Devices	This value represents the maximum number of distributed devices allowed in the configuration.	capability of the module
Current Number of DIO Devices	This value is the number of active and inactive distributed devices and local slaves in the configuration.	number of devices in the **Device List**
Maximum Number of DIO Connections	This value represents the maximum number of connections that the Ethernet communications module can manage.	capability of the module
Current Number of DIO Connections	This value is the number of connections by active devices and local slaves in the configuration.	device configuration in the Control Expert Pro **Device Editor**
Maximum Number of Packets	This is the maximum number of Ethernet I/O scanning packets per second that the Ethernet communications module supports.	capability of the module
Current Number of Input Packets	This is an estimate of the number of input packets per second that the current configuration generates.	device configuration in the Control Expert Pro **Device Editor**
Current Number of Output Packets	This is an estimate of the number of output packets per second that the current configuration generates.	device configuration in the Control Expert Pro **Device Editor**
Current Number of Total Packets	This is an estimate of the total number of Ethernet I/O scanning packets per second that the current configuration generates.	device configuration in the Control Expert Pro **Device Editor**

Request / Connection Summary

Select **Device List** and view the **Request / Connection Summary** table on the **Summary** tab. The Control Expert Pro DTM uses this information to calculate the total bandwidth that distributed devices consume:

Column	Description
Connection Bit	• Connection health bits display the status of each device with one or more connections. • Connection control bits can be toggled on and off using object IDs.
Task	The task type (FAST, MAST).
Input Object	The input object number associated with the request or connection.
Output Object	The output object number associated with the request or connection.
Device	The device **Number** is used for the Health and Control Bit index.
Device Name	The label for the device in the **Device List**.
Type	The target device type: • EtherNet/IP • local slave • Modbus/TCP
Address	The target device IP address (except for local slaves).
Rate (msec)	The RPI (for EtherNet/IP) or the Repetitive Rate (for Modbus TCP). **NOTE:** The **Rate** does not apply to local slaves.
Input Packets per second	The number of input (T->O) Ethernet packets per second generated by this request or connection.
Output Packets per second	The number of output (O->T) Ethernet packets per second generated by this request or connection.
Packets per second	The sum of input packets per second and output packets per second for the request or connection.
Bandwidth Usage	The total amount of network bandwidth (total bytes traffic) consumed by this request or connection.
Size In	The number of input words configured for this request or connection.
Size Out	The number of output words configured for this request or connection.

Device List Parameters

Introduction

Configure parameters for devices in the **Device List** on these tabs:

- **Properties**
- **Address Setting**
- **Request Setting** (Modbus devices only)

View the Configuration Tabs

Navigate to the **Device List** configuration tabs

Step	Action
1	In the **DTM Browser** (**Tools → DTM Browser**), double-click the DTM that corresponds to the Ethernet communication module.
2	In the navigation pane, expand (+) the **Device List** *(see page 136)* to see the associated Modbus TCP and EtherNet/IP devices.
3	Select a device from the **Device List** to view the **Properties**, **Address Setting**, and **Request Setting** tabs. **NOTE:** These tabs are described in detail below.

Properties Tab

Configure the **Properties** tab to perform these tasks:

- Add the device to the configuration.
- Remove the device from the configuration.
- Edit the base name for variables and data structures used by the device.
- Indicate how input and output items are created and edited.

Configure the **Properties** tab:

Field	Parameter	Description
Properties	**Number**	The relative position of the device in the list.
	Active Configuration	**Enabled:** Add this device to the Control Expert project configuration.
		Disabled: Remove this device from the Control Expert project configuration.
IO Structure Name	**Structure Name**	Control Expert automatically assigns a structure name based on the variable name.
	Variable Name	**Variable Name**: An auto-generated variable name is based on the alias name.
	Default Name	Press this button to restore the default variable and structure names.

Field	Parameter	Description
Items Management	**Import Mode**	**Manual**: I/O items are manually added in the **Device Editor**. The I/O items list is not affected by changes to the device DTM.
		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**.
	Reimport Items	Press this button to import 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**.

Click **Apply** to save your edits and leave the window open for further edits.

Address Setting Tab

Configure the **Address Setting** page to perform these tasks:

- Configure the IP address for a device.
- Enable or disable DHCP client software for a device.

NOTE: When the DHCP client software is enabled in a Modbus device, it obtains its IP address from the DHCP server in the Ethernet communication module.

In the **Address Setting** page, edit these parameters to conform to your application's design and functionality:

Field	Parameter	Description
Change Address	**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.

Field	Parameter	Description
Address Server	**DHCP for this Device**	**Enabled**: Activate 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 *(see page 97)*.
		Disabled (default): Deactivates the DHCP client in this device.
		NOTE: For this example, select **Enabled**.
	Identified by	If **DHCP for this Device** is **Enabled**, it indicates the device identifier type: ● **MAC Address** ● **Device Name** **NOTE:** For this example, select **Device Name**.
	Identifier	If **DHCP for this Device** is **Enabled**, it indicates the specific device MAC Address or Name value.
	Subnet Mask	The device subnet mask.
	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.

Click **Apply** to save your edits, and leave the window open for further edits.

Request Setting Tab

Configure the **Request Setting** tab to add, configure, and remove Modbus requests for the Modbus device. Each request represents a separate link between the communication module and the Modbus device.

NOTE: The **Request Setting** tab is available only when a Modbus TCP device is selected in the **Device List**.

Create a request:

Step	Action
1	Press the **Add Request** button to see a new request in the table. Press the **Add Request** button: ● The new request appears in the table. ● The corresponding request items appear in the **Device List**. **NOTE:** The **Add Request** function is enabled only when **Import Mode** on the **Properties** tab is set to **Manual**.
2	Configure the request settings according to the table below.
3	Repeat these steps to create additional requests.
4	Press the **Apply** to save the request.

When you create a request, these **Request Settings** parameters are available:

Setting	Description
Connection Bit	This bit indicates the read-only offset for the health bit for this connection. Offset values (starting at 0) are auto-generated by the Control Expert DTM based on the connection type.
Unit ID	The Unit ID is the number used to identify the target of the connection. **NOTE:** Consult the manufacturer's user manual for the specific target device to find its Unit ID.
Health Time Out (ms)	This value represents the maximum allowed interval between device responses before a time out is detected: ● valid range: 5 ... 65535 ms ● interval: 5 ms ● default: 1500 ms
Repetitive Rate (ms)	This value represents the data scan rate in intervals of 5 ms. (The valid range is 0...60000 ms. The default is 60 ms.)
RD Address	Data that is read from the remote device at this address is stored in the input data image of the Ethernet communication module.
RD Length	This value represents the number of words (0...125) in the Modbus device that the communication module reads.
Last Value	This value represents the behavior of input data in the application if communications are lost: ● **Hold Value** (default) ● **Set To Zero**
WR Address	The output data image in the Ethernet communication module's data structure is written to this address in the remote Modbus device.
WR Length	This value represents the number of words (0...120) in the Modbus device to which the communication module writes.

Remove a request:

Step	Action
1	Click a row in the table.
2	Press the **Remove** button to remove the request. **NOTE:** The corresponding request items disappear from the **Device List**.
3	Press the **Apply** to save the configuration.

The next step is to connect the Control Expert project to the Modbus device.

Section 5.6
Logging DTM Events to a Control Expert Logging Screen

Logging DTM Events to a Control Expert Logging Screen

Introduction

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 Page

In Control Expert:

Step	Action
1	Open a project that includes a BMENOC0321 Ethernet communications module *(see page 52)*.
2	Open the **DTM Browser** (**Tools → DTM Browser**).
3	In the **DTM Browser**, find the name that you assigned to the BMENOC0321 module *(see page 53)*.
4	Double-click the name of the BMENOC0321 (or right-click **Open**) to open the configuration window.
5	Select **Logging** in the navigation tree.

Section 5.7
Logging DTM and Module Events to the SYSLOG Server

Logging DTM and Module Events to the SYSLOG Server

Configuring the SYSLOG Server

The M580 CPU sends SYSLOG events to the SYSLOG server.

Configure the SYSLOG server address for logging DTM and module events:

Step	Action
1	In Control Expert, select **Tools → Project Settings**.
2	In the left pane of the **Project Settings** window, select **Project Settings → General → PLC diagnostics**.
3	In the right pane: ● Select the **PLC event logging** check box. ● In the **SYSLOG server address** field enter the IP address of the SYSLOG server. ● In the **SYSLOG server port number** field, enter the port number. **NOTE:** The SYSLOG server protocol is not configurable, and is set to **tcp** by default.

NOTE: Refer to the *Modicon Controllers Platform Cyber Security Reference Manual* for information on setting up a SYSLOG server in your system architecture *(see page 146)*.

DTM Events Logged to the SYSLOG Server

These DTM events are logged to the SYSLOG server:

- Configuration parameter change
- Adding a device
- Deleting a device
- Switching to **Advanced Mode**
- A **Rebuild All Project** command
- A **Build Changes** command
- Renaming of I/O variables
- Adding tasks
- Modifying tasks

BMENOC0321 Module Events Logged to the SYSLOG Server

The following BMENOC0321 module events are logged to the SYSLOG server:

- TCP connection denied due to **Access Control** list
- Enabling/Disabling communication services outside configuration
- Ethernet port link up/down events
- RSTP topology change
- Configuration download of COM services
- Program operating mode change of COMs (Run, Stop, Init)
- FTP login successful or denied

Chapter 6
Explicit Messaging

What Is in This Chapter?

This chapter contains the following sections:

Section 6.1
Introduction to Explicit Messaging

About Explicit Messaging

Overview

The BMENOC0321 control network module supports explicit messaging through the EtherNet/IP and Modbus TCP protocols:

- *EtherNet/IP*: Use the `DATA_EXCH` function block in application logic to create an EtherNet/IP explicit message.
- *Modbus TCP*: Use the `DATA_EXCH` function block or `WRITE_VAR` and `READ_VAR` function blocks in application logic to create a Modbus TCP explicit message.

NOTE: A single Control Expert application can contain more than 16 explicit messaging blocks, but only 16 explicit messaging blocks can be active at the same time.

This chapter describes how to configure both EtherNet/IP and Modbus TCP explicit messages through these mechanisms:

- `DATA_EXCH` function block (in application logic)
- Control Expert graphical interface

Section 6.2
Explicit Messaging Using the DATA_EXCH Block

Overview

Use this overview of the DATA_EXCH function block to configure both EtherNet/IP and Modbus TCP explicit messages.

These instructions describe the configuration of the DATA_EXCH function block's management parameter, which is common to both Modbus TCP and EtherNet/IP explicit messaging.

In a Hot Standby system, the primary BMENOC0321 control network module sends the explicit message. Even when a switchover occurs and the primary becomes the standby, the module can run the active sections.

What Is in This Section?

This section contains the following topics:

Configuring Explicit Messaging Using DATA_EXCH

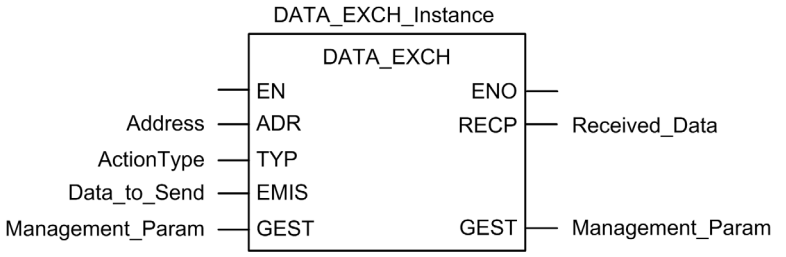
Overview

Use the DATA_EXCH function block to configure both Modbus TCP explicit messages and connected and unconnected EtherNet/IP explicit messages.

The Management_Param, the Data_to_Send, and the Received_Data parameters define the operation.

EN and ENO can be configured as additional parameters.

FBD Representation



Input Parameters

Parameter	Data type	Description
EN	BOOL	This parameter is optional. When this input is set to one, the block is activated and can solve the function blocks algorithm. When this input is set to zero, the block is deactivated and won't solve the function block algorithm.
Address	Array [0...7] of INT	The path to the destination device, the content of which can vary depending on the message protocol. Use the Address function as an is input to the block parameter ADR.. Refer to a description of the Address parameter for: ● EtherNet/IP messages *(see page 159)* ● Modbus TCP messages *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)*

Parameter	Data type	Description
ActionType	INT	The type of action to perform. For both the EtherNet/IP and Modbus TCP protocols, this setting = 1 (transmission followed by await reception).
Data_to_Send	Array [n...m] of INT	The content of this parameter is specific to the protocol, either EtherNet/IP or Modbus TCP. For EtherNet/IP explicit messaging, refer to the topic Configuring the Data_To_Send Parameter *(see page 159)*. For Modbus TCP explicit messaging, refer to Control Expert online help.

Input/Output Parameters

The Management_Param array is local:

Parameter	Data type	Description
Management_Param	Array [0...3] of INT	The management parameter *(see page 154)*, consisting of four words.

Do not copy this array during a switchover from a primary to a standby CPU in a Hot Standby system. Uncheck the **Exchange On STBY** variable in Control Expert when you configure a Hot Standby system.

NOTE: Refer to the description of Hot Standby system data management and the T_M_ECPU_HSBY DDT *(see Modicon M580 Hot Standby, System Planning Guide for, Frequently Used Architectures)* in the M580 Hot Standby System Planning Guide *(see Modicon M580 Hot Standby, System Planning Guide for, Frequently Used Architectures)*.

Output Parameters

Parameter	Data type	Description
ENO	BOOL	This parameter is optional. When you select this output you also get the EN input. ENO output is activated upon successful execution of the function block.
Received_Data	Array [n...m] of INT	The EtherNet/IP (CIP) response *(see page 160)* or the Modbus TCP response *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)*. The structure and content depends upon the specific protocol.

Configuring the DATA_EXCH Management Parameter

Introduction

The structure and content of the management parameter of the `DATA_EXCH` block is common to both EtherNet/IP and Modbus TCP explicit messaging.

Configuring the Management Parameter

The management parameter consists of four contiguous words:

Data source	Register	Description	
		High Byte (MSB)	Low Byte (LSB)
Data managed by the system	`Management_Param[0]`	Exchange number	Two read-only bits: ● Bit 0 = Activity bit *(see page 155)* ● Bit 1 = Cancel bit
	`Management_Param[1]`	Operation report *(see page 386)*	Communication report *(see page 385)*
Data managed by the user	`Management_Param[2]`	Block timeout. Values include: ● 0 = infinite wait ● other values = timeout x 100 ms, for example: ❍ 1 = 100 ms ❍ 2 = 200 ms	
	`Management_Param[3]`	Length of data sent or received: ● Input (before sending the request): length of data in the `Data_to_Send` parameter, in bytes ● Output (after response): length of data in the `Received_Data` parameter, in bytes	

Activity Bit

The activity bit is the first bit of the first element in the table. The value of this bit indicates the execution status of the communication function:

- **1**: The bit is set to 1 when the function launches.
- **0**: The bit returns to 0 upon the completion of the execution. (The transition from 1 to 0 increments the exchange number. If an error is detected during the execution, search for the corresponding code in the operation and communication report *(see page 385)*.)

For example, you can make this declaration in the management table:

```
Management_Param[0] ARRAY [0..3] OF INT
```

For that declaration, the activity bit corresponds to this notation:

```
Management_Param[0].0
```

NOTE: The notation previously used requires configuration of the project properties in such a way as to authorize the extraction of bits on integer types. If this is not the case, `Management_Param[0].0` cannot be accessed in this manner.

Section 6.3
EtherNet/IP Explicit Messaging Using DATA_EXCH

Overview

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

What Is in This Section?

This section contains the following topics:

Explicit Messaging Services

Overview

Every explicit message performs a service. Each service is associated with a service code. Identify the explicit messaging service by its name, decimal number, or hexadecimal number.

You can execute explicit messages using the `DATA_EXCH` function block in the Control Expert DTM.

Services

The services available in Control Expert include, but are not limited to, these service codes:

Service Code		Description	Available in...	
Hex	Dec		DATA_EXCH 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	—
B-C	11-12	(*Reserved*)	—	—
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
14	20	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
"X" indicates the service is available. "—" indicates the service is not available.				

Service Code		Description	Available in...	
Hex	Dec		DATA_EXCH block	Control Expert GUI
1C	28	GroupSync	X	—
1D-31	29-49	(*Reserved*)	—	—
"X" indicates the service is available. "—" indicates the service is not available.				

Configuring EtherNet/IP Explicit Messaging Using DATA_EXCH

Configuring the Address Parameter

To configure the Address parameter, use the `ADDM` function to convert the character string, described below, to an address that is input into the ADR parameter of the DATA_EXCH block:

ADDM('rack.slot.channel{ip_address}message_type.protocol'), where:

This field...	Represents...
rack	the number assigned to the rack containing the communication module
slot	the position of the communication module in the rack
channel	the communication channel—set to a value of **0**
ip_address	the IP address of the remote device, for example 193.168.1.6
message_type	the type of message, presented as a three character string—either: • **UNC** (indicating an unconnected message), or • **CON** (indicating a connected message)
protocol	the protocol type—the three character string **CIP**

Configuring the Data_to_Send Parameter

The `Data_to_Send` parameter varies in size. It consists of contiguous registers that include—in sequence—both the message type and the CIP request:

Offset (words)	Length (bytes)	Data Type	Description
0	2 bytes	Bytes	Message type: • High byte = size of the request in words • Low byte = EtherNet/IP service code
1	`Management_Param[3]` (size of `Data_to_Send`) minus 2	Bytes	The CIP request[1]. **NOTE:** The structure and size of the CIP request depends on the EtherNet/IP service.
1 Structure the CIP request in little endian order.			

Contents of the Received_Data Parameter

The `Received_Data` parameter contains only the CIP response. The length of the CIP response varies, and is reported by `Management_Param[3]` after the response is received. The format of the CIP response is described, below:

Offset (words)	Length (bytes)	Data Type	Description
0	2	Byte	• High byte (MSB) = reserved • Low byte (LSB): reply service
1	2	Byte	• High byte (MSB): length of additional status • Low byte (LSB): EtherNet/IP general status *(see Modicon M340, BMX NOC 0401 Ethernet Communication Module, User Manual)*
2	length of additional status	Byte array	Additional Status[1]
...	`Management_Param[3]` (size of `Received_Data`) minus 4, and minus the additional status length	Byte array	Response data

1. Refer to *The CIP Networks Library, Volume 1, Common Industrial Protocol* at section 3-5.6 *Connection Manager Object Instance Error Codes*.

NOTE: The response is structured in little endian order.

Checking the Received_Data Response for System and CIP Status

Use the contents of the Received_Data parameter to check both the system status and the CIP status of the Ethernet communication module when handling the explicit message.

First: Check the value of the high byte (MSB) of the first response word, positioned at offset 0. If the value of this byte is:

- ❍ equal to 0: the system properly handled the explicit message
- ❍ not equal to 0: a system-based event occurred
  Refer to the list of EtherNet/IP Explicit Messaging Event Codes *(see page 382)* for an explanation of the system-based event code contained in the second response word, positioned at offset 1.

Next: If the system properly handled the explicit message, and the high byte of the first response word equals 0, check the value of the second response word, positioned at offset 1. If the value of this word is:

- ❍ equal to 0: the explicit message was properly handled by the CIP protocol
- ❍ not equal to 0: a CIP protocol-based event occurred
  Refer to your CIP documentation for an explanation of the CIP status displayed in this word.

EtherNet/IP Explicit Message Example: Get_Attribute_Single

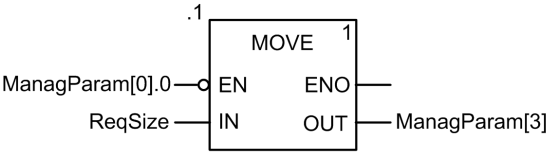
Overview

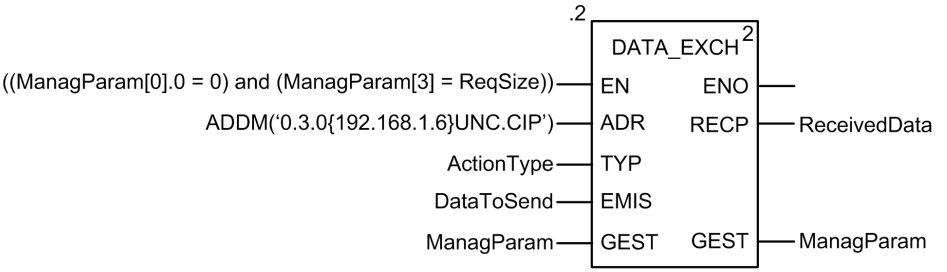
The following unconnected explicit messaging example shows you how to use the DATA_EXCH function block to retrieve diagnostic data from a remote device (at IP address 192.168.1.6). This example is executing a Get_Attribute_Single of assembly instance 100, attribute 3.

You can perform the same explicit messaging service using the **EtherNet/IP Explicit Message** window *(see page 179)*.

Implementing the DATA_EXCH Function Block

To implement the DATA_EXCH function block, create and assign variables for the following blocks:





Configuring the Address Variable

The Address variable identifies the explicit message originating device (in this example, the communication module) and the target device. Note that the Address variable does not include the Xway address elements {Network.Station} because we are not bridging through another PLC station. As an example, use the `ADDM` function to convert the following character string to an address:

`ADDM`('0.1.0{192.168.1.6}UNC.CIP'), where:

- rack = 0
- module (slot number) = 1
- channel = 0
- remote device IP address = 192.168.1.6
- message type = unconnected
- protocol = CIP

Configuring the ActionType Variable

The ActionType variable identifies the function type for the `DATA_EXCH` function block:

Variable	Description	Value (hex)
ActionType	Transmission followed by wait for response	16#01

Configuring the DataToSend Variable

The DataToSend variable identifies the details of the CIP explicit message request:

Variable	Description	Value (hex)
DataToSend[0]	CIP request service information: • High byte = request size in words: 16#03 (3 decimal) • Low byte = service code: 16#0E (14 decimal)	16#030E
DataToSend[1]	CIP request class information: • High byte = class: 16#04 (4 decimal) • Low byte = class segment: 16#20 (32 decimal)	16#0420
DataToSend[2]	CIP request instance information: • High byte = instance: 16#64 (100 decimal) • Low byte = instance segment: 16#24 (36 decimal)	16#6424
DataToSend[3]	CIP request attribute information: • High byte = attribute: 16#03 (3 decimal) • Low byte = attribute segment: 16#30 (48 decimal)	16#0330

Viewing the Response

Use a Control Expert Animation table to display the ReceivedData variable array. Note that the ReceivedData variable array consists of the entire data buffer.

To display the CIP response, follow these steps:

Step	Action	
1	In Control Expert, select **Tools** → **Project Browser** to open the Project Browser.	
2	In the Project Browser, select the **Animation Tables** folder, then click the right mouse button. A pop-up menu appears.	
3	Select **New Animation Table** in the pop-up menu. A new animation table and its properties dialog both open.	
4	In the Properties dialog, edit the following values:	
	Name	Type in a table name. For this example: **ReceivedData**.
	Functional module	Accept the default **<None>**.
	Comment	(Optional) Type your comment here.
	Number of animated characters	Type in **100**, representing the size of the data buffer in words.
5	Click **OK** to close the dialog.	
6	In the animation table's **Name** column, type the name of the variable assigned to the RECP pin: **ReceivedData** and press **Enter**. The animation table displays the ReceivedData variable.	
7	Expand the ReceivedData variable to display its word array, where you can view the CIP response contained in the ReceivedData variable. **NOTE:** Each array entry presents 2 bytes of data in little endian format, where the least significant byte is stored in the smallest memory address. For example, '8E' in word[0] is the lower byte, and '00' is the upper byte.	

EtherNet/IP Explicit Message Example: Read Modbus Object

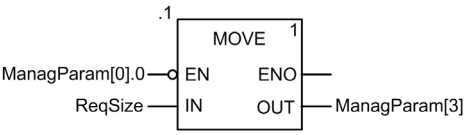
Overview

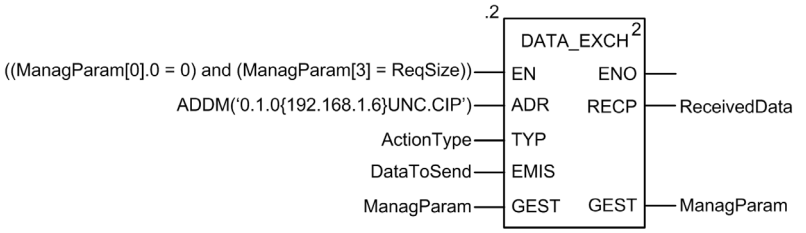
The following unconnected explicit messaging example shows you how to use the DATA_EXCH function block to read data from a remote device.

You can perform the same explicit messaging service using the **EtherNet/IP Explicit Message** window *(see page 179)*.

Implementing the DATA_EXCH Function Block

To implement the DATA_EXCH function block, create and assign variables for the following blocks:





Configuring the Address Variable

The Address variable identifies the explicit message originating device (in this example, the Ethernet communication module) and the target device. Note that the Address variable does not include the Xway address elements {Network.Station} because we are not bridging through another PLC station. Use the ADDM function to convert the following character string to an address:

ADDM('0.1.0{192.168.1.6}UNC.CIP'), where:

- rack = 0
- module (slot number) = 1
- channel = 0
- remote device IP address = 192.168.1.6
- message type = unconnected
- protocol = CIP

Configuring the ActionType Variable

The ActionType variable identifies the function type for the DATA_EXCH function block:

Variable	Description	Value (hex)
ActionType	Transmission followed by wait for response	16#01

Configuring the DataToSend Variable

The DataToSend variable identifies the type of explicit message and the CIP request:

Variable	Description	Value (hex)
DataToSend[0]	CIP request service information: • High byte = request size in words: 16#02 (2 decimal) • Low byte = service code: 16#4E (78 decimal)	16#024E
DataToSend[1]	CIP request class information: • High byte = class: 16#44 (68 decimal) • Low byte = class segment: 16#20 (32 decimal)	16#4420
DataToSend[2]	CIP request instance information: • High byte = instance: 16#01 (1 decimal) • Low byte = instance segment: 16#24 (36 decimal)	16#0124
DataToSend[3]	Location of first word to be read): • High byte = 16#00 (0 decimal) • Low byte = 16#31 (49 decimal)	16#0031
DataToSend[4]	Number of words to read: • High byte = attribute: 16#00 (0 decimal) • Low byte = attribute segment: 16#01 (1 decimal)	16#0001

Viewing the Response

Use a Control Expert Animation table to display the ReceivedData variable array. Note that the ReceivedData variable array consists of the entire data buffer.

To display the CIP response, follow these steps:

Step	Action
1	In Control Expert, select **Tools** → **Project Browser** to open the Project Browser.
2	In the Project Browser, select the **Animation Tables** folder, then click the right mouse button. A pop-up menu appears.
3	Select **New Animation Table** in the pop-up menu. A new animation table and its properties dialog both open.

Step	Action	
4	In the Properties dialog, edit the following values:	
	Name	Type in a table name. For this example: **ReceivedData**.
	Functional module	Accept the default **<None>**.
	Comment	(Optional) Type your comment here.
	Number of animated characters	Type in **49**, representing the size of the data buffer in words.
5	Click **OK** to close the dialog.	
6	In the animation table's **Name** column, type in the name of the variable assigned to the RECP pin: **ReceivedData** and press **Enter**. **Result**: The animation table displays the ReceivedData variable.	
7	Expand the ReceivedData variable to display its word array, where you can view the CIP response contained in the ReceivedData variable. **NOTE:** Each array entry presents 2 bytes of data in little endian format, where the least significant byte is stored in the smallest memory address. For example, 'CE' in word[0] is the lower byte, and '00' is the upper byte.	

EtherNet/IP Explicit Message Example: Write Modbus Object

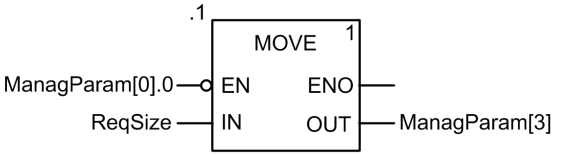
Overview

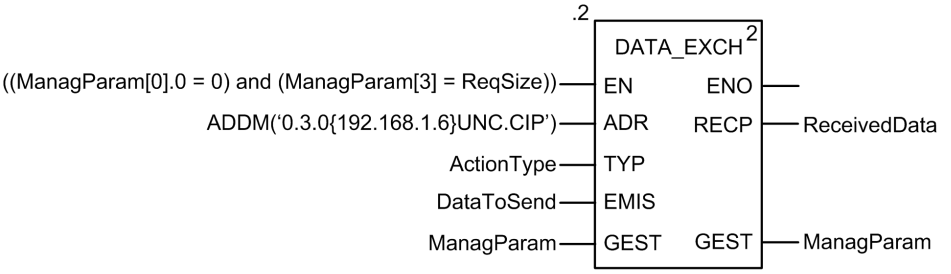
The following unconnected explicit messaging example shows you how to use the DATA_EXCH function block to write data to a remote device using the Write_Holding_Registers service of the Modbus object.

You can perform the same explicit messaging service using the **EtherNet/IP Explicit Message** window *(see page 179)* in the Control Expert DTM.

Implementing the DATA_EXCH Function Block

To implement the DATA_EXCH function block, create and assign variables for the following blocks:





Configuring the Address Variable

The Address variable identifies the explicit message originating device (in this example, the communication module) and the target device. Note that the Address variable does not include the Xway address elements {Network.Station} because we are not bridging through another PLC station. Use the `ADDM` function to convert the following character string to an address:

`ADDM`('0.1.0{192.168.1.6}UNC.CIP'), where:

- rack = 0
- module (slot number) = 1
- channel = 0
- remote device IP address = 192.168.1.6
- message type = unconnected
- protocol = CIP

Configuring the ActionType Variable

The ActionType variable identifies the function type for the `DATA_EXCH` function block:

Variable	Description	Value (hex)
ActionType	Transmission followed by wait for response	16#01

Configuring the DataToSend Variable

The DataToSend variable identifies the type of explicit message and the CIP request:

Variable	Description	Value (hex)
DataToSend[0]	CIP request service information: ● High byte = request size in words: 16#02 (2 decimal) ● Low byte = service code: 16#50 (80 decimal)	16#0250
DataToSend[1]	CIP request class information: ● High byte = class: 16#44 (68 decimal) ● Low byte = class segment: 16#20 (32 decimal)	16#4420
DataToSend[2]	CIP request instance information: ● High byte = instance: 16#01 (1 decimal) ● Low byte = instance segment: 16#24 (36 decimal)	16#0124
DataToSend[3]	Location of first word to write (+ %MW1): ● High byte = 16#00 (0 decimal) ● Low byte = 16#00 (0 decimal)	16#0000
DataToSend[4]	Number of words to write: ● High byte = attribute: 16#00 (0 decimal) ● Low byte = attribute segment: 16#01 (1 decimal)	16#0001
DataToSend[5]	Data to write: ● High byte = attribute: 16#00 (0 decimal) ● Low byte = attribute segment: 16#6F (111 decimal)	16#006F

Viewing the Response

Use a Control Expert Animation table to display the ReceivedData variable array. Note that the ReceivedData variable array consists of the entire data buffer.

To display the CIP response, follow these steps:

Step	Action	
1	In Control Expert, select **Tools** → **Project Browser** to open the Project Browser.	
2	In the Project Browser, select the **Animation Tables** folder, then click the right mouse button. A pop-up menu appears.	
3	Select **New Animation Table** in the pop-up menu. A new animation table and its properties dialog both open.	
4	In the Properties dialog, edit the following values:	
	Name	Type in a table name. For this example: **ReceivedData**.
	Functional module	Accept the default **<None>**.
	Comment	(Optional) Type your comment here.
	Number of animated characters	Type in **49**, representing the size of the data buffer in words.
5	Click **OK** to close the dialog.	
6	In the animation table's **Name** column, type in the name of the variable assigned to the RECP pin: **ReceivedData**, and press **Enter**. **Result**: The animation table displays the ReceivedData variable.	
7	Expand the ReceivedData variable to display its word array, where you can view the CIP response contained in the ReceivedData variable. **NOTE:** Each array entry presents 2 bytes of data in little endian format, where the least significant byte is stored in the smallest memory address. For example, 'D0' in word[0] is the lower byte, and '00' is the upper byte.	

Section 6.4
Modbus TCP Explicit Messaging Using DATA_EXCH

Overview

This section shows you how to configure `DATA_EXCH` function block parameters for Modbus TCP explicit messages.

What Is in This Section?

This section contains the following topics:

Modbus TCP Explicit Messaging Function Codes

Overview

You can execute Modbus TCP explicit messages using either a Control Expert `DATA_EXCH` function block or the Modbus Explicit Message Window.

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

Function Codes

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

Function Code (dec)	Description
1	Read bits (%M)
2	Read input bits (%I)
3	Read words (%MW)
4	Read input words (%IW)
15	Write bits (%M)
16	Write words (%MW)

NOTE: You can use the `DATA_EXCH` function block to execute any Modbus function, via program logic. Because the available function codes are too numerous to list here, refer instead to the Modbus IDA website for more information about these Modbus functions, at *http://www.Modbus.org*.

Configuring Modbus TCP Explicit Messaging Using DATA_EXCH

Introduction

When you use the DATA_EXCH block to create an explicit message for a Modbus TCP device, configure this block the same way you would configure it for any other Modbus communication. Refer to the Control Expert online help for instructions on how to configure the DATA_EXCH block.

Configuring ADDM Block Unit ID Settings

When you configure the DATA_EXCH block, use the ADDM block to set the DATA_EXCH block's Address parameter. The ADDM block presents the configuration format ADDM('rack.slot.channel[ip_address]UnitID.message_type.protocol') where:

Parameter	Description
rack	the number assigned to the rack containing the communication module
slot	the position of the communication module in the rack
channel	the communication channel (set to a value of 0)
ip_address	the IP address of the remote device (for example, 192.168.1.7)
Unit ID	the destination node address, also known as the Modbus Plus on Ethernet Transporter (MET) mapping index value
message_type	the three-character string **TCP**
protocol	the three-character string **MBS**

The Unit ID value in a Modbus message indicates the destination of the message.

Refer to the Modbus diagnostic codes *(see M580 IEC 61850, BMENOP0300 Module, Installation and Configuration Guide)*.

Contents of the Received_Data Parameter

The `Received_Data` parameter contains the Modbus response. The length of the response varies, and is reported by `Management_Param[3]` after the response is received. The format of the Modbus response is described, below:

Offset (words)	Length (bytes)	Description
0	2	First word of the Modbus response: ● High byte (MSB): ❍ if successful: Modbus Function Code ❍ if not: Modbus function code + 16#80 ● Low byte (LSB): ❍ if successful: depends on the request ❍ if not: Modbus exception code
1	Length of the `Received_Data` parameter – 2	Remainder of the Modbus response: depends on the specific Modbus request)

NOTE:

- Structure the response in little endian order.
- In some cases of detected errors, Received_Data is also used to judge the type of detected error along with Management_Param.

Modbus TCP Explicit Message Example: Read Register Request

Introduction

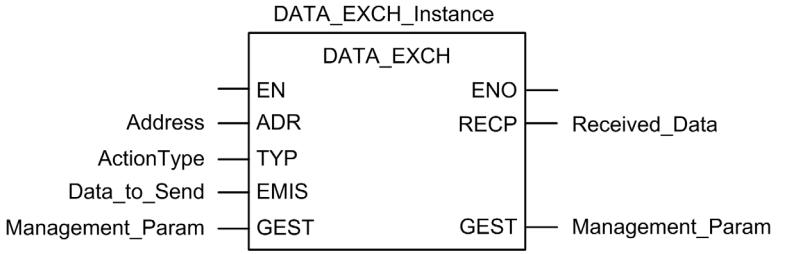
Use the DATA_EXCH function block to send a Modbus TCP explicit message to a remote device at a specific IP address to read a single word located in the remote device.

The Management_Param, the Data_to_Send, and the Received_Data parameters define the operation.

EN and ENO can be configured as additional parameters.

Implementing the DATA_EXCH Function Block

To implement the DATA_EXCH function block, create and assign variables for the for following:



Configuring the Address Variable

The Address variable identifies the explicit message originating device and the target device. Note that the Address variable does not include the Xway address elements {Network.Station} because you are not bridging through another PAC station. Use the ADDM function to convert the following character string to an address:

ADDM('0.1.0{192.168.1.7}TCP.MBS'), where:

- rack = 0
- module (slot number) = 1
- channel = 0
- remote device IP address = 192.168.1.7
- message type = TCP
- protocol = Modbus

Configuring the ActionType Variable

The ActionType variable identifies the function type for the DATA_EXCH function block:

Variable	Description	Value (hex)
ActionType	Transmission followed by wait for response	16#01

Configuring the DataToSend Variable

The DataToSend variable contains the target register address and the number of registers to read:

Variable	Description	Value (hex)
DataToSend[0]	• High byte = Most significant byte (MSB) of register address 16#15 (21 decimal) • Low byte = function code: 16#03 (03 decimal)	16#1503
DataToSend[1]	• High byte = Most significant byte (MSB) of the number of registers to read: 16#00 (0 decimal) • Low byte = Least significant byte (LSB) of register address: 16#0F (15 decimal)	16#000F
DataToSend[2]	CIP request instance information: • High byte = not used: 16#00 (0 decimal) • Low byte = Least significant byte (LSB) of the number of registers to read: 16#01 (1 decimal)	16#0001

NOTE: For detailed information about M580 network topologies, refer to the *Modicon M580 Standalone System Planning Guide for Frequently Used Architectures* and *Modicon M580 System Planning Guide for Complex Topologies*.

Viewing the Response

Use a Control Expert Animation table to display the ReceivedData variable array. Note that the ReceivedData variable array consists of the entire data buffer.

To display the Modbus TCP response, follow these steps:

Step	Action	
1	In Control Expert, select **Tools → Project Browser**.	
2	In the Project Browser, select the **Animation Tables** folder, and click the right mouse button. **Result**: A pop-up menu appears.	
3	Select **New Animation Table** in the pop-up menu. **Result**: A new animation table and its properties dialog open.	
4	In the Properties dialog, edit the following values:	
	Name	Type in a table name. For this example: **ReceivedData**.
	Functional module	Accept the default **<None>**.
	Comment	(Optional) Type your comment here.
	Number of animated characters	Type in **100**, representing the size of the data buffer in words.
5	Click **OK** to close the dialog.	

Step	Action
6	In the animation table's **Name** column, type in the name of the variable assigned to the databuffer: **ReceivedData** and press **Enter**. **Result**: The animation table displays the ReceivedData variable.
7	Expand the ReceivedData variable to display its word array, where you can view the CIP response contained in the ReceivedData variable. **NOTE:** Each array entry presents 2 bytes of data in little endian format. For example, '03' in word[0] is the low byte, and '02' is the high byte.

Section 6.5
Explicit Messaging via the Control Expert GUI

What Is in This Section?

This section contains the following topics:

Before You Begin

Introduction

Use the Modbus Explicit Message window in the Control Expert DTM *(see page 181)* to send an explicit message to a Modbus TCP module or distributed device on the network. You can use explicit messaging to perform many different services. Not every Modbus TCP device supports every service.

Connect the DTM

Before you can configure explicit messaging for EtherNet/IP or Modbus TCP devices, make the connection between the DTM for the target communication module and the physical module:

Step	Action
1	In the **DTM Browser**, find the name that is assigned to your Ethernet communications module.
2	Right-click on the module name.
3	Scroll to **Connect**.

Sending Explicit Messages to EtherNet/IP Devices

Overview

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

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

- *unconnected*: With unconnected messaging, a CIP connection to the destination is not established before the point-to-point transfer of data.
- *connected*: With connected messaging, node resources are reserved in advance of the data transfer and are dedicated and always available.

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

The EtherNet/IP explicit message configuration window presents an example of both the configuration of an EtherNet/IP explicit message and the response. The explicit message is addressed to a distributed module to obtain diagnostic information.

Sending Explicit Messages

Execute an EtherNet/IP explicit message:

Step	Action	
1	In the **DTM Browser**, select the communication module that is upstream of the target device.	
2	Right-click the module and select **Device menu → Additional functions → EtherNet/IP Explicit Message**.	
3	Configure explicit messages in these fields:	
	IP Address	The IP address of the target device identifies the target of the explicit message.
	Class	The class identifier of the target device is used to construct the message path. It is an integer value (1...65535).
	Instance	The class instance of the target device is used to construct the message path. It is an integer value (1...65535).
	Attribute	(Optional) The specific device attribute (or property) is the target of the explicit message that is used to construct the message path. It is an integer value (1...65535). **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 (1...127) associated with the service to be performed by the explicit message. **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.
	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. **NOTE:** Displayed only when **Advanced Mode** is enabled.
	Data	The data to be sent to the target device, for services that send data.
	Messaging	Select the type of explicit message to send: • **Connected** • **Unconnected**
	Repeat 500 ms	Select this check box to re-send the explicit message every 500 ms.
4	After your explicit message is configured, click **Send to Device**. Data in the **Response (hex)** area was sent to the configuration tool by the target device in hexadecimal format. Messages in the **Status** area indicate 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 in the Control Expert DTM to send an explicit message from an EtherNet/IP module or distributed device on the network.

You can use explicit messaging to perform many different services. Not every Modbus TCP device supports every service.

The Modbus TCP explicit message configuration window shows both the configuration of a Modbus TCP explicit message and the response.

Sending Explicit Messages

Execute a Modbus TCP explicit message:

Step	Action	
1	In the **DTM Browser**, select the communication module that is upstream of the target device.	
2	Right-click the module and select **Device menu → Additional functions → Modbus TCP Explicit Message**.	
3	Configure explicit messages in these fields:	
	IP Address	The IP address of the target device, used to identify the target of the explicit message.
	Start Address	A component of the addressing path.
	Quantity	A component of the addressing path.
	Read Device Id Code	Read-only identification of the service that the explicit message is intended to perform.
	Object Id	(read-only) Specify the object the explicit message is intended to access.
	Refer to your Modbus TCP device user manual for Start Address, Quantity, Read Device Id Code, and Object Id values.	
	Unit Id	The Unit ID is the number used to identify the target of the connection. **NOTE:** Consult the manufacturer's user manual for the specific target device to find its Unit ID.
	Number	The read-only integer (0 ... 255) associated with the service to be performed by the explicit message.
	Name	Select the service the explicit message is intended to perform.
	Repeat 500ms	Select this check box to re-send the explicit message every 500 ms. Leave this check box de-selected.

Step	Action
4	After your explicit message is configured, click **Send to Device**. Data in the **Response** area was sent to the configuration tool by the target device in hexadecimal format. Messages in the **Status** area indicate whether or not the explicit message has succeeded.
5	Click **Close** to close the window.

Chapter 7
Diagnosing the BMENOC0321 Module

Overview

This chapter describes the diagnostics for the BMENOC0321 module.

NOTE: For details on diagnostics at the system level, refer to the systems diagnostics topic in the Modicon M580 *System Planning Guide*.

What Is in This Chapter?

This chapter contains the following sections:

Section 7.1
LED Indicators

Visual Indicators on the BMENOC0321 Module

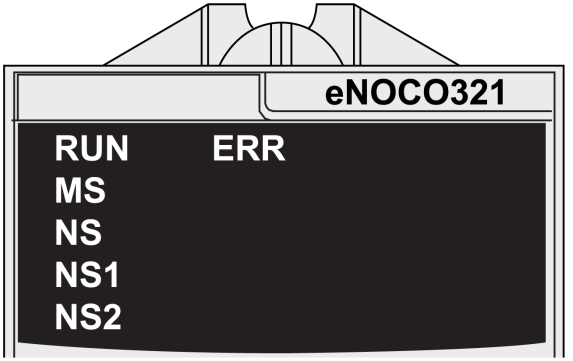
Introduction

There are two sets of LED indicators on the front of the BMENOC0321 control network module:

- LEDs that report the performance of the module and its communications with the network appear as words (or abbreviations) at the top of the module.
- Small LEDs that report the status of activity and connectivity of the Ethernet ports are next to each RJ45 connector on the front of the module.

LED Indications

This is the LED display on the front of the BMENOC0321 module:



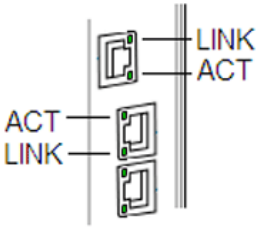
NOTICE
UNINTENTIONAL EQUIPMENT BEHAVIOR Confirm that each module has a unique IP address. Duplicate IP addresses can cause unpredictable module/network behavior. **Failure to follow these instructions can result in equipment damage.**

This table describes the LEDs:

LED	Color	State	Description
RUN	green	on	The module is configured.
		off	There is no power to the module, or the module is not configured.
		blinking	The module is in power-up test or in OS update.
ERR	red	on	A detected error that is **not** an X Bus communication error.
		off	There is no power to the module, no errors are detected, or the module is in OS update.
		blinking	The module is not configured.
			An X Bus communication error is detected.
MS (module status)	–	off	There is no power to the module.
	green	on	The module is operating correctly.
		blinking	The module is not configured.
	red	on	A major non-recoverable error (firmware error as an example) is detected.
		blinking	A recoverable error is detected or the BMENOC0321 has a duplicate IP address.
NS, **NS1**, **NS2** (network status) (See the note below.)	–	off	There is no power to the module or there is no link on all external ports, including the Ethernet backplane port.
	green	on	At least one CIP connection for which the BMENOC0321 module is the originator has been established.
		blinking	The module has an IP address, but there is no CIP connection.
	red	on	The module has a duplicate IP address, or the module is in OS update.
		blinking	At least one exclusive owner CIP connection (for which the BMENOC0321 is the target) is timed out. The LED blinks until the connection is reestablished or the module is reset.
NOTE: The **NS**, **NS1**, and **NS2** LEDs indicate the network status of their respective subnets.			

Ethernet Port LEDs

There are two LEDs associated with each RJ-45 connector:



These LEDs report the activity and connectivity of the associated Ethernet port:

LED	Color	State	Description
LINK (link/speed)	green	on	The 1000 Mbps link is detected.
	yellow	on	The 10/100 Mbps link is detected.
	—	off	No link to the port is detected.
ACT (activity)	green	blinking	There is transmit or receive activity on the port.
		on	The link is detected, but there is no activity on the port.
		off	There is no link to the port.

Section 7.2
Device DDT for the BMENOC0321

BMENOC0321 Device DDT

Introduction

Accessing the Device DDT

View the Device DDT variables for the BMENOC0321 control network module:

Step	Action
1	Open a Control Expert project that includes a BMENOC0321 module on the local rack.
2	In the **Project Browser**, expand (+) **Variables and Feedback Instances**.
3	Double-click **Device DDT Variables**.
4	On the **Variables** tab, expand (+) the name that corresponds to your BMENOC0321 module.

Objects

The Ethernet communication module contains two objects:

- input object (object number: 0):
  - ETH_STATUS
  - SERVICE_STATUS
  - SERVICE_STATUS2
  - ETH_PORT_1_2_STATUS
  - ETH_PORT3_BKP_STATUS
  - FIRMWARE_VERSION
  - FDR_USAGE
  - IN_PACKETS
  - IN_ERRORS
  - OUT_PACKETS
  - OUT_ERRORS
  - CONF_SIG
  - LS_HEALTH
  - DIO_HEALTH
- output object (object number: 1)
  - DIO_CTRL

NOTE: This content applies to the T_BMENOC0321_2 DDDT only. This DDDT can’t be localized and it is supported in Unity Pro 11.1 or later.

Input Parameters

The following tables describe the input parameters in the device DDT for the module:

ETH_STATUS: This table describes the bits associated with the ETH_STATUS (word):

Parameter	Type	Bit	Description
PORT1_LINK	BOOL	0	0: Ethernet port 1 (ETH 1) link is down. 1: Ethernet port 1 (ETH 1) link is up.
PORT2_LINK	BOOL	1	0: Ethernet port 2 (ETH 2) link is down. 1: Ethernet port 2 (ETH 2) link is up.
PORT3_LINK	BOOL	2	0: Ethernet port 3 (ETH 3) link is down. 1: Ethernet port 3 (ETH 3) link is up.
ETH_BKP_PORT_LINK	BOOL	3	0: Backplane port link is down. 1: Backplane port link is up.
SCANNER_OK	BOOL	6	0: I/O scanner operations are not normal.
			1: At least one configured device is scanned.
GLOBAL_STATUS	BOOL	7	0: At least one service is not operating normally.
			1: All services are operating normally.
NETWORK_HEALTH	BOOL	8	0: A potential network broadcast storm is detected. **NOTE:** Check your wiring and your CPU and BMENOC0321 configurations.
			1: A network broadcast storm is not detected.

SERVICE_STATUS: This table describes the bits associated with the SERVICE_STATUS (word):

Parameter	Type	Bit	Description
RSTP_SERVICE	BOOL	0	0: The RSTP service is not operating normally.
			1: The RSTP service is operating normally or is disabled.
PORT502_SERVICE	BOOL	2	0: Port 502 is not operating normally.
			1: Port 502 is operating normally or is disabled.
SNMP_SERVICE	BOOL	3	0: SNMP is not operating normally.
			1: SNMP service is operating normally or is disabled.
MAIN_IP_ADDRESS_STATUS	BOOL	4	0: The main IP address is duplicated or not assigned.
			1: The main IP address is unique and valid.
EIP_SCANNER	BOOL	7	0: The EtherNet/IP scanner service is not operating normally.
			1: The EtherNet/IP scanner service is operating normally or is disabled.

Parameter	Type	Bit	Description
MODBUS_SCANNER	BOOL	8	0: The Modbus scanner service is not operating normally.
			1: The Modbus scanner service is operating normally or is disabled.
SNTP_CLIENT	BOOL	10	0: The SNTP client service is not operating normally.
			1: The SNTP client service is operating normally or is disabled.
WEB_SERVER	BOOL	11	0: The web server service is not operating normally.
			1: The web server service is operating normally or is disabled.
FIRMWARE_UPGRADE	BOOL	12	0: The firmware upgrade service is not operating normally.
			1: The firmware upgrade service is operating normally or is disabled.
FTP	BOOL	13	0: The FTP server service is not operating normally.
			1: The FTP server service is operating normally or is disabled.
FDR_SERVER	BOOL	14	0: The FDR server service is not operating normally.
			1: The FDR server service is operating normally or is disabled.
EIP_ADAPTER	BOOL	15	0: The EtherNet/IP adapter service is not operating normally.
			1: The EtherNet/IP adapter service is operating normally or is disabled.

SERVICE_STATUS2: This table describes the parameters associated with the SERVICE_STATUS2 (word):

Parameter	Type	Bit	Description
LLDP_SERVICE	BOOL	1	0: The LLDP service is not operating normally.
			1: The LLDP service is operating normally or is disabled.
EVENT_LOG_STATUS	BOOL	2	0 = Event log service is not operating normally.
			1 = Event log service is operating normally or is disabled.

Parameter	Type	Bit	Description
LOG_SERVER_NOT_REACHABLE	BOOL	3	1 = No acknowledgment received from the syslog server.
			0 = Acknowledgment received from the syslog server
SMTP	BOOL	4	1 = This service is operating normally or is disabled.
			0 = This service is not operating normally.
IP_Forwarding	BOOL	5	1 = This service is operating normally or is disabled.
			0 = This service is not operating normally.
Fieldbus_network_IP_ADDRESS_STATUS	BOOL	6	1 = The main IP address is unique.
			0 = There is a duplicate address (for the main IP address) or there is no address assignment.
Extended_network_IP_ADDRESS_STATUS	BOOL	7	1 = The main IP address is unique.
			0 = There is a duplicate address (for the main IP address) or there is no address assignment.

Other Input Parameters: The scanner device DDT contains these other parameters:

Parameter	Type	Description
ETHERNET_PORT_1_2_STATUS (BYTE)	Bits 1...0	0: ETH 1 disabled
		1: ETH 1 access port
		2: ETH 1 port mirroring
		3: ETH 1 extended network port
	Bits 3...2	reserved (0)
	Bits 5...4	0: ETH 2 disabled
		1: ETH 2 access port
		2: ETH 2 port mirroring
		3: ETH 2 control network port
	Bits 7...6	0: ETH 2 alternate RSTP port
		1: ETH 2 backup RSTP port
		2: ETH 2 designated RSTP port
		3: ETH 2 root RSTP port

Parameter	Type	Description
ETHERNET_PORT3_BKP_STATUS (BYTE)	Bits 1...0	0: ETH 3 disabled
		1: ETH 3 access port
		2: ETH 3 port mirroring
		3: ETH 3 control network port
	Bits 3...2	0: ETH 3 alternate RSTP port
		1: ETH 3 backup RSTP port
		2: ETH 3 designated RSTP port
		3: ETH 3 root RSTP port
	Bits 5...4	0: The Ethernet backplane port is disabled *(see page 88)*.
		3: The Ethernet backplane port is enabled *(see page 88)* to support Ethernet communications.
	Bits 7...6	Reserved (0)
FIRMWARE_VERSION	WORD	MSB = major revision; LSB = minor revision
FDR_USAGE	BYTE	% of FDR sever usage
IN_PACKETS	UINT	Number of packets received by the module
IN_ERRORS	UINT	Number of inbound packets that contain detected errors
OUT_PACKETS	UINT	Number of packets sent from the module
OUT_ERRORS	UINT	Number of packets from the module that contain detected errors
CONF_SIG	UDINT	Signature of all PRM files on the local module FDR server

Device Health Bits

This table describes the health bits for the BMENOC0321 that is scanned by a remote device:

Parameter	Type	Bit	Description
LS_HEALTH	BOOL	0: Local slaves and distributed equipment are not operating normally. 1: Local slaves and distributed equipment are operating normally or are disabled.	local slave health bits (local slave 1 to 12) ARRAY [1...12] of BOOL
DIO_HEALTH	BOOL		distributed equipment health bits (1 bit per distributed device up to 128 devices) ARRAY [0...127] of BOOL

Output Parameters

This table describe the output parameters in the device DDT for the module:

Parameter	Type	Bit	Description
DIO_CTRL	BOOL	0: Enable normal communications to the DIO device. 1: Disable communications to the device. In this case, outputs are not be written and inputs are not updated.	distributed equipment control bits (1 bit per distributed device up to 128 devices) ARRAY [0...127] of BOOL

NOTE: The array index for the DIO device is mapped to the device number in the request/connection summary *(see page 137)* of the BMENOC0321 module's **Device List**.

Section 7.3
Diagnostics through the Control Expert DTM Browser

What Is in This Section?

This section contains the following topics:

Introducing Diagnostics in the Control Expert DTM

Introduction

The Control Expert DTM provides diagnostics information that is collected at configured polling intervals. Use this information to diagnose the operation of your Ethernet communications module.

Connect the DTM

Before you can open the diagnostics page, make the connection between the DTM for the target communication module:

Step	Action
1	Open a Control Expert project that includes the Ethernet communications module.
2	Open the Control Expert **DTM Browser** (**Tools → DTM Browser**).
3	Find the name that is assigned to your Ethernet communications module in the **DTM Browser**.
4	Right-click on the module name.
5	Scroll to **Connect**.

Open the Page

Access the **Diagnosis** information:

Step	Action
1	Right-click the name that is assigned to your Ethernet communications module in the **DTM Browser**.
2	Scroll to **Device Menu → Diagnosis** to view the available diagnostics pages.

Diagnostics Information

The diagnostics window has two distinct areas:

- left pane: LED icons indicate the operating status of modules, devices, and connections.
- right pane: These pages show diagnostics data for these items:
  - Ethernet communications module
  - local slave nodes that are activated for the communication module
  - EtherNet/IP connections between the communication module and a remote EtherNet/IP device

When the appropriate 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 these status icons over the module, device, or connection in the left pane of the **Diagnostic** window to indicate its current status:

Icon	Communication module	Connection to a remote device
	Run state is indicated.	The health bit for every EtherNet/IP connection and Modbus TCP request (to a remote device, sub-device, or module) is set to active (1).
	One of these states is indicated: ● unknown ● stopped ● not connected	The health bit for at least one EtherNet/IP connection or Modbus TCP request (to a remote device, sub-device, or module) is set to inactive (0).

Communication Module Ethernet Diagnostics

Introduction

Use the **Ethernet Diagnostic** page to view the dynamic and static data for the Ethernet ports on the Ethernet communications module.

NOTE: Before you can open the diagnostics page, make the connection between the DTM for the target communication module and the physical module.

Open the Page

Access the **Ethernet Diagnostic** information:

Step	Action
1	In the **DTM Browser**, find the name that is assigned to your Ethernet communications module.
2	Right-click on the module name.
3	Scroll to **Device menu → Diagnosis**.
4	In the left pane of the **Diagnosis** window, select the communication module node.
5	Select the **Ethernet Diagnostic** tab to open that page.

NOTE: The number of ports on the communication module determines the number of columns displayed in this page.

Data Display

Use the **Refresh Every 500ms** checkbox to display the static or dynamic data:

Checkbox	Description
Selected	• Display data that is dynamically updated every 500 ms. • Increment the number at the top of the table each time data is refreshed.
De-selected	• Display static data. • Do not increment the number at the top of the table. That number now represents a constant value.

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 (no link), 10, 100, 1000 (Mbits/s)
Interface Flags	Bit 0: Link Status (0 = **Inactive link**; 1 = **Active link**)
	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, **Inactive link**) : The interface can activate changes to link parameters automatically. 1 (**Active link**): Devices require a reset service to be issued to its Identity.
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 detected 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 detected errors (does not include In Discards)
Unknown Protocols	Outbound packets with unknown protocol
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

Parameter	Description
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 detected and generated
Deferred Transmissions	Frames for which first transmission attempt is delayed because the medium is busy
Late Collisions	Number of times a collision is detected later than 512 bit times into the transmission of a packet
Excessive Collisions	Frames for which transmission does not finish due to excessive collisions
MAC Transmit Errors	Frames for which transmission does not finish due to detected 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 does not finish due to a detected internal MAC sublayer receive error

Communication Module Bandwidth Diagnostics

Introduction

Use the **Bandwidth** page to view the dynamic and static data for the bandwidth use by the Ethernet communications module.

NOTE: Before you can open the diagnostics page, make the connection between the DTM for the target communication module and the physical module.

Open the Page

Access the **Bandwidth** information:

Step	Action
1	In the **DTM Browser**, find the name that is assigned to your Ethernet communications module.
2	Right-click on the module name.
3	Scroll to **Device menu → Diagnosis**.
4	In the left pane of the **Diagnosis** window, select the communication module node.
5	Select the **Bandwidth** tab to open that page.

Data Display

Use the **Refresh Every 500ms** checkbox to display the static or dynamic data:

Checkbox	Description
Selected	• Display data that is dynamically updated every 500 ms. • Increment the number at the top of the table each time data is refreshed.
De-selected	• Display static data. • Do not increment the number at the top of the table. That number now represents a constant value.

Bandwidth Diagnostic Parameters

The **Bandwidth** 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 in packets/second.
EtherNet/IP Received	The number of EtherNet/IP packets the module has received in packets/second.
Modbus TCP Received	The number of Modbus TCP requests the module has sent in packets/second.
Modbus TCP Responses	The number of Modbus TCP responses the module has received in packets/second.

Parameter	Description
I/O - Adapter:	
EtherNet/IP Sent	The number of EtherNet/IP packets (per second) the module has sent in the role of a local slave.
EtherNet/IP Received	The number of EtherNet/IP packets (per second) the module has received in the role of a local slave.
I/O - Module	
Module Capacity	The maximum number of packets (per second) that the module can process.
Module Utilization	The percentage of communication module capacity being used by the application.
Messaging - Client:	
EtherNet/IP Activity	The number of explicit messages (packets per second) sent by the module using the EtherNet/IP protocol.
Modbus TCP Activity	The number of explicit messages (packets per second) sent by the module using the Modbus TCP protocol.
Messaging - Server:	
EtherNet/IP Activity	The number of server messages (packets per second) received by the module using the EtherNet/IP protocol.
Modbus TCP Activity	The number of server messages (packets per second) received by the module using the Modbus TCP protocol.
Module:	
Processor Utilization	The percent of Ethernet communication module processor capacity used by the present level of communication activity.

Communication Module RSTP Diagnostics

Introduction

Use the **RSTP Diagnostic** page to view the status of the RSTP service of the Ethernet communications module. The page displays dynamically generated and static data for the module.

NOTE: Before you can open the diagnostics page, make the connection between the DTM for the target communication module and the physical module.

Open the Page

Access the **RSTP Diagnosis** information:

Step	Action
1	In the **DTM Browser**, find the name that is assigned to your Ethernet communications module.
2	Right-click on the module name.
3	Scroll to **Device menu → Diagnosis**.
4	In the left pane of the **Diagnosis** window, select the communication module node.
5	Select the **RSTP Diagnostic** tab to open that page.

Data Display

Use the **Refresh Every 500ms** checkbox to display the static or dynamic data:

Checkbox	Description
Selected	• Display data that is dynamically updated every 500 ms. • Increment the number at the top of the table each time data is refreshed.
De-selected	• Display static data. • Do not increment the number at the top of the table. That number now represents a constant value.

RSTP Diagnostic Parameters

The **RSTP Diagnostic** page displays the following parameters for each communication module port:

Parameter	Description
Bridge RSTP Diagnostic:	
Bridge Priority	This 8-byte field contains the two-byte value that is assigned to the module's embedded Ethernet switch.
MAC Address	The Ethernet address of the module, found on the front of the module.
Designated Root ID	The Bridge ID of the root device.
Root Path Cost	The aggregate cost of port costs from this switch back to the root device.

Parameter	Description
Default Hello Time	The interval at which Configuration BPDU messages are transmitted during a network convergence. For RSTP this is a fixed value of 2 seconds.
Learned Hello Time	The current Hello Time value learned from the root switch.
Configured Max Age	The value (6 ... 40) that other switches use for MaxAge when this switch is acting as the root.
Learned Max Age	The maximum age learned from the root switch. This is the actual value currently used by this switch.
Total Topology Changes	The total number of topology changes detected by this switch since the management entity was last reset or initialized.
Ports ETH 2 and ETH 3 RSTP Statistics:	
Status	The port's current state as defined by RSTP protocol. This state controls the action the port takes when it receives a frame. Possible values are: disabled, discarding, learning, forwarding.
Role:	The port’s current role per RSTP protocol. Possible values are: root port, designated port, alternate port, backup port, disabled port.
Cost	The logical cost of this port as a path to the root switch. If this port is configured for AUTO then the cost is determined based on the connection speed of the port.
STP Packets	A value in this field indicates that a device on the network has the STP protocol enabled. **NOTE:** ● Other devices that are enabled for STP can severely affect the network convergence times. Schneider Electric recommends that you disable the STP protocol (but not the RSTP protocol) on every network device that supports STP. ● The communication module does not support the STP protocol. The module’s embedded switch ignores STP packets.

IP Forwarding Diagnostics

Diagnosing the IP Forwarding Service

Use the **IP Forwarding** page to display dynamically generated data describing the IP forwarding service *(see page 111)* of the BMENOC0321 module.

NOTE: Before you can open the diagnostics page, make the connection between the DTM *(see page 194)* for the target communication module and the physical module.

Open the page:

Step	Action
1	In the **DTM Browser**, right-click the BMENOC0321 module to open a pop-up menu.
2	In the menu, select **Device menu → Diagnostic**. The **Diagnostic** window opens.
3	In the left pane of the **Diagnostic** window, select the communication module node.
4	Click on the **IP Forwarding** tab to open that page.

IP Forwarding Diagnostic Parameters

This table describes the parameters to diagnose the IP Forwarding service:

Parameter	Description
Refresh Every 500ms	Select this to dynamically update this page every 500ms. The number of times this page has been refreshed appears immediately to the right.
Forwarding Status	Status of the IP Forwarding service: • 1: enabled • 0: disabled
Current Forwarding Load	The total load handled by the IP Forwarding service (in packets per second).
Network Destination	These entries in the network routing table are associated with the IP forwarding network configuration.
Net Mask	
Gateway Address	
IP Interface	

Email Diagnostics

Diagnosing SMTP Transmissions

Use the **Email Diagnostic** page to display dynamically generated data describing the communication module's Email message transmissions.

NOTE: Before you can open the diagnostics page, make the connection between the DTM *(see page 194)* for the target communication module and the physical module.

Open the 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**. The **Diagnostic** window opens.
3	In the left pane of the **Diagnostic** window, select the communication module node.
4	Click on the **Email Diagnostic** tab to open that page.

Email diagnostic Parameters

This table describes the diagnostic parameters for the email service:

Parameter	Description
Refresh Every 500ms	Select this to dynamically update this page every 500ms. The number of times this page has been refreshed appears immediately to the right.
Email Service Status	The status of this service in the Ethernet communication module: ● green = operational (OK) ● orange = not operational (NOK)
SMTP Server IP Address	IP address of the SMTP server
Sender	The three header fields of the last Email message sent.
Receivers	
Subject	
Number of Emails Sent	Total number of emails sent and successfully acknowledged by the SMTP server.
Time Since Last Email	Counts the number of seconds since the last email was successfully sent.
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 detected transmission errors.

Parameter	Description
Number of Errors	Total number of emails that either: ● could not be sent ● were sent but were not successfully acknowledged by the SMTP server
Email Service Not Reachable	Number of times the SMTP server could not be reached. (Link checked every 30 minutes.)

Click the **Reset Counter** button to reset the counting statistics on this page to 0.

Network Time Service Diagnostics

Introduction

Use the **Network Time Service Diagnostic** page to display dynamically generated data describing the operation of the simple network time protocol (SNTP) service that you configured in the network time server page *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* in Control Expert.

NOTE: Before you can open the diagnostics page, make the connection between the DTM for the target communication module and the physical module.

Refer to the *System Time Stamping User Guide (see System Time Stamping, User Guide)* for detailed diagnostic information.

Open the Page

Access the **NTP Diagnostic** information:

Step	Action
1	In the **DTM Browser**, find the name that is assigned to your Ethernet communications module.
2	Right-click on the module name.
3	Scroll to **Device menu → Diagnosis**.
4	In the left pane of the **Diagnosis** window, select the communication module node.
5	Select the **NTP Diagnostic** tab to open that page.

Click the **Reset Counter** button to reset the counting statistics on this page to 0.

Network Time Service Diagnostic Parameters

This table describes the time synchronization service parameters:

Parameter	Description
Refresh Every 500ms	Check this box to dynamically update the page every 500ms. The number of times this page has been refreshed appears immediately to the right.
Network Time Service	Monitor the operational status of the service in the module: ● *green*: operational ● *orange*: disabled
Network Time Server Status	Monitor the communication status of the NTP server: ● *green*: The NTP server is reachable. ● *red*: The NTP server is not reachable.
Last Update	Elapsed time, in seconds, since the most recent NTP server update.
Current Date	System date
Current Time	The system time is presented in the *hh:mm:ss* format.

Parameter	Description	
DST Status	Set the status of the automatic daylight savings service: ● *ON*: The automatic adjustment of daylight savings is enabled. The current date and time reflect the daylight savings time adjustment. ● *OFF*: The automatic adjustment of daylight savings is disabled. (The current date and time may not reflect the daylight savings time adjustment.)	
Quality	This correction (in seconds) applies to the local counter at every NTP server update. Numbers greater than 0 indicate increasingly excessive traffic condition or an NTP server overload.	
Requests	This value represents the total number of client requests sent to the NTP server.	
Responses	This value represents the total number of server responses sent from the NTP server.	
Errors	This value represents the total number of unanswered NTP requests.	
Last Error	This value indicates the 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: primary and secondary IP addresses that are not NTP server address ● 7: unrecoverable NTP transmission ● 9: invalid NTP server IP address ● 15: invalid syntax in the custom time zone rules file	
Primary / Secondary NTP Server IP	The IP addresses correspond to the primary and secondary NTP servers. **NOTE:** A green LED to the right of the primary or secondary NTP server IP address indicates the active server.	
Auto Adjust Clock for Daylight Savings	Configure the daylight savings adjustment service: ● enabled ● disabled	
DST Start / DST End	Specify the day on which daylight savings time begins and ends:	
	Month	Set the month in which daylight savings time starts or ends.
	Day of Week	Set the day of the week on which daylight savings time starts or ends.
	Week#	Set the occurrence of the specified day within the specified month.
Time Zone	Select the time zone plus or minus Universal Time, Coordinated (UTC)	
Offset	Configure the time (in minutes) to be combined with the time zone selection (above) to produce the system time.	
Polling Period	Set the frequency with which the NTP client requests an updated time from the NTP server	

Hot Standby Diagnostics

Diagnosing Hot Standby

If the synchronization between the Ethernet communication modules does not work properly before the Hot Standby switchover occurs, use the information on the **Hot Standby Diagnostic** page.

NOTE: Before you can open the **Diagnostic** window, 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**.

Open the 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**. The **Diagnostic** window opens.
3	In the left pane of the **Diagnostic** window, select the communication module node.
4	Click on the **Hot Standby Diagnostic** tab to open that page.

Hot Standby Diagnostic Parameters

This table describes the Hot Standby diagnostic parameters:

Parameter		Description
Refresh Every 500ms		Select this to dynamically update this page every 500ms. The number of times this page has been refreshed appears immediately to the right.
Status	Synchronizing	OFF or ON
	Synchronized	YES or NO
	Error Status	*green*: No errors are detected.
		red: At least one error is detected.
Manual Synchronization	Stop Synchronization Service	Select an activity. **NOTE:** If you select **Manual Synchronization**, the **Force Manual Synchronization** field is disabled. In that case, the synchronization status is ON, and the modules are synchronized.
	Copy Files from Standby to Primary	
	Copy Files from Primary to Standby	
	Clear Files in Primary	
	Send	

Local Slave / Connection Diagnostics

Introduction

Use the **Local Slave Diagnostic** page and the **Connection Diagnostic** page to display the I/O status and production/consumption information for a selected local slave or connection.

NOTE: Before you can open the diagnostics page, make the connection *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* between the DTM for the target communication module and the physical module.

Open the Page

Access the diagnostics information:

Step	Action
1	In the **DTM Browser**, find the name that is assigned to your Ethernet communications module.
2	Right-click on the module name.
3	Scroll to **Device menu → Diagnosis**.
4	In the left pane of the **Diagnosis** window, select the communication module node.
5	Select the **Local Slave Diagnostic** tab or the **Connection Diagnostic** tab to open that page.

Data Display

Use the **Refresh Every 500ms** checkbox to display the static or dynamic data:

Checkbox	Description
Selected	• Display data that is dynamically updated every 500 ms. • Increment the number at the top of the table each time data is refreshed.
De-selected	• Display static data. • Do not increment the number at the top of the table. That number now represents a constant value.

Local Slave / Connection Diagnostic Parameters

This following tables display the diagnostic parameters for the selected local slave or scanner connection.

This table shows the **Status** diagnostic parameters for the selected connection:

Parameter	Description
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.

The **Input** and **Output** status diagnostic parameters can present these values:

Input/Output Status (dec)	Description
0	OK
33	Time-out
53	IDLE
54	Connection established
58	Not connected (TCP)
65	Not connected (CIP)
68	Connection establishing
70	Not connected (EPIC)
77	Scanner stopped

This table shows the **Counter** diagnostic parameters for the selected connection:

Parameter	Description
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 was last reset.
Theoretical Packets per second	Packets per second calculated using current configuration value.
Real Packets per second	Actual number of packets per second generated by this connection.

This table shows the **Diagnostic** parameters for the selected connection:

Parameter	Description
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	Actual packet interval (API) of the production connection.
T -> O API	Actual packet interval (API) of the consumption connection.
O -> T RPI	Requested packet interval (RPI) of the production connection.
T -> O RPI	Requested packet interval (RPI) of the consumption connection.

This table shows the **Socket Diagnostics** diagnostic parameters for the selected connection:

Parameter	Description
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.

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 scanner connection.

NOTE: Before you can open the diagnostics page, make the connection *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* between the DTM for the target communication module.

Open the Page

Access the **I/O Values** information:

Step	Action
1	In the **DTM Browser**, find the name that is assigned to your Ethernet communications module.
2	Right-click on the module name.
3	Scroll to **Device menu → Diagnosis**.
4	In the left pane of the **Diagnosis** window, select the communication module node.
5	Select the **I/O Values** tab.

Data Display

Use the **Refresh Every 500ms** checkbox to display the static or dynamic data:

Checkbox	Description
Selected	• Display data that is dynamically updated every 500 ms. • Increment the number at the top of the table each time data is refreshed.
De-selected	• Display static data. • Do not increment the number at the top of the table. That number now represents a constant value.

Local Slave / Scanner Connection I/O Values

This page displays theses parameters for either a local slave or a remote device connection input and output values:

Parameter	Description
Input/Output data display	This parameter displays the input or output data image for a local slave or remote device.
Length	The **Length** parameter shows the number of bytes in an input or output data image.
Status	The **Status** parameter indicates the status of the scanner diagnostic object that is reported in the input or output data image: ● *0*: The connection is OK. ● *54*: The connection is in progress. The I/O data are not exchanged. ● *33*: There is no connection. ● *53*: A notification of IDLE is received.

Section 7.4
Online Action

What Is in This Section?

This section contains the following topics:

Online Action

Introduction

Use the **Online Action** page in the Control Expert DTM to view and edit online parameters for the Ethernet communications module. Online actions support these tasks:

- Display EtherNet/IP objects for the Ethernet communications module or a distributed EtherNet/IP device.
- View and edit the SERVICE port configuration parameters for the Ethernet communications module.
- Ping the Ethernet communications module or a distributed EtherNet/IP or Modbus TCP device to confirm that it is active on the Ethernet network.
- Connect to a distributed device to perform these actions:
  - View the default parameter settings for the device.
  - View the current parameter settings for the device.
  - Edit and download to the device its editable parameter settings.

Connect the DTM

Before you can open the **Online Action** page, make the connection between the DTM for the target communication module and the physical module:

Step	Action
1	In the **DTM Browser**, find the name that is assigned to your Ethernet communications module.
2	Right-click on the module name.
3	Scroll to **Connect**.

Open the Page

Access the **Online Action** information:

Step	Action
1	In the **DTM Browser**, find the name that is assigned to your Ethernet communications module.
2	Right-click on the module name.
3	Scroll to **Device menu → Diagnosis**.
4	In the left pane of the **Diagnosis** window, select the communication module node.
5	Select the **Online Action** tab to open that page.

You can see these tabs:

- **EtherNet/IP Objects**
- **Service Port**
- **Ping**

EtherNet/IP Objects Tab

Introduction

Use the **EtherNet/IP Objects** tab in the **Online Action** window:

- Retrieve and display current data describing the state of CIP objects for the selected communication module or remote EtherNet/IP device.
- Reset the selected communication module or remote EtherNet/IP device.

Access the Page

Open the **EtherNet/IP Objects** tab:

Step	Action
1	Connect the DTM to the module *(see page 215)*.
2	Open the **Online Action** page *(see page 215)*.
3	Select the **EtherNet/IP Objects** tab.

Available CIP Objects

You can retrieve CIP objects according to the Control Expert operating mode:

Mode	Available CIP Objects
Standard	Identity object *(see page 225)*
Advanced	Identity object *(see page 225)*
	Connection Manager object *(see page 230)*
	TCP/IP Interface object *(see page 237)*
	Ethernet Link object *(see page 239)*
	QoS object *(see page 235)*

Advanced Mode

When advanced mode *(see page 72)* is enabled, select an object in the **Object** list.

These buttons are available in advanced mode:

Button	Action
Refresh	Click this button to update the data.
Reset Device	Click this button to reset a communication module or remote EtherNet/IP device.

Service Port Tab

Introduction

Use the **Service Port** tab in the **Online Action** window to view and edit communication port properties for a distributed EtherNet/IP device. Use this tab to execute these commands:

- *Refresh*: Use a Get command to retrieve port configuration settings from a distributed EtherNet/IP device.
- *Update*: Use a Set command to write all or selected edited values to the same distributed EtherNet/IP device

The configuration information on the **Service Port** tab is sent in EtherNet/IP explicit messages that employ the address and messaging settings configured for Ethernet/IP explicit messaging (below).

Access the Page

Open the **EtherNet/IP Objects** tab:

Step	Action
1	Connect the DTM to the module *(see page 215)*.
2	Open the **Online Action** page *(see page 215)*.
3	Select the **EtherNet/IP Objects** tab.
4	Configure the Service port with the instructions from the offline configuration *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.
5	Click the **Update** button to apply the new configuration.

Pinging a Network Device

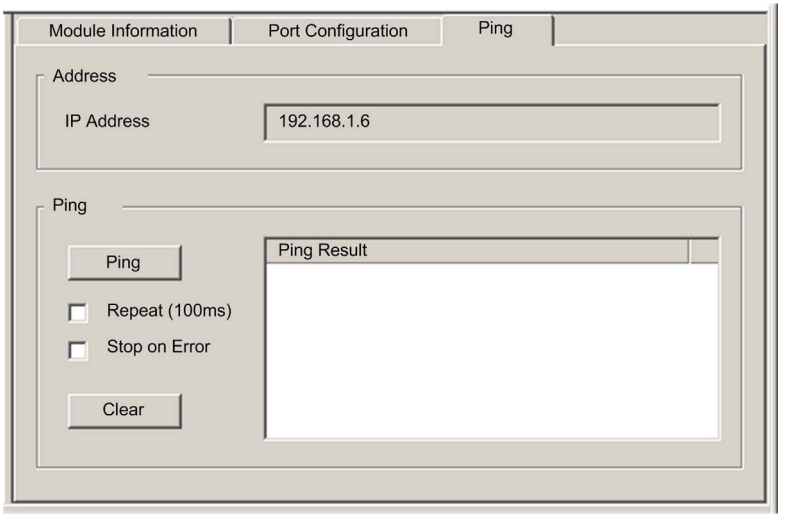
Overview

Use the Control Expert ping function to send an ICMP echo request to a target Ethernet 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. Enter only valid IP addresses in the **IP Address** field.

The ping function can be performed in the **Ping** page of the **Online Action** window:



Pinging a Network Device

Ping a network device:

Step	Action
1	In the **DTM Browser**, select the communication module upstream of the remote EtherNet/IP device you want to ping.
2	Click the right mouse button and select **Device Menu → Online Action** in the pop-up menu. The **Online Action** window opens.
3	In the **Online Action** window, select the device you want to ping. The window displays pages containing online information for the selected device. **NOTE:** The specific collection of displayed pages depends on the type of device selected: ● the communications module ● a remote EtherNet/IP device ● a remote Modbus TCP device
4	Select the **Ping** page. To send... ● a single ping, de-select the **Repeat** checkbox ● a series of pings—1 every 100 ms—select **Repeat** checkbox
5	(Optional) Select **Stop on Error** to stop pinging an unsuccessful communication.
6	Click **Ping** once to begin pinging.
7	Click **Ping** a second time to stop repeated pinging, where no error has been detected.
8	The **Ping Result** box displays the ping outcome. Click **Clear** to empty the **Ping Result** box.

Section 7.5
Diagnostics Available through Modbus/TCP

Modbus Diagnostic Codes

Introduction

CPUs and BMENOC0301/11 communication modules in M580 systems support the diagnostic codes in these tables.

Function Code 3

Some module diagnostics (I/O connection, extended health, redundancy status, FDR server, etc.) are available to Modbus clients that read the local Modbus server area. Use Modbus function code 3 with the unit ID set to 100 for register mapping:

Type	Offset Modbus Address	Size (Words)
Basic Networks Diagnostic Data	0	39
Ethernet Port Diagnostics Data (Internal port)	39	103
Ethernet Port Diagnostics Data (**ETH 1**)	142	103
Ethernet Port Diagnostics Data (**ETH 2**)	245	103
Ethernet Port Diagnostics Data (**ETH 3**)	348	103
Ethernet Port Diagnostics Data (backplane)	451	103
Modbus TCP/Port 502 Diagnostic Data	554	114
Modbus TCP/Port 502 Connection Table Data	668	515
SNTP Diagnostics	1218	57
QoS Diagnostics	1275	11
Identify	2001	24

For a description of available function codes refer to the list of supported Modbus diagnostic codes in the topic *Modbus Diagnostic Codes (see Quantum EIO, Remote I/O Modules, Installation and Configuration Guide)* in the *Quantum EIO Control Network Installation and Configuration Guide*.

Function Code 8

Modbus function code 08 provides a variety of diagnostic functions:

Operation Code	Diag. Control	Description
0x01	0x0100	network diagnostic data
	0x0200	Read the Ethernet port diagnostic data from the switch manager.
	0x0300	Read the Modbus TCP/port 502 diagnostic data from the Modbus server.
	0x0400	Read the Modbus TCP/port 502 connection table from the Modbus server.
	0x07F0	Read the data structure offset data from the Modbus server.
0x02	0x0100	Clear the basic network diagnostic data. **NOTE:** Only specific parameters of basic network diagnostic data are used to clear requests.
	0x0200	Clear the Ethernet port diagnostic data. **NOTE:** Only specific parameters of basic network diagnostic data are used to clear requests.
	0x0300	Clear the Modbus TCP/port 502 diagnostic data. **NOTE:** Only specific parameters of Modbus port 502 diagnostic data are used to clear requests.
	0x0400	Clear the Modbus TCP/port 502 connection table. **NOTE:** Only specific parameters of Modbus port 502 connection data are use to clear requests.
0x03	0	Clear all diagnostic data. **NOTE:** Only specific parameters of each diagnostic data are used to clear requests.

Read Device Identification

Modbus function code 43, subcode 14: A Modbus request associated with function code 43 (Read Device Identification) asks a Modbus server to return the vendor name, product name, version number, and other optional fields:

Category	Object ID	Object Name	Type	Requirement
Basic	0x00	VendorName (vendor name)	ASCII string	mandatory
	0x01	ProductCode (product code)	ASCII string	mandatory
	0x02	MajorMinorRevision (version number)	ASCII string	mandatory
Regular	0x03	VendorUrl (vendor URL)	ASCII string	optional
	0x04	ProductName (product name)	ASCII string	optional
	0x05	ModelName (model name)	ASCII string	optional
	0x06	UserApplicationName (user application name)	ASCII string	optional
	0x07...0x7F	(reserved)	ASCII string	optional
Extended	0x80...0xFF	device-dependent		optional

This table provides sample responses to the Modbus request (function code 43, subcode 14):

Module	0x00 Vendor ID	0x01 Part Number	0x02 Version
BMEP584020 CPU	Schneider Electric	BMEP584020	v02.10
BMENOC0301 module	Schneider Electric	BMENOC0301	V02.04 build 0009
BMENOC0311 module	Schneider Electric	BMENOC0311	V02.04 build 0009
BMENOC0321 module	Schneider Electric	BMENOC0321	V01.01 build 0004

Section 7.6
Diagnostics Available through EtherNet/IP CIP Objects

Introduction

Modicon M580 applications use CIP within a producer/consumer model to provide communication services in an industrial environment. This section describes the available CIP objects for Modicon M580 modules.

What Is in This Section?

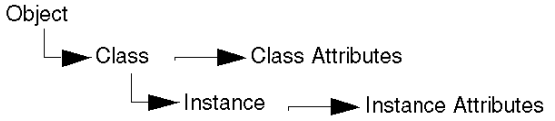
This section contains the following topics:

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. Assembly instances exist only when you configure local slaves *(see page 302)* for the Ethernet communications module.

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 these instance identifiers:

- 0: class
- 101, 102, 111, 112, 121, 122, 131, 132, 136, 137, 141, 142, 146, 147, 151, 152, 156, 157, 161, 162, 166, 167, 171, 172: instance

Attributes

The assembly object consists of these 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 (output data)	Array of BYTE	X	—
102		Local slave 1: O>T (input data)	Array of BYTE	X	—
111	03	Local slave 2: T->O (output data)	Array of BYTE	X	—
112		Local slave 2: O>T (input data)	Array of BYTE	X	—
121	03	Local slave 3: T->O (output data)	Array of BYTE	X	—
122		Local slave 3: O>T (input data)	Array of BYTE	X	—
131	03	Local slave 4: T->O (output data)	Array of BYTE	X	—
132		Local slave 4: O>T (input data)	Array of BYTE	X	—
136	03	Local slave 5: T->O (output data)	Array of BYTE	X	—
137		Local slave 5: O>T (input data)	Array of BYTE	X	—
141	03	Local slave 6: T->O (output data)	Array of BYTE	X	—
142		Local slave 6: O>T (input data)	Array of BYTE	X	—
146	03	Local slave 7: T->O (output data)	Array of BYTE	X	—
147		Local slave 7: O>T (input data)	Array of BYTE	X	—
151	03	Local slave 8: T->O (output data)	Array of BYTE	X	—
152		Local slave 8: O>T (input data)	Array of BYTE	X	—
156	03	Local slave 9: T->O (output data)	Array of BYTE	X	—
157		Local slave 9: O>T (input data)	Array of BYTE	X	—
161	03	Local slave 10: T->O (output data)	Array of BYTE	X	—
162		Local slave 10: O>T (input data)	Array of BYTE	X	—
166	03	Local slave 11: T->O (output data)	Array of BYTE	X	—
167		Local slave 11: O>T (input data)	Array of BYTE	X	—
171	03	Local slave 12: T->O (output data)	Array of BYTE	X	—
172		Local slave 12: O>T (input data)	Array of BYTE	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
X = supported — = not supported						

Attribute ID		Description	Type	GET	SET	Value
hex	dec					
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
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.
05	DSCP Scheduled	USINT	X	X	For CIP transport class 0/1 Urgent priority messages.
06	DSCP High	USINT	X	X	For CIP transport class 0/1 Urgent priority messages.
07	DSCP Low	USINT	X	X	For CIP transport class 0/1 Urgent priority messages.
08	DSCP Explicit	USINT	X	X	For CIP explicit messages (transport class 2/3 and UCMM).
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: ETH 1
- 2: ETH 2
- 3: ETH 3
- 4: backplane port
- 255: internal port

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		Description	Type	GET	SET	Value
hex	dec					
01	01	Interface Speed	UDINT	X	—	Valid values: 0, 10, 100.
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						

Attribute ID		Description	Type	GET	SET	Value
hex	dec					
06	06	Interface Control	STRUCT	X	X	API of the connection
		Control Bits	WORD			Bit 0: Auto-negotiation disabled (0) or enabled (1). **NOTE:** When auto-negotiation is enabled, 0x0C (object state conflict) is returned when attempting to set either: ● forced interface speed ● forced duplex mode
						Bit 1: forced duplex mode (if auto-negotiation bit = 0) 0 = half duplex 1 = full duplex
		Forced Interface Speed	UINT			Valid values include 10000000 and 100000000. **NOTE:** Attempting to set any other value returns the detected error 0x09 (invalid attribute value).
10	16	Interface Label	SHORT_ STRING	X	—	A fixed textual string identifying the interface, that should include 'internal' for internal interfaces. Maximum number of characters is 64.
X = supported — = not supported						

Services

The Ethernet Link object performs the following services upon the listed object types:

Service ID		Description	Class	Instance
hex	dec			
01	01	Get_Attributes_All	X	X
10	16	Set_Attribute_Single	—	X
0E	14	Get_Attribute_Single	X	X
4C	76	Get_and_Clear	—	X
X = supported — = not supported				

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)
- 257 ... 400 (instance): The instance number matches the connection number in the **Connection Settings** configuration *(see page 286)*.

NOTE: The Instance ID number = the Connection ID. For *M580* specifically, you can look up the Connection ID on the DTM Device List screen.

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: 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 2 (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.
X = supported — = not supported					

Attribute ID	Description	Type	GET	CLEAR	Value
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
02	02	Mirror	UINT	X	X	bit 0 (default): ETH 2 port bit 1: ETH 3 port bit 2: backplane port bit 3: internal port
X = supported — = not supported						

NOTE:

- If the SERVICE 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 port acts like a read-only port. That is, you cannot access devices (ping, connection to Control Expert, etc.) through the SERVICE 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					

Router Diagnostics Object

Overview

The Router Diagnostics object presents the instances, attributes and services described below.

Class ID

402 (hex), 1026 (decimal)

Instance IDs

The Router Diagnostics objects presents 2 instance values:

- 0: class
- 1...*N*: instance

Attributes

The Router Diagnostic object attributes are associated with each instance.

Instance ID = 0 (class attributes):

Attribute ID	Description	Type	GET	SET	Value
01	revision: increased by 1 at each new update of the object	UINT	X	—	current value: 1
02	max instance: the maximum instance number of the object	UINT	X	—	default value: 1
03	number of instances: the number of object instances currently created at this class level of the device	UINT	X	—	current value: 1
04	optional attribute list: the number of attributes in the optional attribute list	UINT	X	—	current value: 0
05	optional list: the number of services in the optional services list	UINT	X	—	current value: 0
X = supported — = not supported					

Attribute ID	Description	Type	GET	SET	Value
06	maximum ID number of class attributes: the attribute ID number of the last class attribute of the class definition implemented in the device	UINT	X	—	current value: 7
07	maximum ID number of instance attributes: the attribute ID number of the last instance attribute of the class definition implemented in the device	UINT	X	—	default value: 2
X = supported — = not supported					

Instance ID = 1 to *N* (instance attributes):

Attribute ID	Description	Type	GET	CLEAR	Value
01	forwarding status: whether IP forwarding services are enabled or not	UINT	X	—	enabled (1): forwarding disabled (0): discarding default: 0
02	current forwarding load: total load, in packets per seconds, handled by the IP forwarding service	UINT	X	—	default: 0
X = supported — = not supported					

Services

The Router 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 object
0E	14	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.
X = supported — = not supported					

Router Routing Table Object

Overview

The Router Routing Table object presents the instances, attributes and services described below.

Class ID

403 (hex), 1027 (decimal)

Instance IDs

The Router Routing Table objects presents 2 instance values:

- 0: class
- 1...*N*: instance

Attributes

The Router Routing Table object attributes are associated with each instance.

Instance ID = 0 (class attributes):

Attribute ID	Description	Type	GET	SET	Value
01	revision: increased by 1 at each new update of the object	UINT	X	—	current value: 1
02	max instance: the maximum instance number of the object	UINT	X	—	current value: 32
03	number of instances: the number of object instances currently created at this class level of the device	UINT	X	—	
X = supported — = not supported					

Instance ID = 1 to *N* (instance attributes):

Attribute ID	Description	Type	GET	CLEAR
01	route entry: information about the entry in the routing table, including: ● UDINT: route/network destination ● UDINT: net mask ● UDINT: gateway address ● UDINT: IP interface ● UINT: cost ● UDINT: incoming packets per second ● UDINT: outgoing packets per second	Struct	X	—
X = supported — = not supported				

Services

The Router Routing Table 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 object
0E	14	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.
X = supported — = not supported					

SMTP Diagnostics Object

Overview

The SMTP Diagnostics object presents the instances, attributes and services described below.

Class ID

404 (hex), 1028 (decimal)

Instance IDs

The SMTP Diagnostics object presents two instance values:

- 0: class
- 1: instance

Attributes

SMTP Diagnostics object attributes are associated with each instance, as follows:

Instance ID = 0 (class attributes):

Attribute ID	Description	GET	SET	Data Type
01	Revision	X	—	UINT
02	Max Instance	X	—	UINT
X = supported — = not supported				

Instance ID = 1 to 256 (instance attributes):

Attribute ID	Description	Type	GET	SET	Value
01	SMTP Server IP Address	UDINT	X	—	
02	Email Service Status	UDINT	X	—	1 = Idle 2 = Operationa 3 = Stoppedl
03	Link to SMTP Server Status	UDINT	X	—	1 = OK 2 = NOK
04	Number of Emails Sent	UDINT	X	—	
05	Number of Response from the Server	UDINT	X	—	
06	Number of Errors	UDINT	X	—	
07	Last Error	UDINT	X	—	
X = supported — = not supported					

Attribute ID	Description	Type	GET	SET	Value
08	Last Email Header Used	Array of Octets	X	—	
09	Time Elapsed from the Last Email	DINT	X	—	-1 = no email was sent (or statistics were cleared)
0A	Number of Time Server Was Not Reachable	UDINT	X	—	
X = supported — = not supported					

Services

The SMTP 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	To get all attributes in one message.
0E	14	Get_Attribute_Single	X	X	To get a single attribute as specified.
4C	76	Get_and_Clear	—	X	Clears data in the following attributes: 4, 5, 6, 7, 8, 9, 10
X = supported — = not supported					

Section 7.7
Hot Standby Services

What Is in This Section?

This section contains the following topics:

Hot Standby Synchronization

Introduction

An M580 Hot Standby system includes CPUs on two different racks, rack A and rack B. Rack A is the primary rack and rack B is the standby rack. After a switchover, rack B becomes the primary and rack A becomes the standby. The BMENOC0301/11 modules in rack A synchronize with the BMENOC0301/11 modules in rack B to update rack B with the data from rack A.

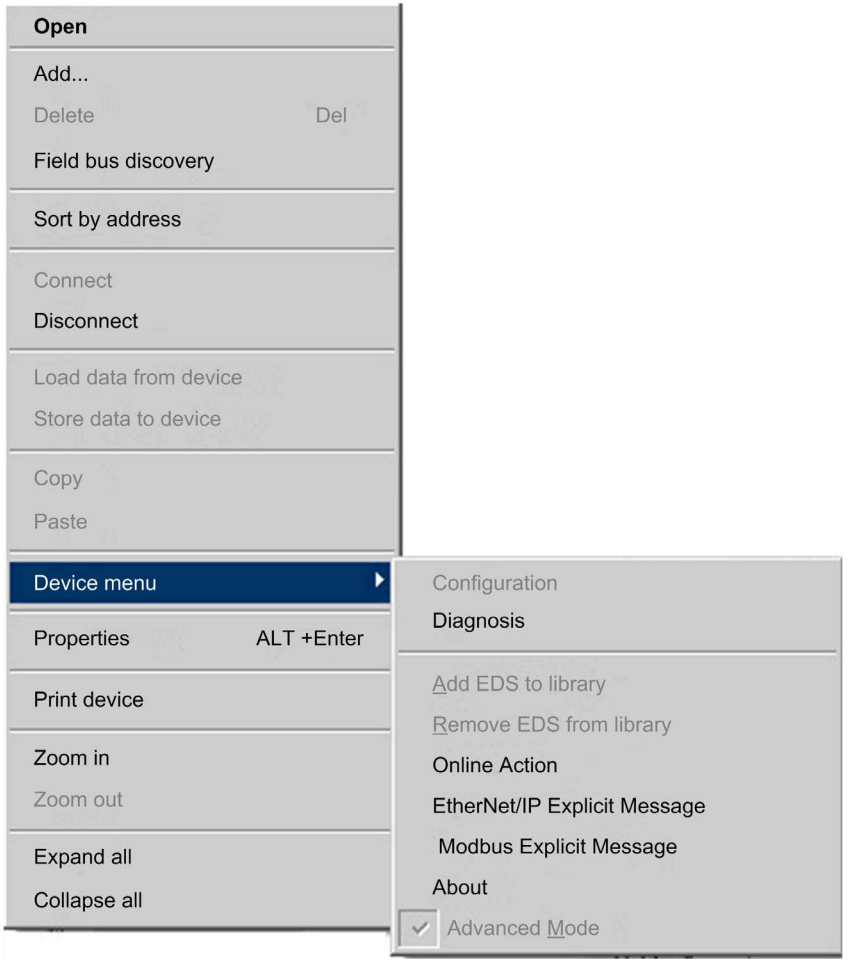
The BMENOC0301/11 standby modules then synchronize with the primary modules every 10 seconds to verify that the system, the PRM files managed by the FDR server in the standby modules has been updated in the primary modules. If the standby modules unsuccessfully synchronize with the primary modules, they keep polling for the primary modules every 10 seconds.

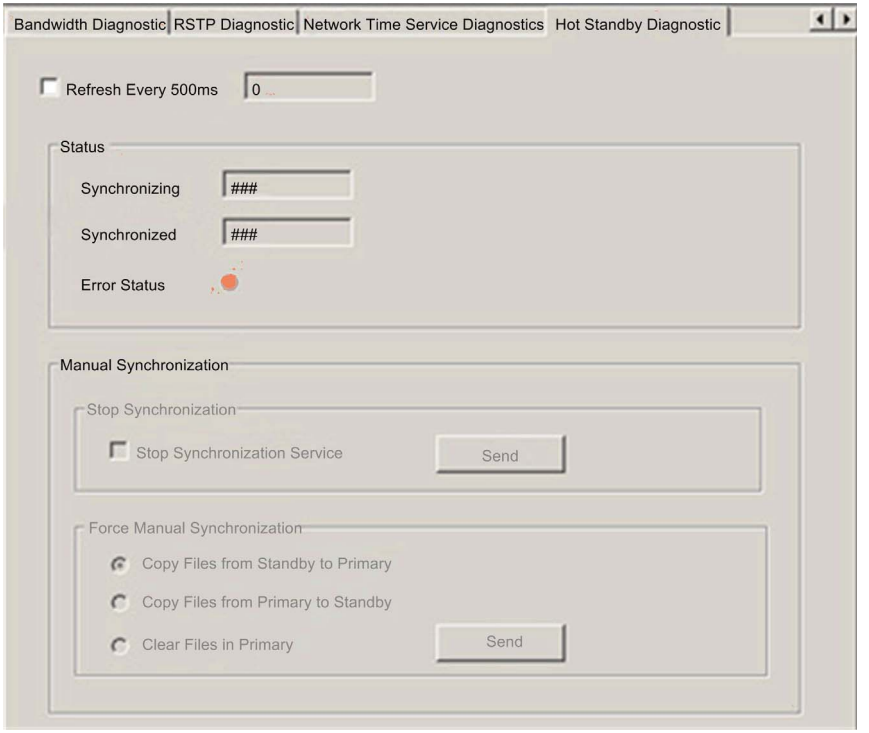
If the PRM files in the standby and primary modules are different, the synchronization stops and a synchronization error is detected in the standby rack. This process checks to see if PRM files were added to the previous primary module before the polling period expired when the Hot Standby system switchover occurred.

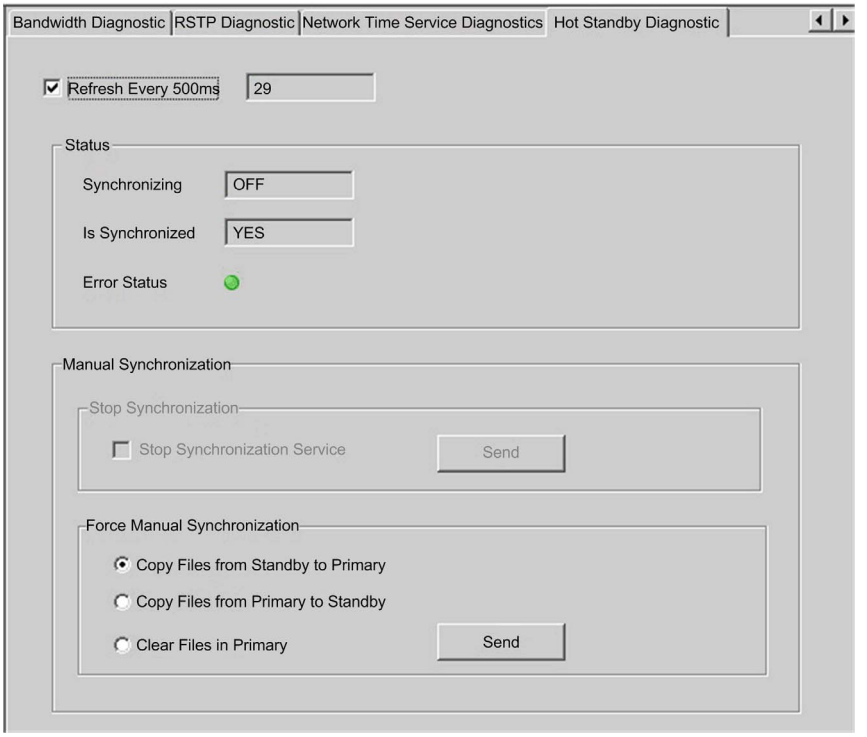
NOTE: When the BMENOC0301/11 standby modules are offline, they do not synchronize.

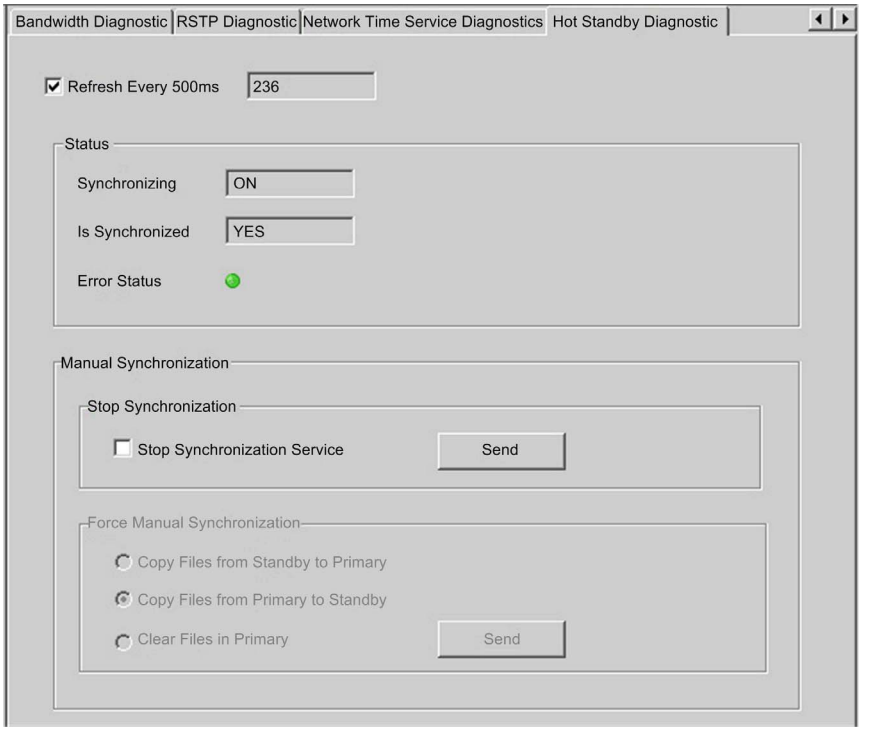
Recovering from a Synchronization Detected Error

If the synchronization between BMENOC0301/11 modules does not work properly, follow these steps:

Step	Action
1	In the **DTM Browser** window, right-click the **BMENOC03•1 module → Connect**.
2	Right-click the **BMENOC03•1 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

BMENOC0321 IP Address Swap Time

The following table details the BMENOC0321 control network module IP address swap time in an M580 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 BMENOC0321 module and the end device. Confirm that the application can tolerate this communication disruption.

Chapter 8
Implicit Messaging

Introduction

Use implicit messaging to create a communications link between the BMENOC0301/11 on an M580 rack and network devices.

The BMENOC0301/11 module manages the communications link to facilitate the exchange of I/O data between the M580 CPU and Modbus TCP and EtherNet/IP devices on the network. Using the BMENOC0301/11 module as a local slave is another example of implicit messaging.

What Is in This Chapter?

This chapter contains the following sections:

Section 8.1
Adding an EtherNet/IP Device to the Network

Introduction

This section extends the sample Control Expert application and contains these instructions:

- Add an STB NIC 2212 EtherNet/IP network interface module to your Control Expert application.
- Configure the STB NIC 2212 module.
- Configure EtherNet/IP connections to link the Ethernet communications module and the STB NIC 2212 network interface module.
- Configure I/O items for the Advantys island.

NOTE: The instructions in this section describe an example of a single, specific device configuration. For other configuration choices, refer to the Control Expert help files.

What Is in This Section?

This section contains the following topics:

Setting Up Your Network

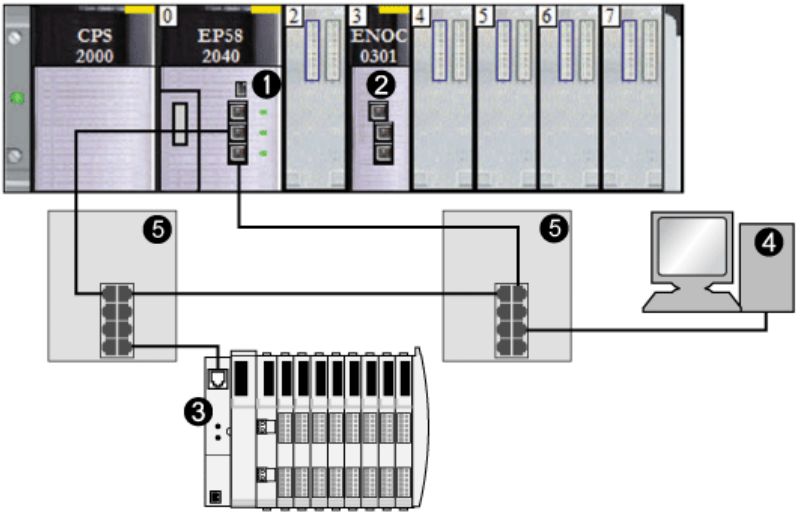
Introduction

Use this example to establish communications between the M580 rack and an Advantys STBNIC2212 network interface module (NIM).

The STBNIC2212 is Schneider Electric's EtherNet/IP network interface module for Advantys islands.

Network Topology

The Ethernet network devices used in this configuration include the following:



1 M580 CPU with DIO scanner service
2 BMENOC0301/11 Ethernet communication module in slot 3 of the local rack
3 STBNIC2212 NIM on an Advantys island
4 PC running Control Expert software
5 dual-ring switch (DRS)

To re-create this example, use the IP addresses from your own configuration for these items:

- PC
- BMENOC0301/11 Ethernet communication module
- STBNIC2212 network interface module

NOTE: Control Expert software running in the PC is used to configure the M580 CPU. In this example, the PC is indirectly wired to the CPU's Ethernet port via the Ethernet switch. Alternatively, you could bypass the switch and directly wire the PC to the CPU's Modbus ports.

Adding an STB NIC 2212 Device

Overview

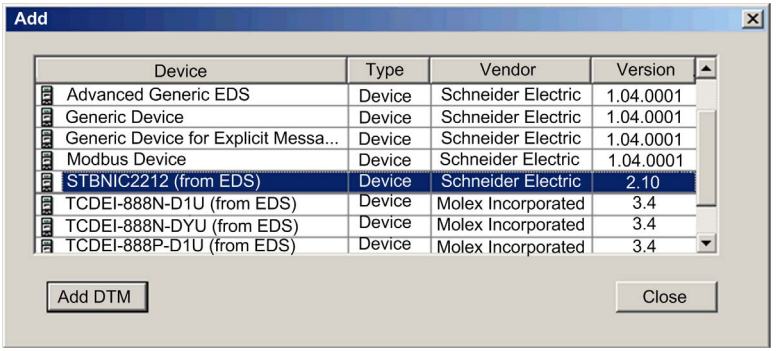
You can use the Control Expert device library to add a remote device—in this example the STB NIC 2212 module—to your project. Only a remote device that is part of your Control Expert device library can be added to your project.

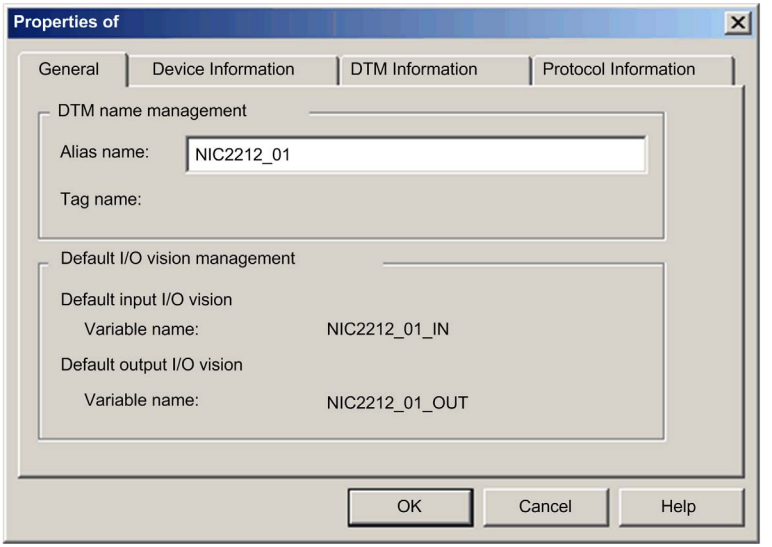
Alternatively, with a remote device already added to your device library, you can use automatic device discovery to populate your project. Perform automatic device discovery by using the **Field bus discovery** command with a communication module selected in the **DTM Browser**.

Adding an STB NIC 2212 Remote Device

NOTE: This example uses a device-specific DTM. If you do not have a device-specific DTM, Control Expert provides a generic device DTM.

Add the STB NIC 2212 to your project:

Step	Action
1	In the **DTM Browser**, right-click the DTM that corresponds to the Ethernet communication module.
2	Scroll to **Add**.
3	Select **STBNIC2212 (from EDS)**:  **NOTE:** Click a column name to sort the list of available devices. (For example, click **Device** to view the items in the first column in alphabetical order.)
4	Click the **Add DTM** button to see the association between the Ethernet communication module and the STB NIC 2212 in the **DTM Browser**.
5	In the **DTM Browser**, right-click the STB NIC 2212 node that is associated with the Ethernet communication module DTM.
6	Scroll to **Properties**.

Step	Action
7	On the **General** tab, create a unique **Alias name**. (Using similar devices that use the same DTM can result in duplicate module names.) In this example, type in the name **NIC2212_01**:  Control Expert uses the **Alias name** as the base for both structure and variable names. **NOTE:** The **Alias name** is the only editable parameter on this tab. The other parameters are read-only.
8	Click **OK** to add the STB NIC 2212 network interface module to the **DTM Browser**, beneath the communication module.

The next step is to configure the device you have just added to the project.

Configuring STB NIC 2212 Properties

Introduction

Use Control Expert to edit the settings for STB NIC 2212 device.

NOTE: To edit these settings, disconnect the DTM from a device *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.

Accessing the Device Properties

View the **Properties** tab:

Step	Action
1	Double-click the DTM for the BMENOC0301 in slot 3 (<192.168.20.10> BMENOC0301_slot3) to access the configuration. **NOTE:** This example uses a BMENOC0301 module. Use the same instructions for other M580 communications modules (like the BMENOC0311 or BMENOC0321).
2	In the navigation tree, expand the **Device List** *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* to see the associated local slave instances.
3	Select the device that corresponds to the name **NIC2212_01** to see the **Properties** *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* and **Address Setting** *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* tabs.

Properties Tab

Configure the **Properties** tab to perform these tasks:

- Add the STB NIC 2212 to the configuration.
- Remove the STB NIC 2212 from the configuration.
- Edit the base name for variables and data structures used by the STB NIC 2212.
- Indicate how input and output items are created and edited.

The descriptions for parameters *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* in the **Properties** tab are described in the configuration chapter. Use these values and names from the sample configuration:

Field	Parameter	Description
Properties	**Number**	Accept the auto-generated value.
	Active Configuration	Accept the default (**Enabled**).
IO Structure Name	**Structure Name**	Control Expert automatically assigns a structure name based on the variable name, in this case **T_STBNIC2212_from_EDS**.
	Variable Name	**Variable Name**: Accept the auto-generated variable name (based on the alias name): **STBNIC2212_from_EDS**.
	Default Name	Press this button to restore the default variable (**T_NIC2212_01**) and structure (**NIC2212_01**) names. For this example, custom names are used.
Items Management	**Import Mode**	Select **Manual**.
	Reimport Items	Press this button to import 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**.

Click **Apply** to save your edits and leave the window open.

Address Setting Tab

Use the **Address Setting** tab to enable the DHCP client in the STB NIC 2212 network interface module. When the DHCP client is enabled in the remote device, it obtains its IP address from the DHCP server in the Ethernet communication module

Configure the **Address Setting** page to perform these tasks:

- Configure the IP address for a device.
- Enable or disable DHCP client software for a device.

The descriptions for parameters *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* in the **Address Setting** tab are described in the configuration chapter. Use these values and names from the sample configuration:

Field	Parameter	Description
Change Address	**IP Address**	Enter the IP address **192.168.1.6**.
Address Server	**DHCP for this Device**	Select **Enabled**.
	Identified by	Select **Device Name**.
	Identifier	Accept the default setting (based on the **Alias name**).
	Mask	Accept the default value (255.255.255.0).
	Gateway	Accept the default value (0.0.0.0).

The next step is to configure the connection between the communication module and the remote device.

Configuring EtherNet/IP Connections

Overview

An EtherNet/IP connection provides a communication link between two or more devices. Properties for a single connection can be configured in the DTMs for the connected devices.

The following example presents settings for a connection between the Ethernet communication module and a remote STB NIC 2212 network interface module. Configuration edits are made to the DTMs for each device.

When making DTM edits, disconnect the selected DTM from the actual module or device *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.

Accessing the Connection Information

View the connection information tabs:

Step	Action
1	Double-click the DTM for the BMENOC0301 in slot 3 (<192.168.20.10> BMENOC0301_slot3) to access the configuration. **NOTE:** This example uses a BMENOC0301 module. Use the same instructions for other M580 communications modules (like the BMENOC0311 or BMENOC0321).
2	In the navigation tree, expand the **Device List** *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* to see the associated local slave instances.
3	Expand (+) the device that corresponds to the name **NIC2212_01**.
4	Select **Read Input/ Write Output Data** to see the **Connection Settings** and **Connection Information** tabs.

Connection Settings

Control Expert automatically creates a connection between a communication module and remote device when the remote device is added to the Control Expert project. Thereafter, many edits to the connection can be made in the DTM for the remote device. However, some of the connection parameters can also be configured in the DTM for the communication module, as demonstrated below.

Edit these parameters on the **Connection Settings** tab. Use settings that are appropriate to your application:

Parameter	Description
Connection Bit	The (read-only) offset for both the health bit and the control bit for this connection. Offset values are auto-generated by the Control Expert DTM.
Request Packet Interval (RPI)	The refresh period for this connection in ms (2 ... 65535). Default = 12 ms. Enter **30** ms. **NOTE:** This parameter can be set in the DTM for the communication module or 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. For this example, accept the default (**x4**). **NOTE:** To view the **Time-out Multiplier** parameter, confirm that Control Expert is operating in **Advanced Mode**.
Input Fallback Mode	This value is **Set To Zero** when communication is lost.

NOTE: The connection Information page is read-only when the communication module is selected. This information needs to be set in the DTM for the remote device.

Click **OK** to save your settings.

Configuring Connection Settings in the Remote Device DTM

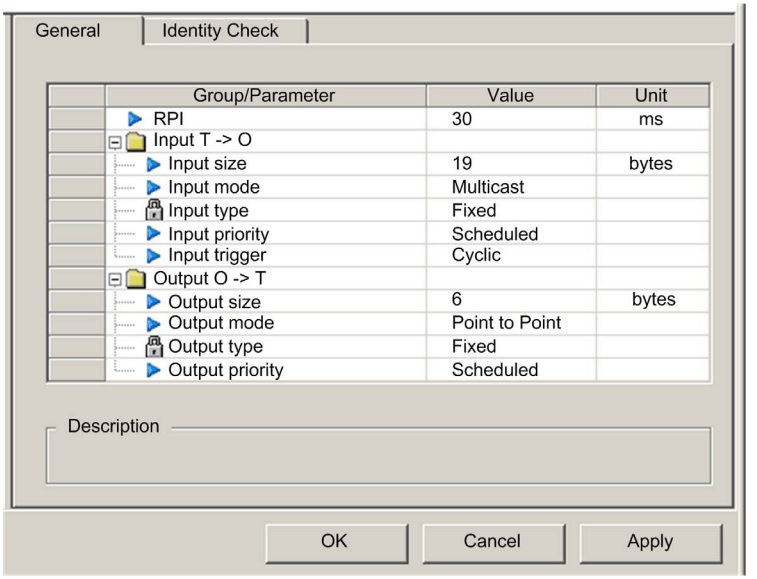
Connections between a communication module and remote device can be created and edited in the DTM for the remote device.

In this example, the following configuration edits are made to the connection that Control Expert automatically created, when the remote device was added to the project. Use settings that are appropriate for your actual application:

Step	Action
1	In the **DTM Browser**, expand the master DTM for the BMENOC0301 in slot 3 (<192.168.20.10> BMENOC0301_slot3).
2	Double-click the device DTM that corresponds to the name **NIC2212_01** to open the configuration window.
3	To view the connection type, expand **NIC2212_01** in the navigation pane. If the connection type is not of the type **Read Input / Write Output Data**, delete the existing connection and add a new one: **a.** Select the connection in the left pane. **b.** Click the **Remove Connection** button to remove the existing connection. **c.** Click the **Add Connection** button to open the **Select the connection to add** dialog. **d.** Scroll to the **Read Input / Write Output Data** connection type. **e.** Click **OK** to close the **Select the connection to add** dialog and add the new connection node to the **NIC2212_01**. **f.** Click **Apply** to save the new connection and leave the configuration window open.

General Tab

In the navigation pane, select **Read Input / Write Output Data** to see the **General** tab:



Edit the settings in the **General** tab:

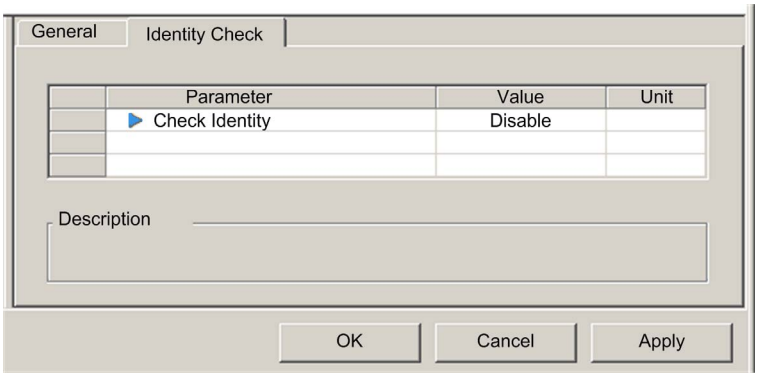
Parameter	Description
RPI	The refresh period for this connection. Accept the value of **30** ms. (This parameter can be set in the DTM for the communication module or the remote device.)
Input size	The number of bytes (0 ... 509) configured in the STB NIC 2212 module. For this example, enter **19** to reserve 20 bytes of input memory.
Input mode	Transmission type: ● Multicast ● Point to Point For this example, accept the default (Multicast).
Input type	Ethernet packet type (fixed or variable length) to be transmitted. (Only **Fixed** length packets are supported.)

Parameter	Description
Input priority	The transmission priority value depends upon the device DTM. These are the available values: ● Low ● High ● Scheduled For this example, accept the default selection (Scheduled). **NOTE:** For remote modules that support more than one priority value, you can use this setting to specify the order in which the Ethernet communication module handles packets. For more information, refer to the topic describing QoS packet prioritization *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.
Input trigger	These are the available transmission trigger values: ● Cyclic ● Change of state or application For input I/O data, select **Cyclic**.
Output size	The number of bytes configured in the STB NIC 2212 module in increments of 4 bytes (2 words). For this example, enter **6** to reserve 8 bytes of output memory.
Output mode	Accept the default (**Point to Point**).
Output type	(Read-only). Only **Fixed** length packets are supported.
Output priority	Accept the default (**Scheduled**).

Click **Apply** to save your settings and leave the window open.

Identity Check Tab

Use the **Identity Check** tab to set rules for comparing the identity of the network devices (as defined by their DTM or EDS files) against the identity of the actual network device:



Use the **Check Identity** parameter to set the rules that the BMENOC0301 uses to compare the configured versus the actual remote device:

- **Must match exactly**: The DTM or EDS file exactly matches the remote device.
- **Disable**: No checking occurs. The identity portion of the connection is filled with zero values (the default setting).
- **Must be compatible**: If 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, to be set individually.

Edit the settings in the **Identity Check** tab:

Parameter	Description
Compatibility Mode	**True**: For each of the following selected tests, the DTM/EDS and remote device need only be compatible.
	False: For each of the following selected tests, the DTM/EDS and remote device need to match exactly.
Compatibility Mode	Make a selection for each of these parameters: • **Compatible**: Include the parameter in the test. • **Not checked**: The parameter is not included in the test.
Minor Version	
Major Version	
Product Code	
Product Type	
Product Vendor	

Click **OK** to save your settings and close the window.

The next step is to configure the I/O settings.

Configuring I/O Items

Overview

The final task in this example is to add I/O items to the configuration of the STB NIC 2212 and its I/O modules. To accomplish this:

- use the Advantys configuration software to identify the relative position of each I/O module's inputs and outputs
- use the Control Expert **Device Editor** to create input and output items, defining each item's:
  - name
  - data type

I/O Item Types and Sizes

The goal is to create a collection of input items and output items that equal the input size and output size specified for the STB NIC 2212 *(see Premium using EcoStruxure™ Control Expert, TSX ETC 101 Ethernet Communication Module, User Manual)*.

The Control Expert **Device Editor** provides great flexibility in creating input and output items. You can create input and output items in groups of 1 or more single bits, 8-bit bytes, 16-bit words, 32-bit words, or 32-bit IEEE floating values. The number of items you create depends on the data type and size of each item.

Mapping Input and Output Items

Use the **Fieldbus Image** page of the **I/O Image Overview** window in the Advantys configuration software to identify the number and type of I/O items you create:

Step	Action
1	In the Advantys configuration software, select **Island → I/O Image Overview**. The **I/O Image** window opens to the **Fieldbus Overview** page.
2	Select the first cell (word 1, cell 0) in the **Input Data** table to display—in the middle of the page—a description of the cell data and its source module.
3	Make a note of the word, bit(s), module and item information for that cell.
4	Repeat the above steps for each cell that contains an S or an integer.

NOTE: The fieldbus image presents input and output data in the form of 16-bit words (starting with word 1). You need to rearrange this data for the Control Expert Ethernet Configuration Tool, which presents the same data in the form of 8-bit bytes (starting with byte 0).

NOTE: When you create items, align items of data type `WORD` and `DWORD`, as follows:

- `WORD` items: align these items on a 16-bit boundary
- `DWORD` items: align these items on a 32-bit boundary.

This example shows you how to create input bytes and output bytes. To use space efficiently, this example creates items in this sequence:

- input bit items
- input byte and word items
- output bit items
- output byte and word items

Open the **Items** configuration in Control Expert *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.

Creating Input Bit Items

Create input bit items *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* for the STB NIC 2212 example, beginning with discrete inputs for the NIC 2212 status:

Step	Action
1	Select the **Input (bit)** tab and follow directions to create input bit items. Use the default root name to represent the device status (DDI3232_in_data) in the **Default Items Name Root** field.
2	In the **Items List**, select the first two rows in the table. (These rows represent bits 0-1 in byte.)
3	Click the **Define Item(s)** button to open the **Item Name Definition** dialog box. **NOTE:** An asterisk (*) in the **Item Name** field indicates that discrete items with the same root name are created.
4	Accept the default **Item Name** and click **OK** to create two discrete input items.
5	Click **Apply** to save the items and leave the page open.
6	Repeat these steps for each group of discrete input items you need to create.

Creating Input Items

To create input items *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* for the STB NIC 2212 example, begin with an input data byte that contains the low byte status for the STB NIC 2212 module:

Step	Action
1	Select the **Input** tab. **NOTE:** In this example, both the **Offset/Device** and **Offset/Connection** columns represent the byte address. The items you create are 8-bit bytes or a 16-bit words.
2	Enter **NIC22212_01_LO_st** in the **Default Item Name Root** field.
3	Select a single row at byte 8.
4	Click the **Define Item(s)** button to open the **Item Name Definition** dialog box.
5	Select **Byte** as the **New Item(s) Data Type**.
6	Click **OK** to create the byte.
7	Click **Apply** to save the items and leave the page open.
8	Repeat these steps to create new byte or word input items.

Creating Output Bit Items

Create output bit items *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* for the STB NIC 2212 example, beginning with two output bits for a STB DDO3200 module:

Step	Action
1	Select the **Output (bit)** tab. **NOTE:** In this example, both the **Offset/Device** and **Offset/Connection** columns represent the byte address of an output. The **Position in Byte** column indicates the bit position (within the byte) of each discrete output item.
2	Enter **DDO3200_out_data** in the **Default Item Name Root** field.
3	Select the rows that correspond to bits 0 and 1 in byte 0 (the first two rows).
4	Click the **Define Item(s)** button to open the **Item Name Definition** dialog box. **NOTE:** An asterisk (*) in the **Item Name** field indicates that discrete items with the same root name are created.
5	Accept the default **Item Name** and click **OK** to create two discrete output items.
6	Click **Apply** to save the items and leave the page open.
7	Repeat these steps to create new output items.

Creating Numeric Output Items

To create output items *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* for the STB NIC 2212, example, beginning with an output data word for the STB AVO 1250 module:

Step	Action
1	Select the **Output** tab. **NOTE:** In this example, both the **Offset/Device** and **Offset/Connection** columns represent the byte address. The items you create will be 16-bit words comprising 2 bytes.
2	Enter **AVO1250_CH1_out_data** in the **Default Item Name Root** field.
3	Starting at the next available whole word, select two rows (rows 2 and 3).
4	Click the **Define Item(s)** button to open the **Item Name Definition** dialog box.
5	Click **OK** to create the output word.
6	Click **Apply** to save the items and leave the page open.
7	Repeat these steps to create a new word for the AVO 1250 channel 2 output data (at bytes 4 and 5).
8	Click **OK** to close the **Items** window.
9	Select **File → Save** to save your edits.

EtherNet/IP Implicit Messaging

Overview

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 BMENOC0301/11 module are accessed through explicit messages or the DTM.

In this situation, these timeout multiplier *(see page 154)* settings are recommended:

RPI (ms)	Recommended Timeout Multiplier	Connection Timeout (ms)
5	32	160
10	16	160
20	8	160
25	4	100

NOTE: If you use values that are lower than those recommended in the table, the network can consume unnecessary bandwidth. That can affect the performance of the module within the system.

Section 8.2
Adding a Modbus TCP Device to the Network

Overview

This section extends the sample Control Expert application. It includes these instructions:

- Add a Modbus TCP module to your Control Expert application.
- Configure the Modbus TCP module.
- Configure a Modbus TCP connection that links the Ethernet communication module and the Modbus TCP module.

NOTE: The instructions in this chapter describe a single, specific device configuration example. Refer to the Control Expert help files for additional information about alternative configuration choices.

What Is in This Section?

This section contains the following topics:

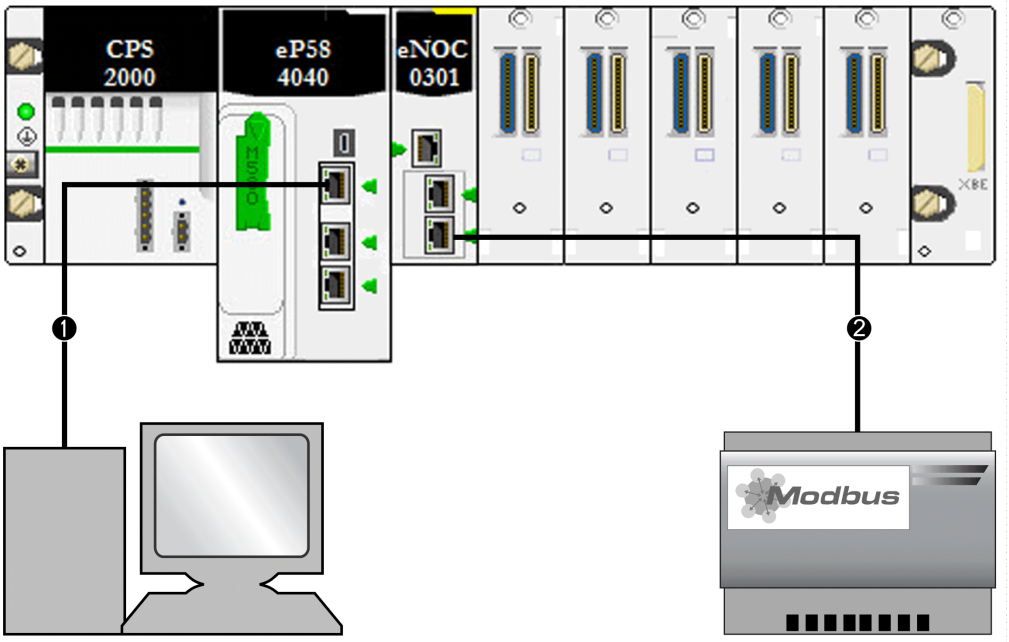
Connection to Modbus TCP Device

Introduction

Use this example to establish communications between the M580 rack and a single-port Modbus TCP device.

Standalone Network Topology

The example shows a generic Modbus TCP device in a simple configuration:



1 An M580 CPU in the local rack is connected to a PC that runs Control Expert.
2 A BMENOC0301/11 Ethernet communications module in the local rack is connected to a generic Modbus TCP device.

To re-create this example, use the IP addresses from your own configuration for these items:

- CPU
- PC
- BMENOC0301/11 Ethernet communication module
- generic Modbus TCP device

NOTE: Control Expert software running in the PC is used to configure the Modicon M580 controller.

Adding a Modbus Device to a Control Expert Project

Overview

Use thes instructions to add a Modbus device to your M580 Control Expert project.

Add the Device

Add a Modbus device to your Control Expert project:

Step	Action
1	Open a Control Expert project that includes a BMENOC0301/11 module *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.
2	Open the **DTM Browser** (**Tools → DTM Browser**).
3	In the **DTM Browser**, right-click the name that you assigned to the BMENOC0301/11 module. *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.
4	Scroll to **Add...** to see the **Add** dialog box.
5	From the **Device** column in the **Add** dialog box, select **Modbus Device**. **NOTE:** This selection (**Modbus Device**) is the generic Modbus DTM. If available, use the manufacturer-specified DTM that corresponds to your particular device.
6	Click **Add DTM** to open the **Properties** window for the Modbus device.
7	On the **General** tab, assign this **Alias name**: **MB1** **NOTE:** Control Expert uses the **Alias name** (**MB1**) as the based name for structure and variable names. No additional editing needs to be performed in the pages of this window. Except for the **Alias name** field, parameters are read-only.
8	Note that the Modbus DTM is added to the BMENOC0301/11 module in the **DTM Browser** as a subnode (<*IP_address*> Modbus:192.68.20.12).
9	Save you configuration (**File → Save**).

The next step is to configure the device you have just added to the project.

Configuring Properties for the Modbus Device

Introduction

Use Control Expert to edit the settings for a Modbus device.

NOTE: To edit these settings, disconnect the DTM from a device *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.

Accessing the Device Properties

For Modbus TCP devices, navigate to the configuration tabs:

Step	Action
1	In the **DTM Browser** (**Tools → DTM Browser**), double-click the DTM that corresponds to the Ethernet communication module that is associated with DTM of the generic Modbus device (**...MB1**). **NOTE:** These instructions assume that you selected **Modbus Device** from the **Add** window when you created a local slave instance *(see page 307)*.
2	In the navigation pane, expand (+) the **Device List** *(see page 136)* to see the associated Modbus TCP and EtherNet/IP devices.
3	Select the Modbus device in this example (**MB1: <MBD:192.168.20.12>**).

These configuration tabs are available for Modbus devices:

- **Properties**
- **Address Setting**
- **Request Setting**

Properties

Configure the **Properties** tab to perform these tasks:

- Add the Modbus device to the configuration.
- Remove the Modbus device from the configuration.
- Edit the base name for variables and data structures used by the Modbus device.
- Indicate how input and output items are created and edited.

The descriptions for parameters *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* in the **Properties** tab are described in the configuration chapter. Use these values and names from the sample configuration:

Field	Parameter	Description
Properties	**Number**	Accept the default.
	Active Configuration	Accept the default (**Enabled**).

Field	Parameter	Description
IO Structure Name	**Structure Name**	Control Expert automatically assigns a structure name based on the variable name, in this case **T_MB1**.
	Variable Name	**Variable Name**: Accept the auto-generated variable name (based on the alias name): **MB1**.
	Default Name	Press this button to restore the default variable and structure names. For this example, custom names are used.
Items Management	**Import Mode**	Select **Manual**.
	Reimport Items	Press this button to import 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**.

Address Setting

When the DHCP client software is enabled in the Modbus device, it obtains its IP address from the DHCP server in the Ethernet communication module.

Configure the **Address Setting** page to perform these tasks:

- Configure the IP address for a device.
- Enable or disable DHCP client software for a device.

NOTE: When the DHCP client software is enabled in a Modbus device, it obtains its IP address from the DHCP server in the Ethernet communication module.

The descriptions for parameters *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* in the **Address Setting** tab are described in the configuration chapter. Use these values and names from the sample configuration:

Field	Parameter	Description
Change Address	**IP Address**	In our continuing example, type in the address **192.168.1.17**.
Address Server	**DHCP for this Device**	Select **Enabled**.
	Identified by	Select **Device Name**.
	Identifier	Accept the default setting of **NIP2212_01** (based on the **Alias name**).
	Subnet Mask	Accept the default value (255.255.255.0).
	Gateway	Accept the default value (0.0.0.0).

The next step is to configure the connection between the communication module and the Modbus device.

Request Setting

Configure the **Request Setting** tab to add, configure, and remove Modbus requests for the Modbus device. Each request represents a separate link between the communication module and the Modbus device.

NOTE: The **Request Setting** tab is available only when a Modbus TCP device is selected in the **Device List**.

These topics for the **Request Setting** tab are described in the configuration chapter *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*:

- Create a Modbus request.
- **Request Setting** parameters
- Remove a Modbus request.

Section 8.3
Configuring the BMENOC0301/11 Module as an EtherNet/IP Adapter

Introduction

This section describes the configuration of the BMENOC0301/11 Ethernet communications module as an EtherNet/IP adapter using local slave functionality.

What Is in This Section?

This section contains the following topics:

Introducing the Local Slave

About Local Slaves

The BMENOC0321 Ethernet communications module scans network modules on behalf of the M580 CPU.

However, you can enable the communications module as an EtherNet/IP adapter (or local slave). When the local slave functionality is enabled, network scanners can access the M580 CPU data that is mapped to local slave assembly objects *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* in the CPU program.

NOTE: The BMENOC0321 module continues to function as a scanner when it is enabled as an EtherNet/IP adapter.

The module supports up to 12 instances of local slaves (**Local Slave 1** ... **Local Slave 12**). Each enabled local slave instance supports these connections:

- one exclusive owner connection
- one listen-only connection

Process Overview

These are the steps in the local slave configuration process:

Stage	Description
1	Enable and configure the BMENOC0321 module as a local slave *(see page 306)*.
2	Configure local slave instances in the scanner device *(see page 307)*. (Local slave instances correspond to each enabled local slave that is scanned.)
3	Specify the size of local slave input and output assemblies in the scanner device (originator). (Use sizes that match the input and output sizes of the enabled local slave.)

Implicit and Explicit Messaging

In its role as an EtherNet/IP adapter, the BMENOC0321 module responds to these requests from network scanners:

- *implicit messages*: Implicit messaging requests are sent from a network scanner device to the communications module. When the local slave functionality is enabled, network scanners can perform these tasks:
  - Read messages from the communications module.
  - Write messages to the communications module.

  Implicit messaging is especially suited to the exchange of peer-to-peer data at a repetitive rate.
- *explicit messages*: The communications module responds to explicit messaging requests that are directed to its CIP objects. When local slaves are enabled by the CPU, explicit messaging requests can access the communications module's CIP assembly instances. (This is a read-only function.)

Scanner Configuration

Configure the scanner:

Configuration	Description
Control Expert	If the scanner device that is used to communicate with the local slave can be configured using Control Expert, use the DTMs that correspond to the BMENOC0321 modules to add those modules to your configuration.
third-party scanner	Third-party EtherNet/IP scanners that access the local slave assembly instances through the BMENOC0321 module do so with respect to the assembly mapping table *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*. That module is delivered with its corresponding EDS file. Third-party scanners can use the contents of the EDS file to map inputs and outputs to the appropriate assembly instances of the BMENOC0321 module.

Local Slave Configuration Example

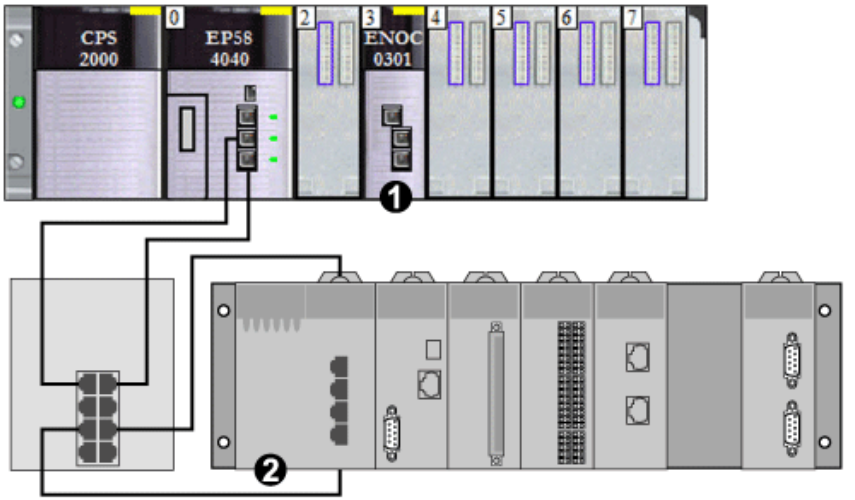
Introduction

Use these instructions to create a simple local slave configuration that includes a network scanner (originator, **O**) and a BMENOC0301 that is enabled as a local slave (target, **T**).

NOTE: This example uses a BMENOC0301 module. Use the same instructions for other M580 communications modules (like the BMENOC0311 or BMENOC0321).

Originator and Target Devices

This simple network shows the enabled local slave and the master device:



1 BMENOC0301: This Ethernet communication module is in slot 3 of the local M580 rack. In this example, you will enable this module as a local slave device (or target, **T**).

2 Modicon M340 rack: In this example, the scanner (or originator, **O**) on this rack scans the CPU data on the M580 rack through the enabled local slave (BMENOC0301).

Enabling Local Slaves

Introduction

In a sample configuration, you will enable **Local Slave 4** and **Local Slave 5**.

First, use these instructions to enable **Local Slave 4** in the BMENOC0301 module configuration. At the end of this exercise, repeat these instructions to enable **Local Slave 5**.

NOTE: This example uses a BMENOC0301 module. Use the same instructions for other M580 communications modules (like the BMENOC0311 or BMENOC0321).

Enabling a Local Slave

Enable the BMENOC0301 module in the M580 local rack as a target device (local slave):

Step	Action
1	Open a Modicon M580 Control Expert project.
2	Add a BMENOC0301 module to slot 3 in the local rack *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.
3	On the **General** tab, assign this **Alias name** to the BMENOC0301 module: BMENOC0301_slot3
4	In the **DTM Browser** (**Tools → DTM Browser**), double-click the DTM that corresponds to the alias name of the BMENOC0301 module to open the configuration window.
5	In the navigation pane, expand (+) **EtherNet/IP Local Slaves** to see the available local slaves.
6	Select a local slave to see its properties. (For this example, select **Local Slave 4**.)
7	In the drop-down list (**Properties → Active Configuration**), scroll to **Enabled**.
8	Press **Apply** to enable **Local Slave 4**.
9	Press **OK** to apply the changes and close the configuration window.

You now have enabled **Local Slave 4** for a BMENOC0301 at IP address 192.168.20.10.

EtherNet/IP scanners that scan the network for the BMENOC0301 at that IP address can use implicit messages to read from and write to the assembly instances that are associated with the local slave instance *(see page 307)*.

Enabling Another Local Slave

This example uses two local slave connections. Make a second connection for **Local Slave 5**:

Step	Action
1	Repeat the steps above to enable a second local slave (**Local Slave 5**). **NOTE:** The appropriate IP address for this example (192.168.20.10) was already assigned to the BMENOC0301 module in the assignment of **Local Slave 4**.
2	Continue to the next procedure to configure the network scanner (originator, **O**).

Accessing Local Slaves with a Scanner

Introduction

Use these instructions to map local slave instances in a network scanner to the enabled local slaves in the BMENOC0301 (**Local Slave 4**, **Local Slave 5**).

NOTE: This example uses a BMENOC0301 module. Use the same instructions for other M580 communications modules (like the BMENOC0311 or BMENOC0321).

In this example, the BMX NOC 0401 Ethernet communication module is a network scanner (originator, **O**) that scans the BMENOC0301 module when it is enabled as a local slave (target, **T**).

Configure the BMX NOC 0401 module in an M340 Control Expert project.

Adding the Device DTM

Create a local slave instance that corresponds to an enabled local slave by name:

Step	Action
1	Open an M340 Control Expert project that includes a BMX NOC 0401 Ethernet communication module.
2	Right-click the BMX NOC 0401 module in the **DTM Browser** (**Tools → DTM Browser**) and scroll to **Add**.
3	Open the **Add** dialog box.
4	Select the DTM that corresponds to the BMENOC0301 module (**BMENOC0301 (from EDS)**). **NOTE:** • The DTM used in this example (**BMENOC0301 (from EDS)**) corresponds to the BMENOC0301 module. For other target devices, use the DTM from the manufacturer that corresponds to your scanner device. • The corresponding input I/O vision and output I/O vision variables are automatically created with the respective suffixes **_IN** and **_OUT**.
5	Press the **Add DTM** button to open the **Properties of device** dialog window.
6	Assign a context-sensitive **Alias name** that corresponds to **Local Slave 4** for the M580 BMENOC0301 module. (For this example, enter BMENOC0301_from_EDS_LS4.)
7	Press **OK** to see the local slave instance in the **DTM Browser**.

Mapping Local Slave Numbers

In the M340 Control Expert project, associate the local slave instances in the BMX NOC 0401 scanner with specific local slaves that are enabled for the BMENOC0301 module:

Step	Action
1	In the **DTM Browser**, double-click the local slave instance that corresponds to **Local Slave 4** in the BMENOC0301 target device (BMENOC0301_from_EDS_LS4). **NOTE:** The default connection is **Local Slave 1 - Exclusive Owner**, which is most applicable to **Local Slave 1** in the target device. It is not appropriate for the local slave instance BMENOC0301_from_EDS_LS4, which is associated with **Local Slave 4** by the assigned context-sensitive name (... _LS4).
2	Select **Local Slave 1 - Exclusive Owner**.
3	Press **Remove Connection** to delete the connection to **Local Slave 1**.
4	Press **Add Connection** to open the dialog box (**Select connection to add**).
5	Scroll to **Local Slave 4 - Exclusive Owner**.
6	Press the **Apply** button.

The local slave (**Local Slave 4**) is now the target of a local slave instance with a context-sensitive connection name (**Local Slave 4 - Exclusive Owner**).

Mapping IP Addresses

Associate the IP address of the local slave (target, **T**) with the local slave instances in the scanner (originator, **O**) configuration:

Step	Action
1	Double-click the BMX NOC 0401 module in the **DTM Browser**.
2	In the navigation pane, expand the **Device List** *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.
3	Select a local slave instance (BMENOC0301_from_EDS_LS4).
4	Select the **Address Setting** tab.
5	In the **IP Address** field, enter the IP address of the local slave device (192.168.20.10).
6	Click in the navigation pane to make the **Apply** button active. **NOTE:** You may have to select **Disabled** in the drop-down menu (**DHCP for this device**) to activate the **OK** and **Apply** buttons.
7	Configure the data size. **NOTE:** Refer to the instructions for configuring input and output items *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.
8	Press **Apply**.

Configuring an Additional Connection

You have created one local slave instance that corresponds by name and IP address to an enabled local slave. That is, the local slave instance BMENOC0301_from_EDS_LS4 in the M340 Control Expert project corresponds to **Local Slave 4** in the M580 Control Expert project.

Because this example uses two local slave connections, you will make another connection (for **Local Slave 5**):

Step	Action
1	Repeat the above steps to create a second local slave instance that corresponds to **Local Slave 5**.
2	Build the Control Expert project.

Accessing the Device DDT Variables

Step	Action
1	In the **Project Browser** (**Tools → Project Browser**) expand **Variables & FB instances**.
2	Double-click **Device DDT Variables** to see the device DDTs that correspond to the BMENOC0301 module in slot 3.

Local Slave Parameters

Accessing the Configuration

Open the **EtherNet/IP Local Slaves** configuration page:

Step	Action
1	Open the Control Expert project that includes a BMENOC0321 module.
2	Open the **DTM Browser** (**Tools → DTM Browser**).
3	In the **DTM Browser**, double-click the name that you assigned to the BMENOC0321 *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* to open the configuration window. **NOTE:** You can also right-click on the module and scroll to **Open** to open the configuration window.
4	Expand (+) **Device List** in the navigation tree to see the local slave instances.
5	Select the local slave instance **BMENOC0321_from_EDS_LS4 <EIP:192.168.20.10>** to view the **Properties** and **Assembly** configuration tabs.

Properties

Identify and enable (or disable) the local slave on the **Properties** tab:

Parameter	Description	
Number	The Control Expert DTM assigns a unique identifier (number) to the device. These are the default values: • *local slave 1*: 112 • *local slave 2*: 113 • *local slave 3*: 114 • ... • *local slave 12*: 123	
Active Configuration	Enabled	Enable the local slave with the configuration information in the **Assembly** fields when the BMENOC0321 module is an adapter for the local slave node.
	Disabled	Disable and deactivate the local slave. Retain the current local slave settings.
Comment	Enter an optional comment (maximum: 80 characters).	
Connection Bit	The auto-generated value in this field represents the association to the local slave in the **Request/Connection Summary** table *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*. **NOTE:** This setting is auto-generated after the local slave settings are edited and the network configuration is saved.	

Assembly

Use the **Assembly** area of the **Local Slave** page to configure the size of the local slave inputs and outputs. Each device is associated with these assembly instances:

- Outputs
- Inputs
- Configuration
- Heartbeat (The heartbeat assembly instance is for listen-only connections only.)

The Control Expert assembly numbers are fixed according to this table, where **O** indicates the originator (scanner) device and **T** indicates the target device:

Local Slave	Number		Connection
	Device	Assembly	
1	112	101	Outputs (T->O)
		102	Inputs (O->T)
		103	Configuration Size
		199	Heartbeat
2	113	111	Outputs (T->O)
		112	Inputs (O->T)
		113	Configuration Size
		200	Heartbeat
3	114	121	Outputs (T->O)
		122	Inputs (O->T)
		123	Configuration Size
		201	Heartbeat
4	115	131	Outputs (T->O)
		132	Inputs (O->T)
		133	Configuration Size
		202	Heartbeat
5	116	136	Outputs (T->O)
		137	Inputs (O->T)
		138	Configuration Size
		202	Heartbeat
6	117	141	Outputs (T->O)
		142	Inputs (O->T)
		143	Configuration Size
		202	Heartbeat

Local Slave	Number		Connection
	Device	Assembly	
7	118	146	Outputs (T->O)
		147	Inputs (O->T)
		148	Configuration Size
		202	Heartbeat
8	119	151	Outputs (T->O)
		152	Inputs (O->T)
		153	Configuration Size
		202	Heartbeat
9	120	156	Outputs (T->O)
		157	Inputs (O->T)
		158	Configuration Size
		202	Heartbeat
10	121	161	Outputs (T->O)
		162	Inputs (O->T)
		163	Configuration Size
		202	Heartbeat
11	122	166	Outputs (T->O)
		167	Inputs (O->T)
		168	Configuration Size
		202	Heartbeat
12	123	171	Outputs (T->O)
		172	Inputs (O->T)
		173	Configuration Size
		202	Heartbeat

NOTE: When using explicit messaging to read the BMENOC0321 module's assembly instance, you need to allocate sufficient room for the response. The size of the response equals the sum of: assembly size + 1 byte (Reply service) + 1 byte (General Status)

Limitations (from the perspective of the local slave):

- *maximum RPI value*: 65535 ms
- *maximum timeout value*: 512 * RPI
- *outputs (T->O)*: 509 bytes maximum
- *inputs (O->T)*: 505 bytes maximum
- *configuration for the Ethernet communication module*: 0 (fixed)

Working with Device DDTs

Introduction

Use Control Expert to create a collection of device derived data types (DDDTs) and variables that support communications and the transfer of data between the PAC and the various local slaves, distributed devices, and corresponding I/O modules.

You can create DDDTs and corresponding variables in the Control Expert DTM to support your network design.

Use the DDDTs for these tasks:

- Read status information from the Ethernet communication module.
- Write control instructions to the Ethernet communication module.

You can double-click the name of the DDDT in the **Project Browser** at any time to view its properties and open the corresponding EDS file.

View the Device DDTs

View the DDDT characteristics of the BMENOC0321 module in Control Expert:

Step	Action
1	In a Control Expert project, add a BMENOC0321 module *(see page 53)*.
2	Build the Control Expert project.
3	In the **Variables & Feedback Instances** tab, view the variables *(see page 187)*.

These are the default characteristics of the BMENOC0321 module in the **Variables** tab:

- default variable name: BMENOC0321 (Unity Pro v11.0)
- default variable type: T_BMENOC0321 (Unity Pro v11.0)

NOTE: For applications that require multiple DDDTs, create an **Alias name** that logically identifies the DDDT with the configuration (module, slot, local slave number, etc.).

DDDT Variables

You can access the DDDTs and the corresponding variables in Control Expert and add them to a user-defined **Animation Table**. Use that table to monitor read-only variables and edit read-write variables.

Use these data types and variables to perform these tasks:

- Read the status of connections and communications between the Ethernet communication module and distributed EtherNet/IP and Modbus TCP devices:
  - The status is displayed in the form of a HEALTH_BITS array consisting of 32 bytes.
  - A bit value of 0 indicates the connection is lost or the communication module can no longer communicate with the distributed device.

- Toggle a connection ON (1) or OFF (0) by writing to a selected bit in a 16-word DIO_CONTROL array
- Monitor the value of local slave and distributed device input and output items that you created in Control Expert.

Displaying the Order of Input and Output Items

In the **Project Browser**, view the DDDTs.

The **Data Editor** displays each input and output variable. When you open the first input and output variables, you can see both the connection health bits (DIO_HEALTH) and the connection control (DIO_CTRL) bits.

This table shows the rule assignment for connection numbers:

Inputs	Order	Outputs
health bits (note 1)	1	control bits (note 1)
Modbus TCP input variables (note 2)	2	Modbus TCP output variables (note 2)
local slave input variables (note 3)	3	local slave output variables (note 3)
EtherNet/IP input variables (note 2)	4	EtherNet/IP output variables (note 2)

NOTE 1: Health and control bits are in this format:
- **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)

NOTE 2: Device variables are in this format:
- **i.** by device number
- **ii.** within a device (by connection number)
- **iii.** within a connection (by item offset)

NOTE 3: Local slave variables are in this format:
- **i.** by local slave number
- **ii.** within each local slave (by item offset)

Section 8.4
Accessing Device DDT Variables

Device DDTs and Scanned Devices

Introduction

You can access the device DDT for EtherNet/IP and Modbus TCP devices that are scanned by the Ethernet communication module after you perform one of these tasks:

- Add an EtherNet/IP device to the network *(see page 278)*.
- Add a Modbus TCP device to the network *(see page 296)*.
- Configure the Ethernet communication module as an EtherNet/IP adapter *(see page 302)*.

Access the Device DDT Variables

Access the device DDT for the Ethernet communication module in Control Expert:

Step	Action
1	Open the Control Expert **Project Browser** (**Tools → Project Browser**).
2	Expand (+) **Variables & FB instances**.
3	Double-click **Device DDT Variables**.

You can add the variable to an Animation Table *(see page 163)* to read the status and set the device control bit.

NOTE: The red arrow and lock icons in the **Device DDT** table indicate that the variable name was auto-generated by Control Expert based on the configuration of the communication module, local slave, or distributed device. (You cannot edit the variable name.)

This table describes the input and output bits associated with the EtherNet/IP and Modbus TCP devices:

Name	Description
Freshness	This is a global bit: ● **1:** All input objects below (**Freshness_1**, **Freshness_2**, etc.) for the associated device are true (**1**) and provide up-to-date data. ● **0:** One or more inputs (below) is not connected and does not provide up-to-date data.
Freshness_1	These bits represent individual input objects for the device: ● **1:** The input object in this row is connected and provides up-to-date data. ● **0:** The input object is not connected and does not provide up-to-date data.
Freshness_2 **Freshness_3** ...	These bits represent individual input objects for the device: ● **1:** The input object in this row is true (**1**) and provides up-to-date data. ● **0:** The input object is not connected (**0**) and does not provide up-to-date data.
(available)	The rows after the **Freshness** data are organized in groups of **Inputs** and **Outputs** that have user-defined names. The number of input and output rows depends on the number of input and output requests configured for a particular device.

Section 8.5
Hardware Catalog

Introduction

The Control Expert **Hardware Catalog** displays the modules and devices that you can add to a Control Expert project. Each module or device in the catalog is represented by a DTM that defines its parameters.

What Is in This Section?

This section contains the following topics:

Introduction to the Hardware Catalog

Introduction

The Control Expert **Hardware Catalog** contains a list of modules and devices that you can add to a Control Expert project. EtherNet/IP and Modbus TCP devices are located in the **DTM Catalog** tab at the bottom of the **Hardware Catalog**. Each module or device in the catalog is represented by a DTM that defines its parameters.

EDS Files

Not all devices in today's market offer device-specific DTMs. Some devices are defined by device-specific EDS files. Control Expert displays EDS files in the form of a DTM. In this way, you can use Control Expert to configure devices that are defined by an EDS file in the same way you would configure a device defined by its DTM.

Other devices lack both a DTM and an EDS file. Configure those devices by using the generic DTM on the **DTM Catalog** page.

View the Hardware Catalog

Open the Control Expert **Hardware Catalog**:

Step	Action
1	Open Control Expert.
2	Find the **PLC bus** in the **Project Browser**.
3	Use one method to open the catalog: • Use the pull-down menu (**Tools** → **Hardware Catalog**). • Double-click an empty slot in the **PLC bus**.

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 PC that is running Control Expert).

The installation process for the DTM is defined by the device manufacturer. Consult the documentation from the device manufacturer 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 to see the new DTM in the catalog. The DTM can then be added to a Control Expert project.

Adding an EDS File to the Hardware Catalog

Introduction

You may want to use an EtherNet/IP device for which no DTM is in the catalog. In that case, use these instructions to import the EDS files into the catalog to create a corresponding DTM.

Control Expert includes a wizard you can use to add one or more EDS files to the Control Expert **Hardware Catalog**. The wizard presents instruction screens to execute these commands:

- Simplify the addition of EDS files to the **Hardware Catalog**.
- Provide a redundancy check when you add duplicate EDS files to the **Hardware Catalog**.

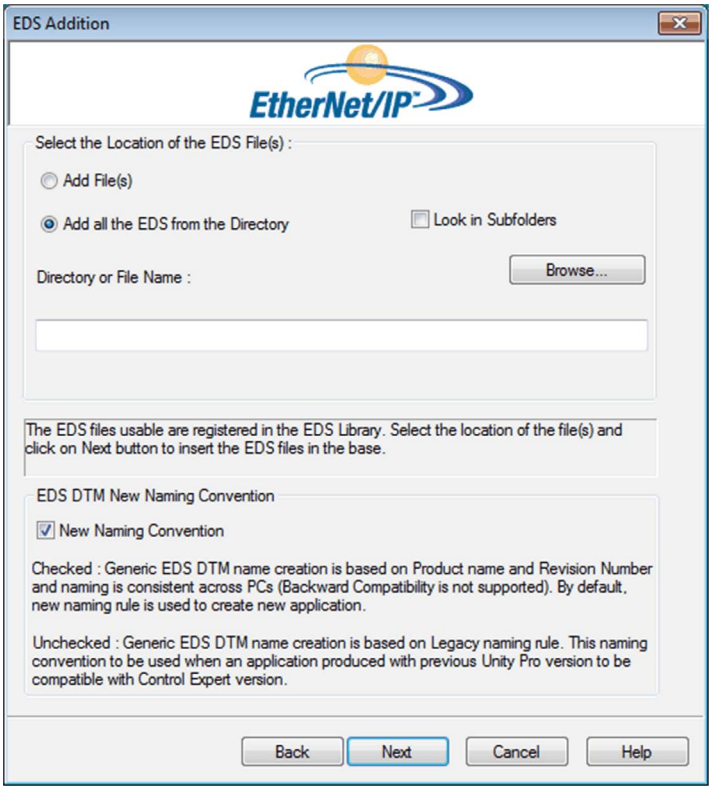
NOTE: The Control Expert **Hardware Catalog** displays a partial collection of DTMs and EDS files that are registered with the ODVA. This library includes DTMs and EDS files for products that are not manufactured or sold by Schneider Electric. The non-Schneider Electric EDS files are identified by vendor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

Adding EDS Files

Open the **EDS Addition** dialog box:

Step	Action
1	Open a Control Expert project that includes an Ethernet communication module.
2	Open the **DTM Browser** (**Tools → DTM Browser**).
3	In the **DTM Browser**, select a communication module.
4	Right-click on the communication module and scroll to **Device menu → Additional functions → Add EDS to library**.
5	In the **EDS Addition** window, click **Next**.

You can now see this page:



Add one or more EDS files to the library:

Step	Action
1	Use these commands in the **Select the Location of the EDS File(s)** area of the **EDS Addition** dialog box to identify the location of the EDS files: ● **Add File(s)**: Add one or more EDS files that are individually selected. ● **Add all the EDS from the Directory**: Add all files from a selected folder. (Check **Look in Subfolders** to add EDS files from the folders within the selected folder.)
2	Click **Browse** to open a navigation dialog box.
3	Select the location of the EDS file(s): ● Navigate to at least one EDS file. ● Navigate to a folder that contains EDS files. **NOTE:** Keep the location selected (highlighted).
4	Click **Select** to close the navigation window. **NOTE:** Your selection appears in the **Directory or File Name** field.

Step	Action
5	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 in the library is 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.
6	Click **Next** to compare the selected EDS files to the files in the library. **NOTE:** If one or more selected EDS files is a duplicate, a **File Already Exists** message appears. Click **Close** to hide the message.
7	The next page of the **EDS Addition** wizard opens. It indicates the status of each device you attempted to add: ● check mark (green): The EDS file can be added. ● informational icon (blue): There is a redundant file. ● exclamation point (red): There is an invalid EDS file. **NOTE:** You can click **View Selected File** to open and view the selected file.
8	Click **Next** to add the non-duplicate files. **Result**: The next page of the **EDS Addition** wizard opens to indicate that the action is complete.
9	Click **Finish** to close the wizard. **Result**: The hardware catalog automatically updates.

Removing an EDS File from the Hardware Catalog

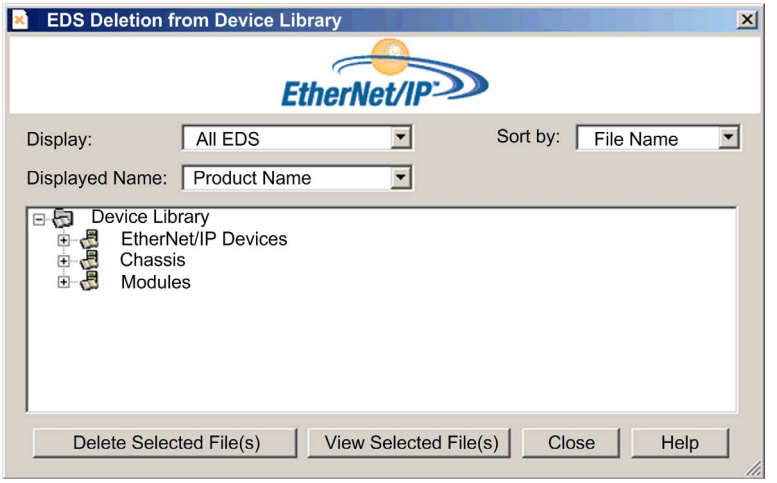
Introduction

You can remove a module or device from the list of available devices in the Control Expert **Hardware Catalog** by removing its **EDS** file from the library.

When you remove an EDS file from the library, the device or module disappears from the **DTM Catalog**. However, removing the file from the library does not delete the file from its stored location, so you can import the file again later.

Removing an EDS File from the Catalog

Use these steps to remove an EDS file from the catalog:

Step	Action
1	Open the Control Expert **DTM Browser** (**Tools → DTM Browser**).
2	In the **DTM Browser**, select an Ethernet communication module.
3	Right-click the module and scroll to **Device menu → Additional functions → Remove EDS from library** to open the **EDS Deletion from Device Library** window: 

Step	Action	
4	Use the selection lists in the heading of this window to specify how EDS files are displayed:	
	Display	Choose criteria to filter the list of EDS files: • **All EDS** (no filtering) • **Only Devices** • **Only Chassis** • **Only Modules**
	Sort by	Choose criteria to sort the list of displayed EDS files: • **File Name** • **Manufacturer** • **Category** • **Device Name**
	Displayed Name	Choose the identifier for each device: • **Catalog Name** • **Product Name**
5	Expand (+) the **Device Library** navigation tree and select the EDS file you want to remove. **NOTE:** Click **View Selected File** to see the read-only contents of the selected EDS file.	
6	Click the **Delete Selected File(s)** button to open the **DeleteEDS** dialog box.	
7	Click **Yes** to remove the selected EDS file from the list.	
8	Repeat these steps for each EDS file you want to delete.	
9	Click **Finish** to close the wizard. **Result**: The hardware catalog automatically updates.	

Export / Import EDS Library

Introduction

To use the same project on two Control Expert installations (for example a source, and a target Host PCs), you may have to update the DTM **Hardware Catalog** of the target Host PC.

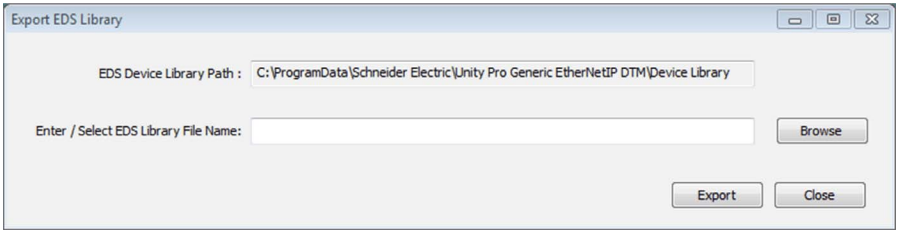
Instead of adding one by one the missing EDS files in the target Host PC, you can update the DTM **Hardware Catalog** in two steps:

- Exporting the EDS library from the source Host PC.
- Importing the EDS library in the target Host PC.

NOTE: When you export the EDS library, the software generates an **.DLB** file which contains all the DTM created form EDS files.

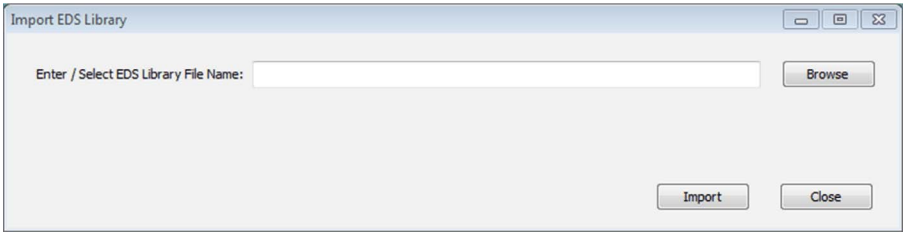
Exporting EDS Library

Open the **Export EDS Library** dialog box:

Step	Action
1	Open a Control Expert project that includes an Ethernet communication module.
2	Open the **DTM Browser** (**Tools → DTM Browser**).
3	In the **DTM Browser**, select a communication module.
4	Right-click on the communication module and scroll to **Device menu → Additional functions → Export EDS library** to open the **Export EDS library** window: 
5	For the archived EDS library you want to create: ● Enter the full folder path along with the file name in the **Enter / Select EDS Library File Name** field, or ● Click **Browse** to open a navigation dialog box: ❍ Select the location, and ❍ Enter the file name, and ❍ Click **Save** to close the navigation window and your selection appears in the **Enter / Select EDS Library File Name** field.
6	Click **Export** to create the archived EDS library. **Result:** A new wizard opens to indicate that the export is complete. Click **Ok** to close the wizard.
7	In the **Export EDS library** window, click **Close**.

Importing EDS Library

Use these steps to import an archived EDS library:

Step	Action
1	Open the Control Expert **DTM Browser** (**Tools → DTM Browser**).
2	In the **DTM Browser**, select an Ethernet communication module.
3	Right-click the module and scroll to **Device menu → Additional functions → Import EDS library** to open the **Import EDS library** window: 
4	For the archived EDS library you want to import: ● Enter the full folder path along with the file name in the **Enter / Select EDS Library File Name** field, or ● Click **Browse** to open a navigation dialog box: ❍ Select the location, and ❍ Enter the file name, and ❍ Click **Save** to close the navigation window and your selection appears in the **Enter / Select EDS Library File Name** field.
5	Click **Import**. **Result:** A new wizard opens to indicate that the export is complete. Click **Ok** to close the wizard.
6	In the **Import EDS library** window, click **Close**.

Section 8.6
Managing Connection Bits

Connection Health Bits and Connection Control Bits

Introduction

Use these instructions to configure these bits:

- *connection health bits*: Display the status of each device with one or more connections.
- *connection control bits*: Toggle each connection on and off using object IDs.

Identifying the Connection Health Bits

For the Ethernet communications module, discover the health bit that is mapped to a specific distributed device.

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

- 1: All connections that are configured for the device are active.
- 0: One or more connections that are configured for the device are not active.

In the Control Expert **Project Browser**, double-click **Variables & FB instances** to view health bits in an 8-word array.

EtherNet/IP Connection Health Bits

For EtherNet/IP devices, navigate to a connection node:

Step	Action
1	In the DTM Browser (**Tools → DTM Browser**), double-click the DTM that corresponds to the appropriate Ethernet communications module.
2	In the navigation pane, expand the **Device List**.
3	Select the connection that corresponds to a node in the **Device List**.
4	Select the **Connection Settings** tab.
5	Locate the value in the **Connection Bit** row. **NOTE:** For example, a **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 `HEALTH_BITS_IN[0].2`.

NOTE: To diagnose the device health, refer to the device DDTs for the Ethernet communication module *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.

Modbus TCP Connection Health Bits

For Modbus TCP devices, navigate to the main device node:

Step	Action
1	In the DTM Browser (**Tools → DTM Browser**), double-click the DTM that corresponds to the appropriate communications module. **NOTE:** These instructions assume that you selected **Modbus Device** from the **Add** window when you created a local slave instance *(see page 307)*.
2	In the navigation pane, expand the **Device List** *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* to see the associated Modbus TCP devices.
3	Select a Modbus TCP device.
4	Select the **Request Setting** tab.
5	Locate the value in the **Connection Bit** column. **NOTE:** For example, a **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 `HEALTH_BITS_IN[0].0`.

Access the Modbus connection settings :

Step	Action
1	In the **DTM Browser**, select a communications module for which you have configured a Modbus device.
2	Double-click the communications module to open the configuration window.
3	In the navigation pane, expand the **Device List**.
4	Select the Modbus device.
5	Select the **Request Setting** tab.
6	Configure requests: ● *Add a request*: Click **Add Request** to see the request data in the next available row. ● *Remove a request*: Click the row that corresponds to the request you want to remove and click **Remove**. **NOTE:** When you add or remove a request, the corresponding request in the navigation pane (**Request 001: Items**; **Request 002: Items**; **Request 003: Items**; etc.) appears or disappears. You can select a request to configure its input data.
7	Click **Apply**. **NOTE:** You can add or remove multiple requests before you click **Apply**.

Monitoring Connection Health Bits in an Animation Table

Use an animation table to monitor the status of connection health bits and other variables. Add health bits to an animation table:

Step	Action
1	In the **Project Browser**, right-click **Animation Tables** and scroll to **New Animation Table**.
2	In the **New Animation Table**, type these values for these fields: • **Name**: Connection_Health_Bits • **Number of animated characters**: Accept the default (100).
3	Click **OK** to open the **Connection_Health_Bits** animation table
4	Double-click the first empty row in the **Name** column.
5	Click the ellipsis (**...**) button to open the **Instance Selection** dialog box.
6	Find the health bits and select the entire array.
7	Click **OK** to add the array to the **Connection_Health_Bits** animation table. **NOTE:** Remember that each row represents a word that contains 16 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 word.

Elsewhere in this guide are these instructions:

- Modify connection control bits in an animation table *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.
- Display the order of input and output Items *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.

Chapter 9
Firmware Update

Introduction

This chapter describes the steps for updating the firmware for the BMENOC0301/11 Ethernet communications module.

What Is in This Chapter?

This chapter contains the following topics:

Firmware Update with Automation Device Maintenance

Overview

The EcoStruxure™ Automation Device Maintenance is a standalone tool that allows and simplifies the firmware update of devices in a plant (single or multiple).

The tool supports the following features:

- Automatic device discovery
- Manual device identification
- Certificate management
- Firmware update for multiple devices simultaneously

NOTE: For a description of the download procedure, refer to the *EcoStruxure™ Automation Device Maintenance, User Guide*.

Firmware Update with Unity Loader

Introduction

You can update the firmware on the Ethernet communications module by downloading a new firmware version with Unity Loader.

The firmware download can be performed by connecting to the Ethernet network through ETH 1.

Refer to *Unity Loader, User Guide* for a description of the download procedure.

Enabling the Update

To enable the firmware update, check the security settings *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.

Firmware File

The firmware file is a **.ldx* file.

Procedure

Update the firmware for the Ethernet communications module and the BMEXBP••00 rack:

Step	Action
1	Install Unity Loader software.
2	Connect the PC that is running Unity Loader to the Ethernet communications module.
3	Launch Unity Loader.
4	Click **Firmware** tab.
5	In the **PC** list box, select the *.ldx* file that contains the firmware file.
6	When connected with Ethernet, check that the MAC address indicated in the **PLC** box corresponds to the MAC address marked on the connected device (PLC or Ethernet communication module).
7	Check that transfer sign is green to allow transfer from PC to connected device.
8	Click **Transfer**.
9	Click **Close**.

Chapter 10
BMENOC0321 Control Module Web Pages

Introduction

Standard Web Features: Like all Modicon M580 devices, the BMENOC0321 control network module supports a standard set of web pages. These pages provide tools to diagnose the basic functionality of the modules. The standard web site is not customizable.

FactoryCast Web Features: Some Modicon M580 devices, like the BMENOC0321 control network module, use an expanded set of customizable web features called FactoryCast. The FactoryCast web site supports all of the features in the standard web site and many advanced features. You can customize the pages on the FactoryCast web site.

What Is in This Chapter?

This chapter contains the following sections:

Section 10.1
Modicon M580 Standard Web Site

Introduction

An HTTP server transmits standard web pages for monitoring and diagnosing the communications module. The server provides easy access to the Ethernet communications module from standard Internet browsers.

What Is in This Section?

This section contains the following topics:

Introducing the Embedded Web Pages

Introduction

Use the web pages to perform diagnostics for the BMENOC0321 Ethernet communications module to display real-time diagnostic data for both the BMENOC0321 communications module and other networked devices.

Open the Web Page

Access the **Diagnostics** tab:

Step	Action
1	Open an Internet browser.
2	In the address bar, enter the IP address of the Modicon M580 communications module.
3	Press **Enter**.

Menu Items

Expand the menu on the **Diagnostics** tab to access this diagnostics information:

Menu Items		Description
Module	Status Summary *(see page 339)*	View status information for the communications module.
	Performance *(see page 341)*	View performance statistics for the communications module.
	Port Statistics *(see page 342)*	View statistics for each port on the communications module.
Connected Devices	I/O scanner *(see page 344)*	View the scanner status and connection statistics for the communications module.
	Messaging *(see page 346)*	View current information for open Modbus TCP connections on port 502.
Services	QoS *(see page 347)*	View information about the QoS service.
	NTP *(see page 349)*	View the operating parameters for the network time service.
	Redundancy *(see page 351)*	View the configured values for the RSTP configuration of the communications module.
	E-mail *(see page 353)*	View the diagnostics information that correspond to the e-mail service.
System	Alarm viewer *(see page 355)*	View the diagnostics information that correspond to running services and communications module operations:

Software Requirements

The embedded web server in the M580 CPUs displays data in standard HTML web pages.

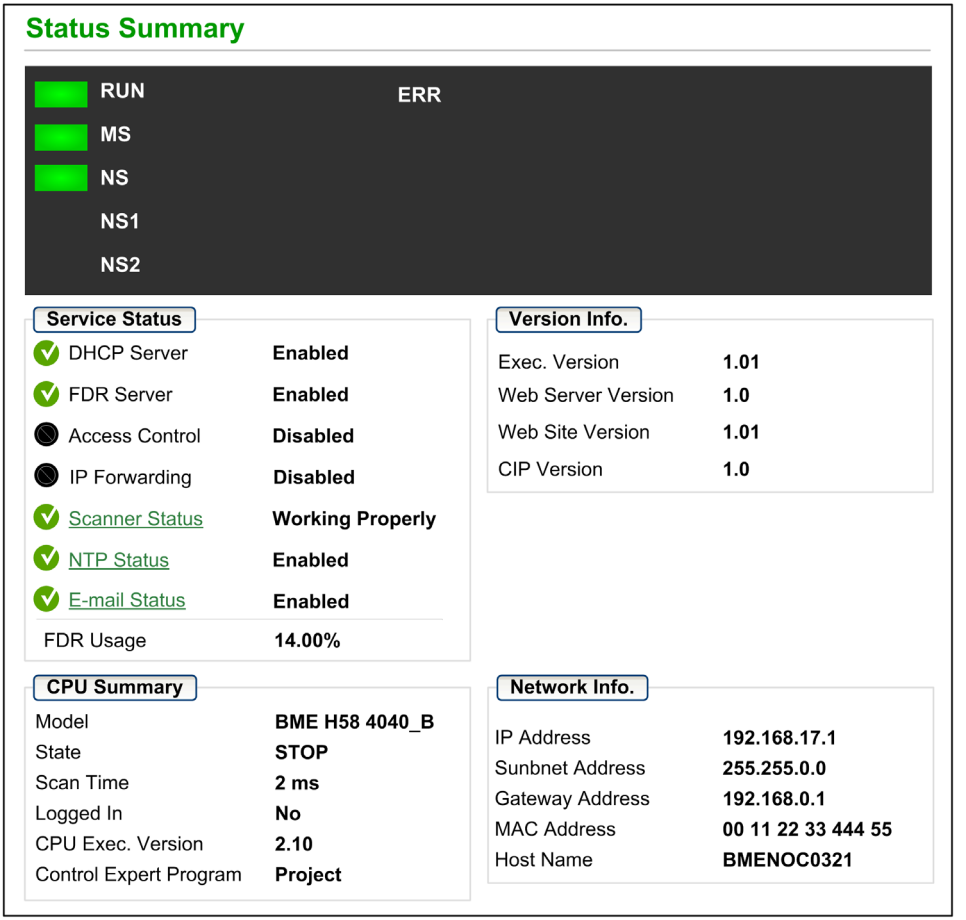
Observe these requirements to access embedded web pages with a PC, iPad, or Android tablet:

	Application	Requirement
Browser (in order of recommendation)	Google Chrome	v11 or later
	Mozilla Firefox	v4 or later
	Internet Explorer	v8 or later
	Safari	v5.1.7 or later
Browser Plug-in	Java	version 1.7u51 or later
	Microsoft Silverlight	v5 or later

Status Summary

Open the Page

Access the **Status Summary** page on the **Diagnostics** tab (**Menu → Module → Summary**):



Diagnostic Information

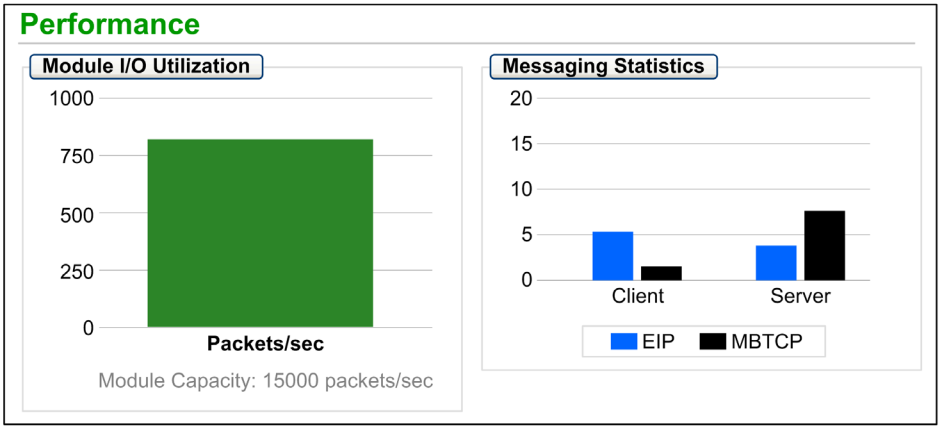
The objects on this page provide status information:

Parameters	Description	
LEDs	The black field contains LED indicators (**RUN**, **ERR**, etc.). **NOTE:** Refer to the description of LED activity and indications *(see page 184)*.	
Service Status	green	The available service is operational and running.
	red	An error is detected in an available service.
	black	The available service is not present or not configured.
Version Info.	This field describes the software versions that are running on the Ethernet communications module.	
CPU Summary	This field describes the CPU hardware and the applications that are running on the CPU.	
Network Info.	This field contains network and hardware address information and connectivity that corresponds to the Ethernet communications module.	

Performance

Open the Page

Access the **Performance** page from the **Diagnostics** tab (**Menu → Module → Performance**):



NOTE:

- Move the mouse over the dynamic graphs to see the current numeric values.
- This page is updated every 5 seconds.

Diagnostic Information

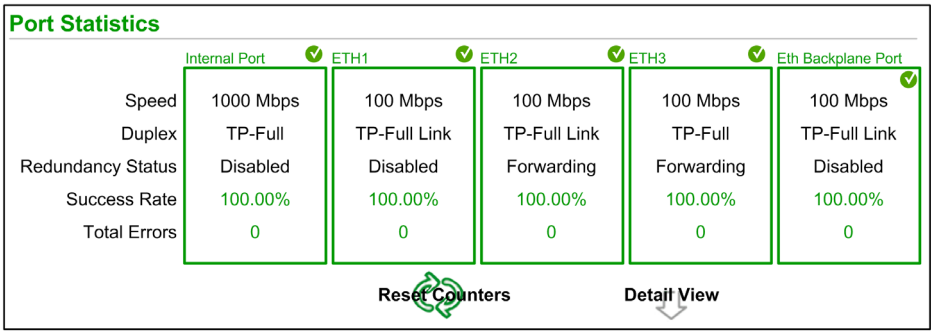
This table describes the performance statistics:

Field	Description
Module I/O Utilization	This graph shows the total number of packets (per second) the communications module can handle at once.
Messaging Statistics	This graph shows the number of Modbus/TCP or EtherNet/IP messages per second for the client or server.

Port Statistics

Open the Page

Access the **Port Statistics** page from the **Diagnostics** tab (**Menu → Connected Devices → Port Statistics**):



NOTE: This page is updated every 5 seconds. Click **Reset Counters** to reset all dynamic counters to 0.

Diagnostic Information

This page shows the statistics for each port on the communications module. This information is associated with the configuration of the Ethernet ports *(see page 93)* and the configuration of the SERVICE port *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.

The frame color indicates the port activity:

- *green*: active
- *gray*: inactive
- *yellow*: error detection
- *red*: error detection

View these statistics:

Statistic	Description
Speed	the configured port speed (0, 100, 1000 Mbps)
Duplex	The current duplex mode is composed of some combination of these elements: ● TP/Fiber ● -Full/-Half/-None ● Link/(no word) **NOTE:** When the thirteenth bit of the word in the Modbus response is 1, **Link** is added to the duplex mode string (**TP-Full Link**, **TP-Half Link**, etc.).

Statistic	Description
Redundancy Status	The Ethernet port is: ● learning or forwarding information ● discarding information ● disabled
Success Rate	successful transmissions (percentage)
Total Errors	number of detected errors

Expanded View

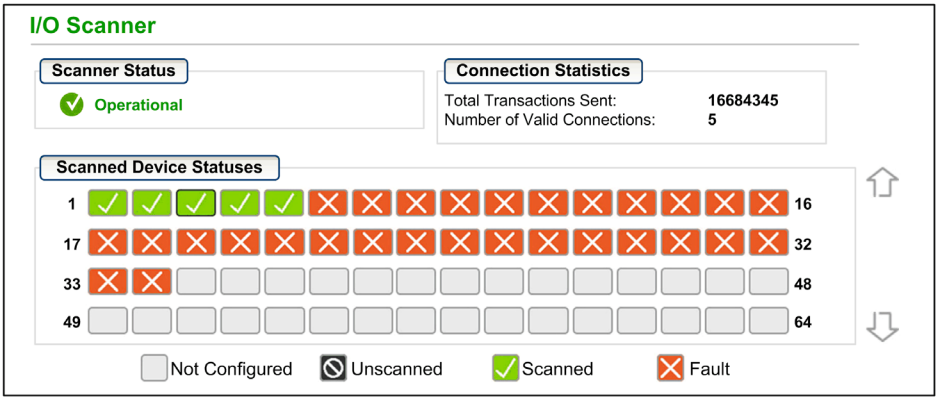
Click **Detail View** to see more statistics:

Statistic	Description
Frames Transmitted	Number of frames successfully transmitted
Frames Received	Number of frames received
Excessive Collisions	Number of excessive Ethernet collisions
Late Collisions	Number of late Ethernet collisions
CRC Errors	Number of detected cyclic redundancy check errors
Bytes Received	Number of bytes received
Inbound Packet Errors	Number of detected inbound packet errors
Inbound Packets Discarded	Number of inbound packets discarded
Bytes Transmitted	Number of bytes transmitted
Outbound Packet Errors	Number of detected outbound packet errors
Outbound Packets Discarded	Number of outbound packets discarded
Carrier Sense Errors	Number of detected carrier sense errors. A carrier sense error is detected when a port tries to transmit a frame, but cannot do so because no carrier is detected.
FCS Errors	Number of detected frame check sequence (FCS) errors. An FCS error is detected when a frame is corrupted during transmission as indicated by its checksum value.
Alignment Errors	The number of byte alignment errors that have been detected. A byte alignment occurs when the number of bits in a frame is not divisible by 8. An alignment error also triggers an FCS error.
Internal MAC Trans. Errors	The number of detected transmit errors that are not late collisions, excessive collisions, or CRC errors.
Internal MAC Rec. Errors	The number of detected receive errors that are not late collisions, excessive collisions, or CRC errors.
SQE Test Errors	The number of detected signal quality error (SQE) instances. Some Ethernet transceivers use an SQE heartbeat to indicate it is connected to a host interface. This detected error indicates that a transceiver has no heartbeat. Note that not all transceivers produce a heartbeat.

I/O Scanner

Open the Page

Access the **I/O Scanner** page from the **Diagnostics** tab (**Menu → Connected Devices → I/O Scanner**):



NOTE: This page is updated every 5 seconds.

Diagnostic Information

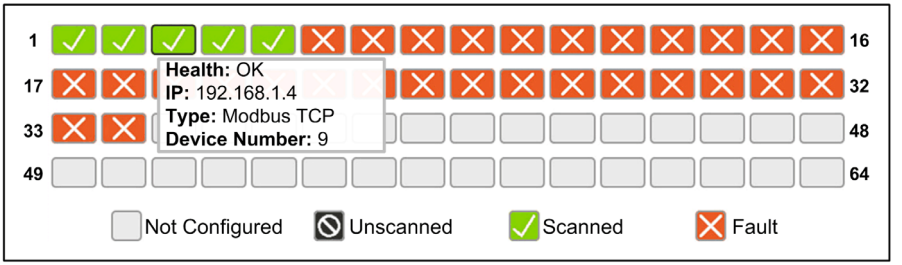
This table describes the scanner status and connection statistics:

Scanner Status	**Operational**	The I/O scanner is enabled.
	Stopped	The I/O scanner is disabled.
	Idle	The I/O scanner is enabled but not running.
	Unknown	The I/O scanner returns unexpected values from the device.
Connection Statistics	**Transactions per Second**	
	Number of Connections	

In the **Scanned Device Status** display, the colors that appear in each block indicate these states for specific remote devices:

Color	Indication	Status
gray	**Not Configured**	There is an unconfigured device.
black	**Unscanned**	The scanning of the specific device has been intentionally disabled.
green	**Scanned**	A device is being scanned successfully.
red	**Fault**	A device that is being scanned is returning detected errors.

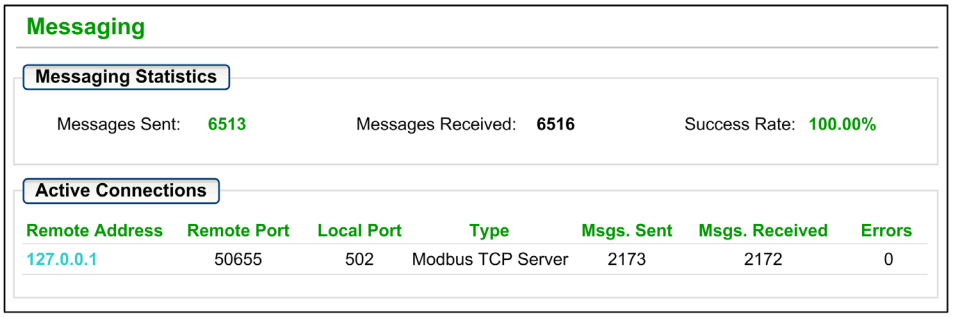
Hold the cursor over any block to get information for a specific device:



Messaging

Open the Page

Access the **Messaging** page from the **Diagnostics** tab (**Menu → Connected Devices → Messaging**):



NOTE: This page is updated every 5 seconds.

Diagnostic Information

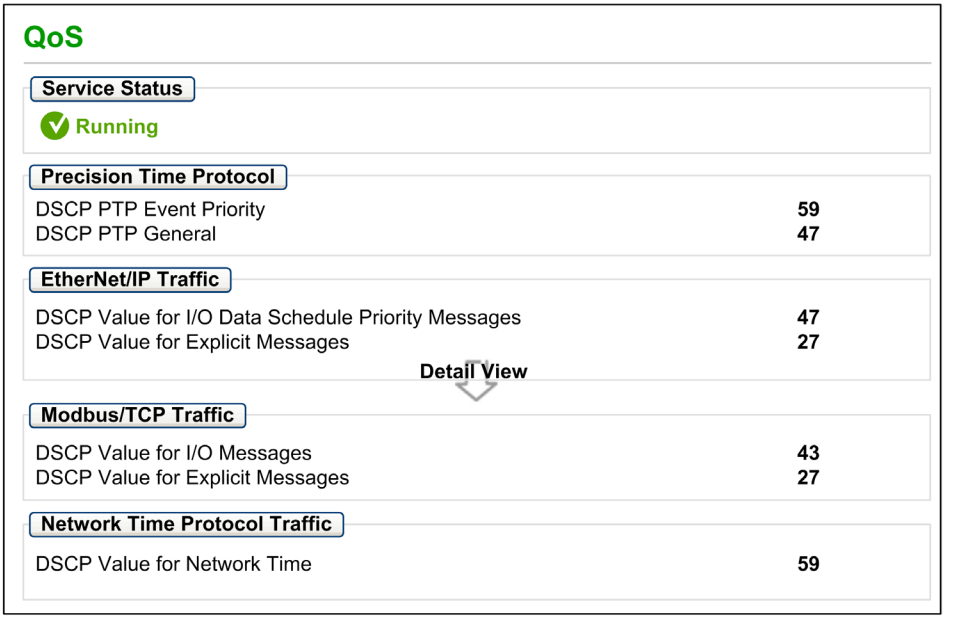
This page shows current information for open Modbus TCP connections on port 502:

Field	Description
Messaging Statistics	This field contains the total number of sent and received messages on port 502. These values are not reset when the port 502 connection is closed. Therefore, the values indicate the number of messages that have been sent or received since the module was started.
Active Connections	This field shows the connections that are active when the **Messaging** page is refreshed.

QoS

Open the Page

Access the **QoS** (quality of service) page from the **Diagnostics** tab (**Menu → Services → QoS**):



NOTE:

- Configure the QoS in Control Expert *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.
- Click **Detail View** to expand the list of parameters.
- This page is updated every 5 seconds.

Service Status

This table shows the possible states for the **Service Status**:

Status	Description
Running	The service is correctly configured and running.
Disabled	The service is disabled.
Unknown	The status of the service is not known.

Diagnostic Information

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:

Field	Parameter	Description
Precision Time Protocol (see note)	DSCP PTP Event Priority	PTP time synchronization.
	DSCP PTP General	PTP general.
EtherNet/IP Traffic	DSCP Value for I/O Data Scheduled Priority Messages	Configure the priority levels to prioritize the management of data packets.
	DSCP Value for Explicit Messages	
Modbus/TCP Traffic	DSCP Value for I/O Messages	—
	DSCP Value for Explicit Messages	
Network Time Protocol Traffic	DSCP Value for Network Traffic	—
NOTE: The Precision Time Protocol QoS attributes are 2 and 3 (class 48h, instance 1). Use these attributes to obtain QoS values for the Precision Time Protocol.		

Considerations

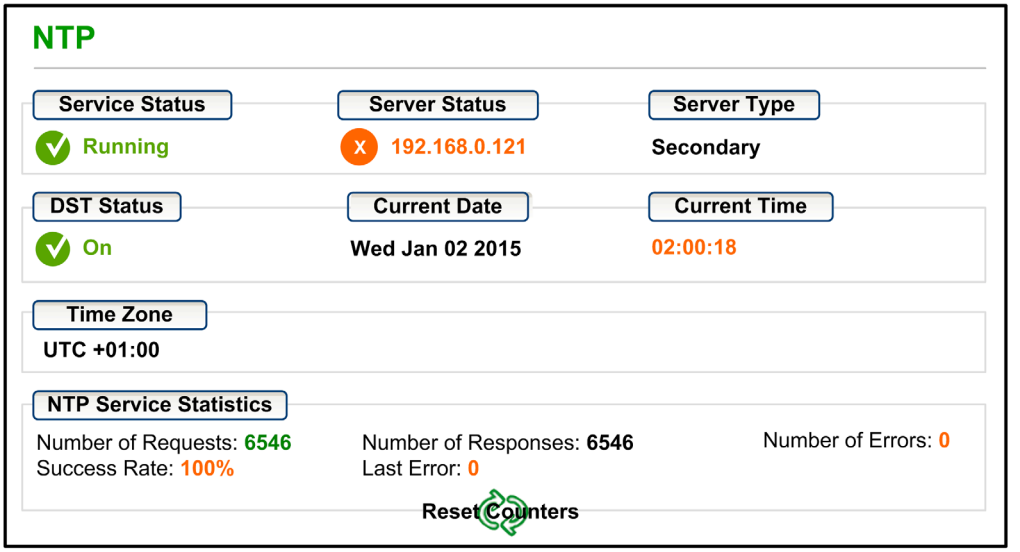
Take measures to effectively implement QoS settings in your Ethernet network:

- Use only network switches that support QoS.
- Apply the same DSCP values to all network devices and switches.
- Use switches that apply a consistent set of rules for handling the different DSCP values when transmitting and receiving Ethernet packets.

Network Time Service

Open the Page

Access the **Network Time Service** page from the **Diagnostics** tab (**Menu → Services → NTP**):



Diagnostic Information

This page displays information about the network time service. Configure this service in Control Expert.

The Network Time Service synchronizes 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):

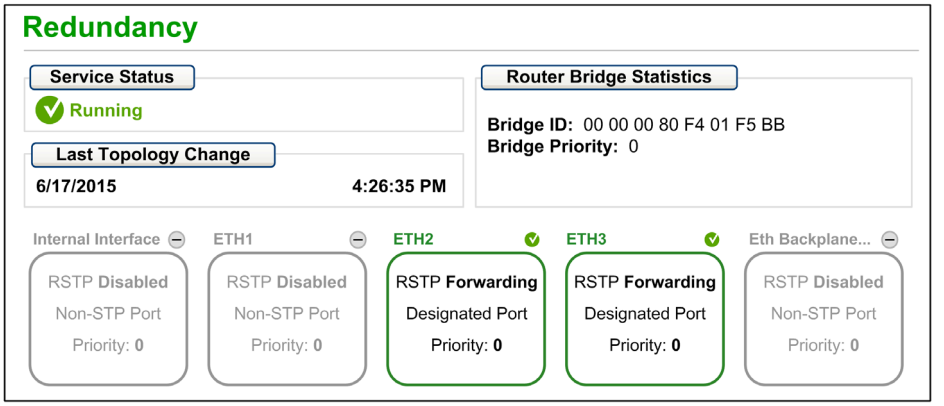
Field	Description	
Service Status	**Running**	The SNTP service is correctly configured and running.
	Disabled	The SNTP service is disabled.
	Unknown	The SNTP service status is unknown.
Server Status	green	The server is connected and running.
	red	A bad server connection is detected.
	gray	The server status is unknown.
Server Type	**Primary**	A primary server polls a master time server for the current time.
	Secondary	A secondary server requests the current time only from a primary server.

Field	Description	
DST Status	**On**	DST (daylight saving time) is configured and running.
	Off	DST is disabled.
	Unknown	The DST status is unknown.
Current Date	This is the current date in the selected time zone.	
Current Time	This is the current time in the selected time zone.	
Time Zone	This field shows the time zone in terms of plus or minus Universal Time, Coordinated (UTC).	
NTP Service Statistics	These fields show the current values for service statistics.	
	Number of Requests	This field shows the total number of requests sent to the NTP server.
	Success Rate	This field shows the percentage of successful requests out of the total number of requests.
	Number of Responses	This field shows the total number of responses received from the NTP server.
	Last Error	This field contains the error code of the last error that was detected during the transmission of an email message to the network.
	Number of Errors	This field contains the total number of SNTP messages that could not be sent to the network or that have been sent but not acknowledged by the server.

Redundancy

Introduction

The **Redundancy** page shows the redundancy status for each port on the communications module. Access the **Redundancy** web page on the **Diagnostic** tab (**Menu → Services → Redundancy**). The RSTP service is configured in Control Expert *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)*.



NOTE: This page is updated every 5 seconds.

Diagnostic Information

This table describes the diagnostics information:

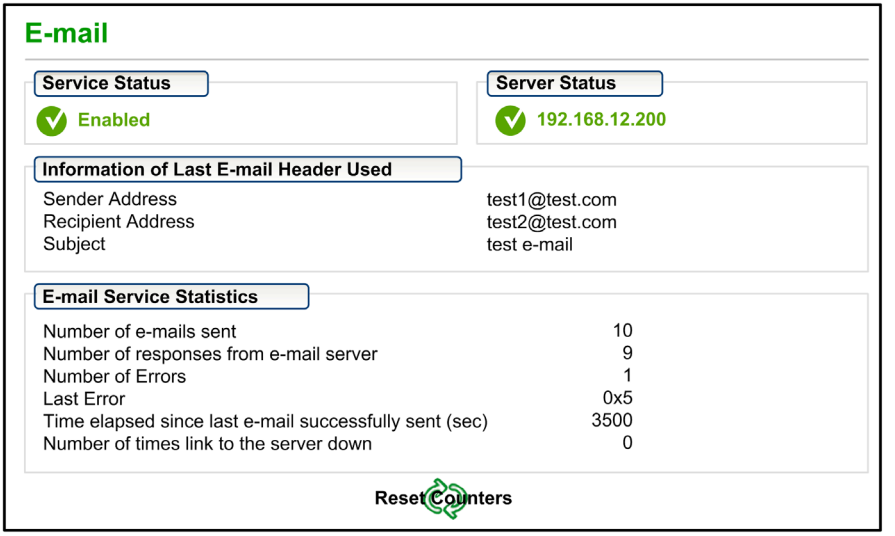
Field	Description	
Service Status	**Running**	The RSTP service on the communications module is running.
	Disabled	The RSTP service on the communications module is disabled.
	Uknown	The status of the RSTP service on the communications module is not known.
Last Topology Change	These values represent the date and time that the last topology change was received for the corresponding **Bridge ID**.	
Router Bridge Statistics	**Bridge ID**	This unique bridge identifier is the concatenation of the bridge RSTP priority and the MAC address.
	Bridge Priority	In Control Expert, configure the RSTP operating state *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* of the **Bridge ID**.

Field	Description	
Port *x* Redundancy Status	green	The designated Ethernet port is learning or forwarding information.
	yellow	The designated Ethernet port is discarding information.
	red	The designated Ethernet port detects errors.
	gray	RSTP is disabled for the designated Ethernet port.

Email Diagnostics

Open the Page

Access the **E-mail** page from the **Diagnostics** tab (**Menu → Services → E-mail**):



Click **Reset Counters** to reset the counters to 0.

Diagnostic Information

Use the **E-mail** web page to display dynamically generated data that describes the BMENOC0321 module e-mail transmissions:

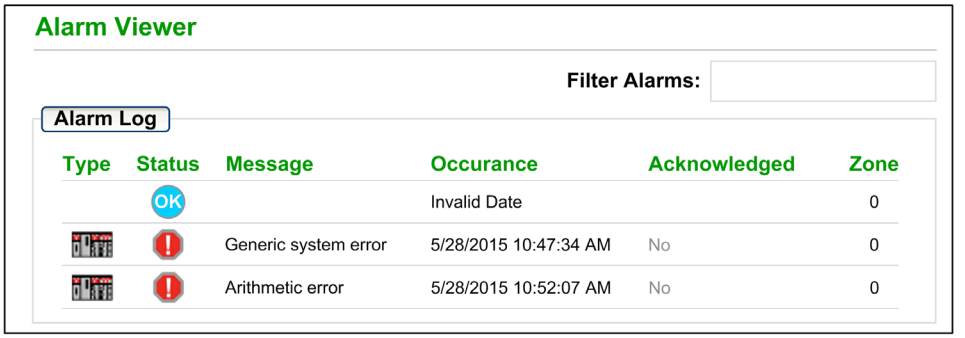
Parameter		Description
Service Status	**Enabled**	The e-mail service is correctly configured and running.
	Disabled	The e-mail service is disabled.
	Unknown	The e-mail service status is unknown.
Server Status	*green*	The e-mail server is connected and running.
	red	A bad e-mail server connection is detected.
	gray	The e-mail server status is unknown.
Information of Last E-mail 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

Parameter		Description
E-mail Service Statistics	**Number of e-mails sent**	Total number of Emails sent and successfully acknowledged by the e-mail server.
	Number of responses from e-mail server	Total number of responses received from the e-mail server
	Number of Errors	Total number of e-mails that either: ● could not be sent ● were sent but were not successfully acknowledged by the e-mail server
	Last Error	Hexadecimal code describing the reason for the last unsuccessful Email transmission. The value “0” indicates no unsuccessful transmissions.
	Time elapsed since last e-mail 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 e-mail server could not be reached. (Link checked every 30 minutes.)

Alarm Viewer

Open the Page

Access the **Alarm Viewer** page from the **Diagnostics** tab (**Menu → System → Alarm Viewer**):



NOTE: This page is updated every 5 seconds.

Diagnostic Information

The **Alarm Viewer** page reports detected application errors. You can read, filter, and sort information about alarm objects on this page. Adjust the type of information displayed by the **Alarm Viewer** in the **Filter Alarms** box.

Each alarm has a timestamp, a description, and an acknowledgement status:

- critical (red)
- acknowledged (green)
- information (blue) (These alarms do not require acknowledgement.)

This table describes the components of the page:

Column	Description	
Type	This column describes the alarm type.	
Status	**STOP**	You need to acknowledge the alarm.
	ACK	An alarm has been acknowledged.
	OK	An alarm does not require acknowledgment.
Message	This column contains the text of the alarm message.	
Occurrence	This column contains the date and time that the alarm occurred.	
Acknowledged	This column reports the acknowledged status of the alarm.	
Zone	This column contains the area or geographical zone from which the alarm comes (0: common area).	

Section 10.2
BMENOC0321 FactoryCast Configuration

Introduction

In addition to the standard web site *(see page 336)*, the BMENOC0321 module supports an expanded set of customizable web features called FactoryCast.

NOTE: Obtain the required privileges to edit the variable in the FactoryCast web pages. Use the Web Designer for FactoryCast software to configure the FactoryCast web pages. (Download the software from www.schneider-electric.com.)

What Is in This Section?

This section contains the following topics:

Navigating the Modicon M580 FactoryCast Web Pages

Introduction

The Modicon M580 FactoryCast web pages contain horizontal and vertical menus to help you navigate among the pages.

Use the horizontal menus across the top of the web pages and the vertical menus on the left of all web pages to navigate among the pages.

Use FactoryCast web pages to perform these tasks:

- Read values from and write values to Control Expert application variables.
- Manage and control access to the embedded web pages by assigning separate passwords to perform these tasks:
  - View the diagnostic web pages.
  - Use the Data Editor to write values to Control Expert application variables.

NOTE:

- To help ensure cyber security, confirm that you change the password with modules that have firmware V1.05 or later.
- You cannot reset the module to factory settings if you lose the password.

Open the Web Page

Access the FactoryCast web pages:

Step	Action
1	Open an Internet browser.
2	In the address bar, enter the IP address of the Modicon M580 communications module.
3	On the **Login** page, enter the **User Name** and **Password**. These are the default values: ● **User Name**: admin ● **Password**: factorycast
4	Click the **Login** button.
5	To access the FactoryCast configuration, select the **Monitoring** tab or the **Setup** tab. **NOTE:** Users with administrative privileges can access the **Setup** tab.

Navigation Tabs

This table describes the tabs on the FactoryCast web pages. Select any tab to see the available configuration options:

Tab	Menu Items	Description
Home	**Add Widget**	Add widgets to create web page functionality.
Monitoring	**Data Tables**	Organize variables into collections to simplify the viewing and editing.
	Graphic Viewer	Add a graphical object that represents a variable and its current value.
	Chart Viewer	Monitor the change in variables over time.
	Program Viewer	Review the structure of the program.
	Custom Pages	Access pages created in the Web Designer program.
Setup	**Administration**	Configure the appearance of web pages. Configure and monitor user access to web pages.

Home

Introduction

The Modicon M580 FactoryCast **Home** web page provides a customizable home page for FactoryCast Modicon X80 products. Use this page as a dashboard on which you can add or move widgets that apply to the data that you want to display. On this single page, you can quickly and easily monitor variables and processes.

Widgets

“Widgets” are customizable components on the **Home** page. These are the available widgets:

Widget	Description
Chart *(see page 366)*	Add a chart to monitor the change in variables over time. (Maximize the widget to access the **Chart Viewer** configuration page.)
Data Tables *(see page 361)*	Organize variables into collections to simplify the viewing and editing. (Maximize the widget to access the **Data Table** configuration page.)
Alarm Viewer *(see page 355)*	Add an alarm viewer to see information about alarm notifications that correspond to running services. (Maximize the widget to access the **Alarm Viewer** configuration page.)
Graphic *(see page 364)*	Add a graphical object that represents a variable and its current value.
Message Board	Post a message that all users can see.

The **Home** page acts as the widget dashboard. You can add up to 12 widgets on the dashboard.

Using Widgets

Add a widget to the **Home** page:

Step	Action
1	Expand (+) the **Add Widget** side menu.
2	Left-click a widget and hold the mouse key down. Notice that empty gray fields appear on the **Home** page.
3	Drag the selected widget to one of the gray areas.
4	Configure the widget according to the instructions elsewhere in this section.

NOTE: At any time, you can grab (left-click) the header of any widget to move it around the **Home** page.

Adjust the widget size:

- Press the maximize icon in the widget's header to maximize the widget.
- Press the minimize icon in the widget's header to minimize the widget.

Reconfigure the widget: Press the configuration icon (wrench) in the widget's header to access the configuration options for the widget.

Delete a widget:

Step	Action
1	View existing widgets on the **Home** page.
2	Click the gray **X** in the widget header to see the **Confirm deletion** dialog box.
3	Press **OK**.

Data Tables

Introduction

You can organize variables into collections to simplify the viewing and editing. These collections (tables) contain entries for multiple configured variables.

The data viewer animates the current value of each table variable.

Creating Data Tables

Create a new data table:

Step	Action
1	Before you create a new table, synchronize the **Data Dictionary** *(see page 376)* in these instances: ● The Ethernet communications module is installed for the first time. ● There is a change in the Control Expert application. ● The Ethernet communications module is moved to a rack with a different CPU.
2	Open the **Create Data Table** page from the **Monitoring** tab (**Menu → Data Tables → Create New Table**).
3	Enter a name for the table in the **New Table Name** field.
4	Enter a description of the table in the (optional) **Description** field.
5	If you wish, you can add variables from the **Namespace** or the **Data Dictionary** to a data table: ● Add variables from the **Namespace**: **a.** Press the **Namespace** button. (The **Namespace** is selected by default.) **b.** Wait for the **Namespace** list to load. **c.** Select any variable in the list to move it into the table. ● Add variables from the **Data Dictionary**: **a.** Press the **Data Dictionary** button. **b.** Wait for the **Data Dictionary** list to load. **c.** Select any variable in the list to move it into the table. **NOTE:** Refer to *Working with Variables* (below).
6	You can remove a **Namespace** variable or a **Data Dictionary** variable from a table by selecting the variable in the column of table contents.
7	Click **OK** to see the new table in the **Data Tables** menu.

Limitations:

- Each data table supports up to 120 variables.
- FactoryCast supports up to 30 data tables.

NOTE: Only data tables that have been created in the **Monitoring** tab can be added to the dashboard. You cannot create a new data table from the dashboard.

Working with Variables

Use these fields when you add variables to a data table:

Field	Description
Filter Variables	When you create a data table (above), you can limit the number of variables that appear in the **Symbol** or **Address** columns for the **Data Dictionary**. Only variable names and types that contain to the string in the **Filter Variables** field appear in the **Symbol** or **Address** columns.
Direct Address	In the **Direct Address** field, manually enter the address of a variable that corresponds to a memory location in the PAC (unless it is an unlocated variable).

Table Data

All system users can see and share all existing tables. To view the configuration of a data table, select the table in the **Data Tables** list (**Monitoring → Data Tables**).

Create and display a data table to see these columns:

Column	Description
Symbol	variable name
Direct Address	address of the variable in the PAC (except for unlocated variables)
Type	data type of the variable
Value	current value of the variable
Format	variable format (decimal, hex, ASCII, binary)
Status	OK or detected error

You can add, edit, delete, search, and sort variables with the data grid above:

Function	Action
sort	Left-click any column heading sort the data according to the column description.
information dialog box	Left-click any variable row to view the charts and information that apply to that variable. You can change the value of a variable if you have the proper user rights. Click the variable in the variable row to edit it. Then click the **Write** button when it appears. You can make a comment that applies to the variable in the **Comment** field.
edit	Left-click any value to edit it.
save	Press **Save** to confirm your configuration changes and comments. (To save screen space, collapse the variable information panel when it is not needed.)

Editing a Data Table

Reconfigure an existing data table:

Step	Action
1	View existing tables by expanding **Data Tables** on the **Monitoring** tab (**Menu → Data Tables**).
2	Select a table.
3	Click the gear symbol next to the table name to return to the configuration parameters.
4	Reconfigure the table.
5	Press **OK**.

Deleting a Data Table

Delete an existing data table:

Step	Action
1	View existing tables by expanding **Data Tables** on the **Monitoring** tab (**Menu → Data Tables**).
2	Select a table.
3	Click the gray **X** next to the name of the selected data table to see the **Confirm table removal** dialog box.
4	Press **Remove**.

Data Table Widget

The **Home** page acts as the widget dashboard *(see page 359)*.

The data table widget is a small version of the **Data Tables** page. The widget shows the variables of a single table and the associated values. Data in the table is updated automatically every second.

To display a **Data Table** widget, pick a table from a list that contains each table name and the number of variables that are available in that table.

Data Table widgets can display 10 variables per page for a maximum of 12 pages. Use the previous page (**<**), and next page (**>**), first page (**<<**), and last page (**>>**) buttons to scroll through the pages of the **Data Tables**.

Graphic Viewer

Introduction

Open the **Graphic Viewer** to view and monitor Web graphics that you created in Web Designer:

Step	Action
1	Select the **Monitoring** tab.
2	Expand the **Graphic Viewer** (**Menu → Graphic Viewer**).
3	Select a graphic from the **Graphic Viewer** sub-menu to view the graphic. (Graphics have user-defined names that were assigned when the graphics were created in Web Designer.)

Graphic Widgets

Use the **Graphical Viewer** to your FactoryCast Dashboard *(see page 359)* to add a graphical object that represents a variable and its current value:

Step	Action
1	Open the **Home** page.
2	Expand (+) the **Add Widget** menu.
3	Drag the **Graphical View** onto the **Home** page.
4	Select a widget type: • **Circular Gauge** • **Linear Gauge** • **Indication Light** • **Numeric Display**
5	Configure each widget type according to the specific instructions below.

Circular Gauge

Use the **Circular Gauge** to represent a numerical variable with a minimum and maximum value:

Step	Action
1	Select the **Circular Gauge** widget from the **Graphical View** menu.
2	Select a single variable in the pull-down menu.
3	Assign a minimum and maximum threshold value for the variable. These values define the valid (green) range on the gauge.
4	Click **Save**.

A **Circular Gauge** shows percentages or the variable rate at which an object is moving (for example, the speedometer in an automobile).

Linear Gauge

The **Linear Gauge** is a bar-type graphic widget that displays the value of numeric variables with minimum and maximum values:

Step	Action
1	Select the **Linear Gauge** widget from the **Graphical View** menu.
2	Select a single variable in the pull-down menu.
3	Select a **Graphic Orientation** for the gauge: • **Horizontal:** A horizontal gauge shows the change in minimum and maximum values from left to right. (Horizontal gauges are often used in Windows programs to show elapsed time.) • **Vertical:** A vertical gauge shows the change in minimum and maximum values from bottom to top. (For example, most thermometers are vertical gauges.)
4	Assign a minimum and maximum threshold value for the variable. These values define the valid (green) range on the gauge.
5	Click **Save**.

Indication Light

The **Indication Light** is a simple graphic representation of a boolean value that is off or on:

Step	Action
1	Select the **Indication Light** widget from the **Graphical View** menu.
2	Select a single variable in the pull-down menu.
3	Select an LED color in the **Color if True** pull-down menu to assign that color to the on (1) state.
4	Select an LED color in the **Color if False** pull-down menu to assign that color to the off (0) state.
5	Click **Save**.

Numeric Display

Use the **Numeric Display** widget to customize the categorical graphic and unit of measurement for a numeric variable or address. The **Numerical Display** shows the current value of the variable and the minimum and maximum values the variable reaches while the widget is on the **Dashboard**:

Step	Action
1	Select the **Numeric Display** widget from the **Graphical View** menu.
2	Select a single variable in the pull-down menu.
3	Scroll to an image in the **Image** pull-down menu that corresponds to the selected variable.
4	Indicate the unit in the **Unit of Measure** field.
5	Click **Save**.

Chart Viewer

Introduction

To illustrate how variables change over time, the FactoryCast **Chart Viewer** displays values on a chart at the speed of the plot frequency. Each chart can report values for five variables at once.

Creating Charts

Create a new chart:

Step	Action
1	Before you create a new chart, synchronize the Data Dictionary *(see page 376)* in these instances: ● The Ethernet communications module is installed for the first time. ● There is a change in the Control Expert application. ● The Ethernet communications module is moved to a rack with a different CPU.
2	Access the **Chart Viewer** page from the **Monitoring** tab (**Menu → Chart Viewer → Create Chart**).
3	In the **Chart Name** field, enter a name for the chart.
4	In the **Plot Frequency** field, scroll to the interval for data plotting that applies to the **Plot frequency unit**.
5	In the **Plot frequency unit** field, scroll to the unit for data plotting (**Milliseconds**, **Seconds**, **Minutes**, **Hours**).
6	In the **Plot Points** field, scroll to the number of points on the chart.
7	Use the **Auto-Scale** check box to scale the chart: ● *checked*: Scale the chart according to the point being plotted. ● *unchecked*: Do not scale the chart according to the point being plotted and enter these fixed values: ❍ *Y Min*: Set the lower limit for the y-axis of the selected object. ❍ *Y Max*: Set the upper limit for the y-axis of the selected object.
8	If you wish, you can add variables from the **Namespace** or the **Data Dictionary** to a chart: ● Add variables from the **Namespace**: **a.** Press the **Namespace** button. (The **Namespace** is selected by default.) **b.** Wait for the **Namespace** list to load. **c.** Select any variable in the list to move it into the chart. ● Add variables from the **Data Dictionary**: **a.** Press the **Data Dictionary** button. **b.** Wait for the **Data Dictionary** list to load. **c.** Select any variable in the list to move it into the chart. **NOTE:** Refer to *Working with Variables* below.

Step	Action
9	You can remove a **Namespace** variable or a **Data Dictionary** variable from a chart by selecting the variable in the column of chart contents.
10	Click **Create Chart**.

The new chart appears in the **Chart Viewer** list on the **Monitoring** tab.

NOTE: Only charts that have been created in the **Monitoring** tab can be added to the dashboard. You cannot create a new chart from the dashboard.

Working with Variables

Use these fields when you add variables to a chart:

Field	Description
Filter Variables	When you create a chart (above), you can limit the number of variables that appear in the **Symbol** or **Address** columns for the Data Dictionary. Only variable names and types that contain to the string in the **Filter Variables** field appear in the **Symbol** or **Address** columns.
Direct Address	In the **Direct Address** field, manually enter the address of a variable that corresponds to a memory location in the PAC (unless it is an unlocated variable).

Presentation Modes

There are several presentation modes in the **Chart Viewer**. These modes present data in a manner that is appropriate to the information associated with the variable(s):

Mode	Description
Bar	Use this mode to see the value of a variable at one moment in time. In this mode, it is easy to compare the relative values of multiple variables.
Line	Use this mode to view the values of variables that change over time. In this mode, it is easy to compare the relative values of multiple variables.
Both	Use this mode to view the bar chart and the line chart on the same page.

Legend

Each chart has a legend that contains the symbol, address, and value associated with each variable. The values in the legend are animated at the speed of the plot frequency.

Editing a Chart

Reconfigure an existing chart:

Step	Action
1	View existing charts by expanding **Chart Viewer** on the **Monitoring** tab (**Menu → Chart Viewer**).
2	Select a chart.
3	Click the gear symbol next to the chart name to return to the configuration parameters.
4	Reconfigure the chart.
5	Press **OK**.

Deleting a Chart

Delete an existing chart:

Step	Action
1	View existing charts by expanding **Chart Viewer** on the **Monitoring** tab (**Menu → Chart Viewer**).
2	Select a chart.
3	Click the gray **X** next to the name of the selected chart to see the **Confirm chart removal** dialog box.
4	Press **Remove**.

Program Viewer

Introduction

Open the **Program Viewer** to view and monitor the Control Expert programs that are in run mode:

Step	Action
1	Select the **Monitoring** tab.
2	Expand the **Program Viewer** (**Menu → Program Viewer**).
3	Click **Open Program Viewer**.

PLC Programs

Control Expert supports these PAC (PLC) programs, which you can view on the **Program Viewer** page:

- Ladder (LD)
- Instruction List (IL)
- Function Block Diagram (FBD)
- Structured Text (ST)
- Sequential Function Chart (SFC)
- Function block diagram LL984

Click the PAC/PLC program section in the navigation tree view to display the selected program section.

Variable Animation

Boolean variables are displayed in different colors:

Color	Indication
green	The value is true.
red	The value is false.
yellow	The value is of a type that is neither true nor false. (Use the **Tool Tip** below to find information about the variable's name, type, address, and comment.)

Values in the **Program Viewer** page are refreshed more than once per second.

Links Animation

The links to boolean variables are displayed in different colors depending on the value of the variable to which they are connected:

Color	Indication
green	The value is true.
red	The value is false.
black	The value of all other links.

Tool Tip

The **Tool Tip** help bubble appears when the cursor hovers over a variable. The bubble displays this information

- The value of the variable if only its name is visible in the **Program Viewer**.
- The type, name, address, and comment if only its value is visible in the viewer.

Click on the variable to display the bubble permanently. Right click on the variable to make the bubble disappear.

The **Program Viewer** gets the program directly from the PAC/PLC. It can detect a program change in order to automatically synchronize to the PAC without any user intervention or configuration. All available sections are displayed.

The **Program Viewer** displays status messages in the Console pane at the bottom of the page. Here are some examples:

- A generic error is detected.
- The PAC/PLC is reserved by someone else.
- The PAC/PLC must be reserved.
- The response could not be built.
- The request has invalid parameters.
- A bad sequence exists.
- The response is too big for the available response buffer.
- The module is not configured.
- The action is not permitted on the object.
- There is an application/PAC compatibility error (RELOAD)
- A general error is detected.

Values in the **Program Viewer** sections are refreshed more than once per second.

Control Expert Project Settings

In the **Property value** column, check the **Program Viewer Information** checkbox and the **Data Dictionary** checkbox in the Control Expert project settings to make the **Program Viewer** available with automatic synchronization of the PAC/PLC program in the **Program Viewer** web page.

URL Parameters

You can configure the parameters in the URL to show or hide the navigation tree (at the left of the PLC program viewer), to show or hide the console (at the bottom of the PLC program viewer), and to focus on a specific section or object in the PLC program.

Log in to the FactoryCast web site and use these URLs:

- Hide the FactoryCast banner:
  `http://<IP>/#monitoring/plcpv?standalone=1`
- View a single section:
  `http://<IP>/#monitoring/plcpv?showTreeview=0&showConsole=0&section=<SECTION_NAME>&standalone=1`

Administration

Introduction

Use the **Administration** page to perform these tasks:

- Configure the look and feel of the web pages.
- Monitor and control access to the website.

Open the **Administration** page:

Step	Action
1	Click the **Setup** tab.
2	Expand (+) **Administration** (**Menu → Administration**).

Menu

You can select these items from the **Administration** menu:

Selection	Description
Themes	Manage the color theme for the web pages.
User Access	Manage users and user access rights.
Access Management	Manage password and security settings.
Namespace Manager	Add variables to the namespace.
	Remove variables from the namespace.
Data Dictionary Sync	Use this page to synchronize the Data Dictionary to make the updated Data Dictionary available to the **Namespace Manager**. Use the updated Data Dictionary to create data tables *(see page 361)* and charts *(see page 366)*.
Logo Manager	Assign graphic elements to be used by themes.

These items are described in detail below.

Themes

A FactoryCast a theme is a named set of GUI options that create the look and feel that is applied to the system.

Configure the color scheme for the web pages:

Step	Action
1	Open the **Theme Management** configuration page on the **Setup** tab (**Menu → Administration → Themes**).
2	Click any item in the **Theme Name** column to change the color scheme.

The **Theme Management** page contains some predefined themes. You can modify or delete some of the predefined themes. The default theme (**Schneider**) represents a standard FactoryCast view and cannot be modified or deleted. (You cannot change the theme for custom pages.)

Create a new theme:

Step	Action
1	Click the plus sign (**+**) next to **Theme Management** to view the **Theme** configuration.
2	Enter a unique name in the **Theme Name** field.
3	Enter a description in the **Description** field.
4	In the **Logo** field, scroll to a logo from the **Logo Manager**.
5	Enter a site title in the **Site Title** field.
6	In the **Import Theme Colors** field, scroll to a **Theme Name** from the **Theme Management** page.
7	In the **Header** area, set the color of the **Title**, **Header Background**, and **Header Text** fields. Set the **Header Background** as an example: **a.** Click the color field next to **Header Background** to see the color-selection window. **b.** Move the dot in the outer circle to select a color range. **c.** Click inside the square to select a particular color. (Note that the background color in the header changes to the selected color.) **d.** Press **OK**.
8	Repeat the above step to change the colors in the **Top Menu**, **Side Menu**, and **Page Content**.
9	Press **Save** to save the new theme to the **Theme Management** list.

NOTE: To remove a theme from the **Theme Name** column, click the minus sign (**-**) in the **Actions** column.

User Access

Open the **User Access** configuration page on the **Setup** tab (**Menu → Administration → User Access**).

The **User Access** page contains information in these columns:

Column	Description	
Locked	checked	You can change or configure web page access for the corresponding user.
	unchecked	You cannot change or configure web page access for the corresponding user.
Username	This column displays the name of the user to whom the row corresponds.	
Password	Click the arrows in this column to reset the password (when permitted). **NOTE:** • To help ensure cyber security, confirm that you change the password with modules that have firmware V1.05 or later. • You cannot reset the module to factory settings if you lose the password.	
Last Login	This column shows the last time that the corresponding user logged in.	
Admin	checked	This user has administrative privileges.
	unchecked	This user is not an administrator.
Write Permission	checked	This user can write to the web pages.
	unchecked	This user has read-only access to the web pages.
# Failed Logins	This value represents the number of times the corresponding user is unable to log in.	
Delete	Click the **X** to delete this user.	

Access Management

Open the **Access Management** configuration page on the **Setup** tab (**Menu → Administration → Access Management**).

Configure the settings for web-page access:

Field	Parameter	Description
Access Management	**Security On**	Click to control access to the web pages. (When you activate the security, you return to the login page.)
	Security Off	Click to hide all **Password Policy** fields and allow unrestricted access to the web pages.
Password Policy	**Enforce Password Policy**	**On:** Click to view and configure password requirements.
		Off: Click to hide password requirements and allow any combination of characters for passwords.
	Password History	**Off:** You can reuse old passwords.
		Last 3: You cannot use any of your last three passwords.
		Last 5: You cannot use any of your last five passwords.
	Require Special Character	**On:** Click to require at least one special character (#, $, &, etc.) in the password.
		Off: Click to allow passwords without special characters.
	Require Numeric Character	**On:** Click to require at least one numeric character (1, 2, 3, etc.) in the password.
		Off: Click to allow passwords without numeric characters.
	Require Alphabetic Character	**On:** Click to require at least one alphabetic character (a, b, c, etc.) in the password.
		Off: Click to allow passwords without alphabetic characters.
	Minimum Password Length	Enter a numeric value to indicate the minimum number of characters in a password.
Buttons	**Save**	Click to save the new password settings.
	Reset	Click to revert to the last saved password settings.

Namespace Manager

Open the **Namespace Manager** configuration page on the **Setup** tab (**Menu → Administration → Namespace Manager**).

Use the **Namespace Manager** to move variables from the **Data Dictionary** on the CPU to a local database on the communications module for faster access.

NOTE: The Namespace can contain a maximum of 1000 variables.

Data Dictionary Sync

Use this page to synchronize the Data Dictionary. The synchronization makes the Data Dictionary available for use by the **Data Table** and **Chart** and **Namespace Manager** (above) pages.

Synchronize the Data Dictionary in these instances:
- The Ethernet communications module is installed for the first time.
- There is a change in the Control Expert application.
- The Ethernet communications module is moved to a rack with a different CPU.

NOTE:
- The synchronization of the data dictionary consumes the first 3600 (± 40) variables from the data dictionary on the CPU and copies them to the database on this device.
- The synchronization process can take several minutes.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Do not interrupt a **Data Dictionary Sync** that is in progress.

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

Synchronize the Data Dictionary:

Step	Action
1	Enable the Data Dictionary in Control Expert. **NOTE:** Refer to the General Project Settings *(see EcoStruxure™ Control Expert, Operating Modes)* in the ***Control Expert Operating Modes*** guide.
2	Select the **Setup** tab.
3	Expand the **Administration** menu.
4	Select **Data Dictionary Sync**.
5	Press **Start Synchronization**.
6	Wait for the synchronization to stop (**Synchronization completed**).

The synchronization function supports these variable types:
- BOOL
- BYTE
- DATE
- DINT
- DT
- DWORD
- EBOOL
- INT
- REAL

- STRING
- TIME
- UDINT
- UINT
- TOD
- WORD

Logo Manager

Import small graphics as logos that you can apply the themes *(see page 373)*.

Open the **Logo Manager** configuration page on the **Setup** tab (**Menu → Administration → Logo Manager**).

Add a new graphic to the **Logo Manager**:

Step	Action
1	Click the plus sign (**+**) next to **Logo Manager**.
2	Drive to a graphic that you want to use as a logo. **NOTE:** The maximum file size is 5KB.
3	Press the **Upload** button to see the new logo in the **Thumbnail** column.

NOTE: To remove a logo from the **Thumbnail** column, click the **X** in the **Delete** column.

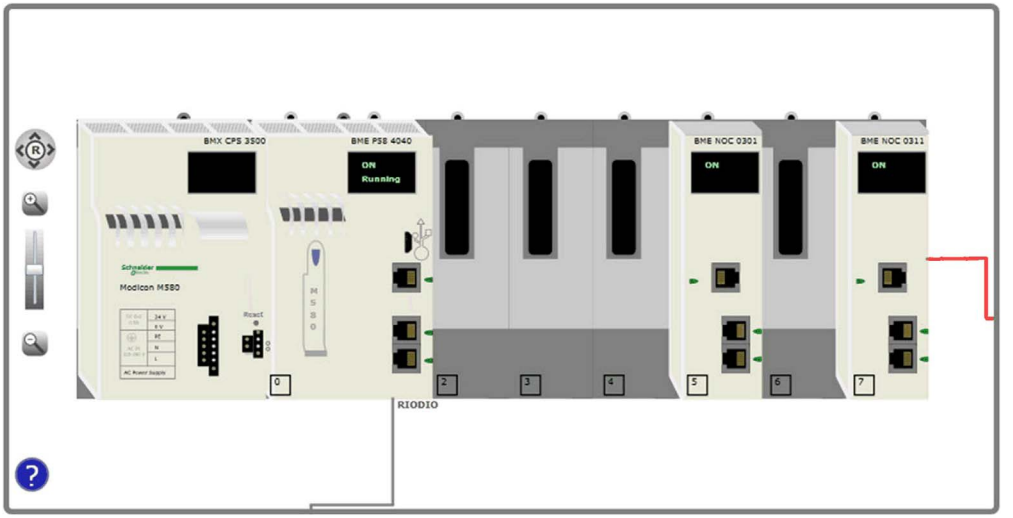
Rack Viewer

Open the Page

Access the **Rack Viewer** page from the **Diagnostics** tab (**Menu → System → Rack Viewer**).

Example

This **Rack Viewer** page for an M580 FactoryCast module shows a local rack that contains a power supply, a CPU, a FactoryCast communications module in slot 5, and a FactoryCast communications module in slot 7:



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:

EtherNet/IP Implicit or Explicit Messaging Detected Error Codes

Introduction

If a DATA_EXCH function block does not execute an EtherNet/IP explicit message, Control Expert returns a hexadecimal detected error code. The code can describe an EtherNet/IP detected error.

EtherNet/IP Detected Error Codes

EtherNet/IP hexadecimal detected error codes include:

Detected Error Code	Description
16#800D	Timeout on the explicit message request
16#8012	Bad device
16#8015	Either: ● Nor resources to handle the message, or ● Internal detected error: 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 detected errors are Forward_Open response detected errors 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
16#8119	Non-Listen Only connection not opened

Detected Error Code	Description
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 detected error 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	Detected error 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	Detected error in data assignment
16#81B0	Optional object state detected error

Detected Error Code	Description
16#81C0	Optional device state detected error
Note: All 16#82xx detected errors are register session response detected errors.	
16#8200	Target device does not have sufficient resources
16#8208	Target device does not recognize message encapsulation header
16#820F	Reserved or unknown detected error from target

Explicit Messaging: Communication and Operation Reports

Overview

Communication and operation reports are part of the management parameters.

NOTE: It is recommended that communication function reports be tested at the end of their execution and before the next activation. On cold start-up, confirm that all communication function management parameters are checked and reset to 0.
It may be helpful to use the%S21 *(see EcoStruxure™ Control Expert, System Bits and Words, Reference Manual)* to examine the first cycle after a cold or warm start.

Communication Report

This report is common to every explicit messaging function. It is significant when the value of the activity bit switches from 1 to 0. The reports with a value between 16#01 and 16#FE concern errors detected by the processor that executed the function.

The different values of this report are indicated in the following table:

Value	Communication report (least significant byte)
16#00	Correct exchange
16#01	Exchange stop on timeout
16#02	Exchange stop on user request (CANCEL)
16#03	Incorrect address format
16#04	Incorrect destination address
16#05	Incorrect management parameter format
16#06	Incorrect specific parameters
16#07	Error detected in sending to the destination
16#08	Reserved
16#09	Insufficient receive buffer size
16#0A	Insufficient send buffer size
16#0B	No system resources: the number of simultaneous communication EFs exceeds the maximum that can be managed by the processor
16#0C	Incorrect exchange number
16#0D	No telegram received
16#0E	Incorrect length
16#0F	Telegram service not configured
16#10	Network module missing
16#11	Request missing
16#12	Application server already active

Value	Communication report (least significant byte)
16#13	UNI-TE V2 transaction number incorrect
16#FF	Message refused

NOTE: The function can detect a parameter error before activating the exchange. In this case the activity bit remains at 0, and the report is initialized with values corresponding to the detected error.

Operation Report

This report byte is specific to each function, and specifies the result of the operation on the remote application:

Value	Operation report (most significant byte)
16#05	Length mismatch (CIP)
16#07	Bad IP address
16#08	Application error
16#09	Network is down
16#0A	Connection reset by peer
16#0C	Communication function not active
16#0D	• Modbus TCP: transaction timed out • EtherNet/IP: request timeout
16#0F	No route to remote host
16#13	Connection refused
16#15	• Modbus TCP: no resources • EtherNet/IP: no resources to handle the message; or an internal detected error; or no buffer available; or no link available; or cannot send message
16#16	Remote address not allowed
16#18	• Modbus TCP: concurrent connections or transactions limit reached • EtherNet/IP: TCP connection or encapsulation session in progress
16#19	Connection timed out
16#22	Modbus TCP: invalid response
16#23	Modbus TCP: invalid device ID response
16#30	• Modbus TCP: remote host is down • EtherNet/IP: connection open timed out
16#80...16#87: Forward_Open response detected errors:	
16#80	Internal detected error
16#81	Configuration detected error: the length of the explicit message, or the RPI rate, needs to be adjusted
16#82	Device detected error: target device does not support this service

Value	Operation report (most significant byte)
16#83	Device resource detected error: no resource is available to open the connection
16#84	System resource event: unable to reach the device
16#85	Data sheet detected error: incorrect EDS file
16#86	Invalid connection size
16#90...16#9F: Register session response detected errors:	
16#90	Target device does not have sufficient resources
16#98	Target device does not recognize message encapsulation header
16#9F	Unknown detected error from target

Electronic Mail Notification Service Detected Error Response Codes

SMTP Codes

The following codes are available only on the Control Expert DTM and web page diagnostic screens for the electronic mail notification service:

Code (hexadecimal)	Description
16#5100	Internal error detected
16#5101	SMTP component not operational
16#5102	Mail header not configured
16#5103	Invalid mail header value detected (1, 2, or 3)
16#5104	Cannot connect to SMTP server
16#5105	Error detected during transmitting content of email body to SMTP server
16#5106	Closing SMTP connection with the server returned a detected error message
16#5107	SMTP HELO request unsuccessful
16#5108	SMTP MAIL request unsuccessful — SMTP server may require authentication
16#5109	SMTP RCPT request unsuccessful
16#510A	No recipient accepted by the SMTP server
16#510B	SMTP DATA request unsuccessful
16#510C	Send email request contains an invalid length
16#510D	Authentication unsuccessful
16#510E	A reset component request was received while the connection was open

Glossary

!

%I

According to the CEI standard, %I indicates a language object of type discrete IN.

%IW

According to the CEI standard, %IW indicates a language object of type analog IN.

%M

According to the CEI standard, %M indicates a language object of type memory bit.

%MW

According to the CEI standard, %MW indicates a language object of type memory word.

%Q

According to the CEI standard, %Q indicates a language object of type discrete OUT.

%QW

According to the CEI standard, %QW indicates a language object of type analog OUT.

%SW

According to the CEI standard, %SW indicates a language object of type system word.

802.1Q

The IEEE protocol designator for Virtual Local Area Network (VLAN). This standard provides VLAN identification and quality of service (QoS) levels.

A

adapter

An adapter is 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

In Control Expert, advanced mode is a selection that displays expert-level configuration properties that help define Ethernet connections. Because these properties should be edited only by people with a good understanding of EtherNet/IP communication protocols, they can be hidden or displayed, depending upon the qualifications of the specific user.

applicative time stamping

Use the applicative time stamping solution to access time stamp event buffers with a SCADA system that does not support the OPC DA interface. In this case, function blocks in the Control Expert PLC application read events in the buffer and formats them to be sent to the SCADA system.

architecture

Architecture describes a framework for the specification of a network that is constructed of these components:

- physical components and their functional organization and configuration
- operational principles and procedures
- data formats used in its operation

ARRAY

An `ARRAY` is a table containing elements of a single type. This is the syntax: `ARRAY [<limits>] OF <Type>`

Example: `ARRAY [1..2] OF BOOL` is a one-dimensional table with two elements of type `BOOL`.

`ARRAY [1..10, 1..20] OF INT` is a two-dimensional table with 10x20 elements of type `INT`.

ART

(*application response time*) The time a CPU application takes to react to a given input. ART is measured from the time a physical signal in the CPU turns on and triggers a write command until the remote output turns on to signify that the data has been received.

AUX

An (AUX) task is an optional, periodic processor task that is run through its programming software. The AUX task is used to execute a part of the application requiring a low priority. This task is executed only if the MAST and FAST tasks have nothing to execute. The AUX task has two sections:

- IN: Inputs are copied to the IN section before execution of the AUX task.
- OUT: Outputs are copied to the OUT section after execution of the AUX task.

B

BCD

(*binary-coded decimal*) Binary encoding of decimal numbers.

BOOL

(*boolean type*) This is the basic data type in computing. A `BOOL` variable can have either of these values: 0 (`FALSE`) or 1 (`TRUE`).

A bit extracted from a word is of type `BOOL`, for example: `%MW10.4`.

BOOTP

(*bootstrap protocol*) A UDP network protocol that can be used by a network client to automatically obtain an IP address from a server. The client identifies itself to the server using its MAC address. The server, which maintains a pre-configured table of client device MAC addresses and associated IP addresses, sends the client its defined IP address. The BOOTP service utilizes UDP ports 67 and 68.

broadcast

A message sent to all devices in a broadcast domain.

C

CCOTF

(*change configuration on the fly*) A feature of Control Expert that allows a module hardware change in the system configuration while the system is operating. This change does not impact active operations.

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.

class 1 connection

A CIP transport class 1 connection used for I/O data transmission via implicit messaging between EtherNet/IP devices.

class 3 connection

A CIP transport class 3 connection used for explicit messaging between EtherNet/IP devices.

connected messaging

In EtherNet/IP, connected messaging uses a CIP connection for communication. A connected message is a logical relationship between two or more application objects on different nodes. The connection establishes a virtual circuit in advance for a particular purpose, such as frequent explicit messages or real-time I/O data transfers.

connection

A virtual circuit between two or more network devices, created prior to the transmission of data. After a connection is established, a series of data is transmitted over the same communication path, without the need to include routing information, including source and destination address, with each piece of data.

connection originator

The EtherNet/IP network node that initiates a connection request for I/O data transfer or explicit messaging.

connectionless

Describes communication between two network devices, whereby data is sent without prior arrangement between the two devices. Each piece of transmitted data also includes routing information, including source and destination address.

control network

An Ethernet-based network containing PACs, SCADA systems, an NTP server, PCs, AMS, switches, etc. Two kinds of topologies are supported:

- flat: All modules and devices in this network belong to same subnet.
- 2 levels: The network is split into an operation network and an inter-controller network. These two networks can be physically independent, but are generally linked by a routing device.

CPU

(*central processing unit*) The CPU, also known as the processor or controller, is the brain of an industrial manufacturing process. It automates a process as opposed to relay control systems. CPUs are computers suited to survive the harsh conditions of an industrial environment.

D

DDT

(*derived data type*) A derived data type is a set of elements with the same type (`ARRAY`) or with different types (structure).

determinism

For a defined application and architecture, you can predict that the delay between an event (change of value of an input) and the corresponding change of a controller output is a finite time *t*, smaller than the deadline required by your process.

Device DDT (DDDT)

A Device DDT is a DDT predefined by the manufacturer and not modifiable by user. It contains the I/O language elements of an I/O module.

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.

device network

An Ethernet-based network within an RIO network that contains both RIO and distributed equipment. Devices connected on this network follow specific rules to allow RIO determinism.

DFB

(*derived function block*) DFB types are function blocks that can be defined by the user in ST, IL, LD or FBD language.

Using these DFB types in an application makes it possible to:

- simplify the design and entry of the program
- make the program easier to read
- make it easier to debug
- reduce the amount of code generated

DHCP

(*dynamic host configuration protocol*) An extension of the BOOTP communications protocol that provides for the automatic assignment of IP addressing settings, including 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.

DIO

(*distributed I/O*) Also known as distributed equipment. DRSs use DIO ports to connect distributed equipment.

DIO cloud

A group of distributed equipment that is not required to support RSTP. DIO 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 the CPU or Ethernet communications modules in the *local rack*. DIO clouds **cannot** be connected to *sub-rings*.

DIO network

A network containing distributed equipment, in which I/O scanning is performed by a CPU with DIO scanner service on the local rack. DIO network traffic is delivered after RIO traffic, which takes priority in an RIO network.

distributed equipment

Any Ethernet device (Schneider Electric device, PC, servers, or third-party devices) that supports exchange with a CPU or other Ethernet I/O scanner service.

DNS

(*domain name server/service*) A service that translates an alpha-numeric domain name into an IP address, the unique identifier of a device on the network.

domain name

An alpha-numeric string that identifies a device on the internet, and which appears as the primary component of a web site's uniform resource locator (URL). For example, the domain name *schneider-electric.com* is the primary component of the URL *www.schneider-electric.com*.

Each domain name is assigned as part of the domain name system, and is associated with an IP address.

Also called a host name.

DRS

(*dual-ring switch*) A ConneXium extended managed switch that has been configured to operate on an Ethernet network. Predefined configuration files are provided by Schneider Electric to downloaded to a DRS to support the special features of the main ring / sub-ring architecture.

DSCP

(*differentiated service code points*) This 6-bit field is in the header of an IP packet to classify and prioritize traffic.

DST

(*daylight saving time*) DST is also called *summer time* and is a practice consisting of adjusting forward the clock near the start of spring and adjusting it backward near the start of autumn.

DT

(*date and time*) The `DT` type, encoded in BCD in a 64-bit format, contains this information:

- the year encoded in a 16-bit field
- the month encoded in an 8-bit field
- the day encoded in an 8-bit field
- the time encoded in an 8-bit field
- the minutes encoded in an 8-bit field
- the seconds encoded in an 8-bit field

NOTE: The eight least significant bits are not used.

The `DT` type is entered in this format:

DT#<Year>**-**<Month>**-**<Day>**-**<Hour>**:**<Minutes>**:**<Seconds>

This table shows the upper/lower limits of each field:

Field	Limits	Comment
Year	[1990,2099]	Year
Month	[01,12]	The leading 0 is displayed; it can be omitted during data entry.
Day	[01,31]	For months 01/03/05/07/08/10/12
	[01,30]	For months 04/06/09/11
	[01,29]	For month 02 (leap years)
	[01,28]	For month 02 (non-leap years)
Hour	[00,23]	The leading 0 is displayed; it can be omitted during data entry.
Minute	[00,59]	The leading 0 is displayed; it can be omitted during data entry.
Second	[00,59]	The leading 0 is displayed; it can be omitted during data entry.

DTM

(*device type manager*) A DTM is a device driver running on the host PC. It provides a unified structure for accessing device parameters, configuring and operating the devices, and troubleshooting devices. 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*) EDS are simple text files that describe the configuration capabilities of a device. EDS files are generated and maintained by the manufacturer of the device.

EF

(*elementary function*) This is a block used in a program which performs a predefined logical function.

A function does not have any information on the internal state. Several calls to the same function using the same input parameters will return the same output values. You will find information on the graphic form of the function call in the [*functional block (instance)*]. Unlike a call to a function block, function calls include only an output which is not named and whose name is identical to that of the function. In FBD, each call is indicated by a unique [number] via the graphic block. This number is managed automatically and cannot be modified.

Position and configure these functions in your program to execute your application.

You can also develop other functions using the SDKC development kit.

EFB

(*elementary function block*) This is a block used in a program which performs a predefined logical function.

EFBs have states and internal parameters. Even if the inputs are identical, the output values may differ. For example, a counter has an output indicating that the preselection value has been reached. This output is set to 1 when the current value is equal to the preselection value.

EIO network

(Ethernet I/O) An Ethernet-based network that contains three types of devices:

- local rack
- X80 remote drop (using a BM•CRA312•0 adapter module), or a BMENOS0300 network option switch module
- ConneXium extended dual-ring switch (DRS)

NOTE: Distributed equipment may also participate in an Ethernet I/O network via connection to DRSs or the service port of X80 remote modules.

EN

EN stands for **EN**able; it is an optional block input. When the EN input is enabled, an ENO output is set automatically.

If EN = 0, the block is not enabled; its internal program is not executed, and ENO is set to 0.

If EN = 1, the block's internal program is run and ENO is set to 1. If a runtime error is detected, ENO is set to 0.

If the EN input is not connected, it is set automatically to 1.

ENO

ENO stands for **E**rror **NO**tification; this is the output associated with the optional input EN.

If ENO is set to 0 (either because EN = 0 or if a runtime error is detected):

- The status of the function block outputs remains the same as it was during the previous scanning cycle that executed correctly.
- The output(s) of the function, as well as the procedures, are set to 0.

Ethernet

A 10 Mb/s, 100 Mb/s, or 1 Gb/s, CSMA/CD, frame-based LAN that can run over copper twisted pair or fiber optic cable, or wireless. The IEEE standard 802.3 defines the rules for configuring a wired Ethernet network; the IEEE standard 802.11 defines the rules for configuring a wireless Ethernet network. Common forms include 10BASE-T, 100BASE-TX, and 1000BASE-T, which can utilize category 5e copper twisted pair cables and RJ45 modular connectors.

Ethernet DIO scanner service

This embedded DIO scanner service of M580 CPUs manages distributed equipment on an M580 device network.

Ethernet I/O scanner service

This embedded Ethernet I/O scanner service of M580 CPUs manages distributed equipment **and** RIO drops on an M580 device network.

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.

explicit messaging client

(*explicit messaging client class*) The device class defined by the ODVA for EtherNet/IP nodes that only support explicit messaging as a client. HMI and SCADA systems are common examples of this device class.

F

FAST

A FAST task is an optional, periodic processor task that identifies high priority, multiple scan requests, which is run through its programming software. A FAST task can schedule selected I/O modules to have their logic solved more than once per scan. The FAST task has two sections:

- IN: Inputs are copied to the IN section before execution of the FAST task.
- OUT: Outputs are copied to the OUT section after execution of the FAST task.

FBD

(*function block diagram*) An IEC 61131-3 graphical programming language that works like a flowchart. By adding simple logical blocks (`AND`, `OR`, etc.), each function or function block in the program is represented in this graphical format. For each block, the inputs are on the left and the outputs on the right. Block outputs can be linked to inputs of other blocks to create complex expressions.

FDR

(*fast device replacement*) A service that uses configuration software to replace an inoperable product.

FDT

(*field device tool*) The technology that harmonizes communication between field devices and the system host.

FTP

(*file transfer protocol*) A protocol that copies a file from one host to another over a TCP/IP-based network, such as the internet. FTP uses a client-server architecture as well as separate control and data connections between the client and server.

full duplex

The ability of two networked devices to independently and simultaneously communicate with each other in both directions.

function block diagram

See FBD.

G

gateway

A gateway device interconnects two different networks, sometimes through different network protocols. When it connects networks based on different protocols, a gateway converts a datagram from one protocol stack into the other. When used to connect two IP-based networks, a gateway (also called a router) has two separate IP addresses, one on each network.

GPS

(*global positioning system*) The GPS standard consists of a space-based positioning, navigation, and timing signals delivered worldwide for civil and military use. Standard positioning service performance depends on satellite broadcast signal parameters, GPS constellation design, the number of satellites in sight, and various environmental parameters.

H

harsh environment

Resistance to hydrocarbons, industrial oils, detergents and solder chips. Relative humidity up to 100%, saline atmosphere, significant temperature variations, operating temperature between -10°C and + 70°C, or in mobile installations. For hardened (H) devices, the relative humidity is up to 95% and the operating temperature is between -25°C and + 70°C.

HART

(*highway addressable remote transducer*) A bi-directional communication protocol for sending and receiving digital information across analog wires between a control or monitoring system and smart devices.

HART is the global standard for providing data access between host systems and intelligent field instruments. A host can be any software application from a technician's hand-held device or laptop to a plant's process control, asset management, or other system using any control platform.

high-capacity daisy chain loop

Often referred to as HCDL, a high-capacity daisy chain loop uses dual-ring switches (DRSs) to connect device sub-rings (containing RIO drops or distributed equipment) and/or DIO clouds to the Ethernet RIO network.

HMI

(*human machine interface*) System that allows interaction between a human and a machine.

Hot Standby

A Hot Standby system uses a primary PAC (PLC) and a standby PAC. The two PAC racks have identical hardware and software configurations. The standby PAC monitors the current system status of the primary PAC. If the primary PAC becomes inoperable, high-availability control is maintained when the standby PAC takes control of the system.

HTTP

(*hypertext transfer protocol*) A networking protocol for distributed and collaborative information systems. HTTP is the basis of data communication for the web.

I

I/O scanner

An Ethernet service that continuously polls I/O modules to collect data, status, event, and diagnostics information. This process monitors inputs and controls outputs. This service supports both RIO and DIO logic scanning.

IEC 61131-3

International standard: programmable logic controllers

Part 3: programming languages

IGMP

(internet group management protocol) This internet standard for multicasting allows a host to subscribe to a particular multicast group.

IL

(*instruction list*) An IEC 61131-3 programming language that contains a series of basic instructions. It is very close to assembly language used to program processors. Each instruction is made up of an instruction code and an operand.

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, plus a connection identifier.

INT

(*INTeger*) (encoded in 16 bits) The upper/lower limits are as follows: -(2 to the power of 15) to (2 to the power of 15) - 1.

Example: `-32768`, `32767`, `2#1111110001001001`, `16#9FA4`.

inter-controller network

An Ethernet-based network that is part of the control network, and provides data exchange between controllers and engineering tools (programming, asset management system (AMS)).

IODDT

(*input/output derived data type*) A structured data type representing a module, or a channel of a CPU. Each application expert module possesses its own IODDTs.

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 network.

IPsec

(*internet protocol security*) An open set of protocol standards that make IP communication sessions private and secure for traffic between modules using IPsec, developed by the internet engineering task force (IETF). The IPsec authentication and encryption algorithms require user-defined cryptographic keys that process each communications packet in an IPsec session.

isolated DIO network

An Ethernet-based network containing distributed equipment that does not participate in an RIO network.

L

LD

(*ladder diagram*) An IEC 61131-3 programming language that represents instructions to be executed as graphical diagrams very similar to electrical diagrams (contacts, coils, etc.).

literal value of an integer

A literal value of an integer is used to enter integer values in the decimal system. Values may be preceded by the "+" and "-" signs. Underscore signs (_) separating numbers are not significant.

Example:

```
-12, 0, 123_456, +986
```

local rack

An M580 rack containing the CPU and a power supply. A local rack consists of one or two racks: the main rack and the extended rack, which belongs to the same family as the main rack. The extended rack is optional.

local slave

The functionality offered by Schneider Electric EtherNet/IP communication modules that allows a scanner to take the role of an adapter. The local slave enables the module to publish data via implicit messaging connections. Local slave is typically used in peer-to-peer exchanges between PACs.

M

M580 Ethernet I/O device

An Ethernet device that provides automatic network recovery and deterministic RIO performance. The time it takes to resolve an RIO logic scan can be calculated, and the system can recover quickly from a communication disruption. M580 Ethernet I/O devices include:

- local rack (including a CPU with Ethernet I/O scanner service)
- RIO drop (including an X80 adapter module)
- DRS switch with a predefined configuration

main ring

The main ring of an Ethernet RIO network. The ring contains RIO modules and a local rack (containing a CPU with Ethernet I/O scanner service) and a power supply module.

MAST

A master (MAST) task is a deterministic processor task that is run through its programming software. The MAST task schedules the RIO module logic to be solved in every I/O scan. The MAST task has two 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.

MB/TCP

(*Modbus over TCP protocol*) This is a Modbus variant used for communications over TCP/IP networks.

MIB

(*management information base*) A virtual database used for managing the objects in a communications network. See SNMP.

Modbus

Modbus is an application layer messaging protocol. Modbus provides client and server communications between devices connected on different types of buses or networks. Modbus offers many services specified by function codes.

multicast

A special form of broadcast where copies of the packet are delivered to only a specified subset of network destinations. Implicit messaging typically uses multicast format for communications in an EtherNet/IP network.

N

network

There are two meanings:

- In a ladder diagram:
  A network is a set of interconnected graphic elements. The scope of a network is local, concerning the organizational unit (section) of the program containing the network.
- With expert communication modules:
  A network is a set of stations that intercommunicate. The term *network* is also used to define a group interconnected graphic elements. This group then makes up part of a program that may comprise a group of networks.

network convergence

Activity of re-configuring the network in situation of network loss to ensure system availability.

network time service

Use this service to synchronize computer clocks over the Internet to record events (sequence events), synchronize events (trigger simultaneous events), or synchronize alarms and I/O (time stamp alarms).

NIM

(*network interface module*) A NIM resides in the first position on an STB island (leftmost on the physical setup). The NIM provides the interface between the I/O modules and the fieldbus master. It is the only module on the island that is fieldbus-dependent — a different NIM is available for each fieldbus.

NTP

(*network time protocol*) Protocol for synchronizing computer system clocks. The protocol uses a jitter buffer to resist the effects of variable latency.

O

O->T

(*originator to target*) See originator and target.

ODVA

(*Open DeviceNet Vendors Association*) The ODVA supports network technologies that are based on CIP.

OFS

(*OPC Factory Server*) OFS enables real-time SCADA communications with the Control Expert family of PLCs. OFS utilizes the standard OPC data access protocol.

OPC DA

(*OLE for Process Control Data Access*) The Data Access Specification is the most commonly implemented of the OPC standards that provide specifications for real-time data communications between clients and servers.

operation network

An Ethernet-based network containing operator tools (SCADA, client PC, printers, batch tools, EMS, etc.). Controllers are connected directly or through routing of the inter-controller network. This network is part of the control network.

originator

In EtherNet/IP, a device is considered the originator when it initiates a CIP connection for implicit or explicit messaging communications or when it initiates a message request for un-connected explicit messaging.

P

PAC

programmable automation controller. The PAC is the brain of an industrial manufacturing process. It automates a process as opposed to relay control systems. PACs are computers suited to survive the harsh conditions of an industrial environment.

port 502

Port 502 of the TCP/IP stack is the well-known port that is reserved for Modbus TCP communications.

port mirroring

In this mode, data traffic that is related to the source port on a network switch is copied to another destination port. This allows a connected management tool to monitor and analyze the traffic.

PTP

(*precision time protocol*) Use this protocol to synchronize clocks throughout a computer network. On a local area network, PDP achieves clock accuracy in the sub-microsecond range, making it suitable for measurement and control systems.

Q

QoS

(*quality of service*) The practice of assigning different priorities to traffic types for the purpose of regulating data flow on the network. In an industrial network, QoS is used to provide a predictable level of network performance.

R

rack optimized connection

Data from multiple I/O modules are consolidated in a single data packet to be presented to the scanner in an implicit message in an EtherNet/IP network.

ready device

Ethernet ready device that provides additional services to the EtherNet/IP or Modbus module, such as: single parameter entry, bus editor declaration, system transfer, deterministic scanning capacity, alert message for modifications, and shared user rights between Control Expert and the device DTM.

RIO drop

One of the three types of RIO modules in an Ethernet RIO network. An RIO drop is an M580 rack of I/O modules that are connected to an Ethernet RIO network and managed by an Ethernet RIO adapter module. A drop can be a single rack or a main rack with an extended rack.

RIO network

An Ethernet-based network that contains 3 types of RIO devices: a local rack, an RIO drop, and a ConneXium extended dual-ring switch (DRS). Distributed equipment may also participate in an RIO network via connection to DRSs or BMENOS0300 network option switch modules.

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*) 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

S908 RIO

A Quantum RIO system using coaxial cabling and terminators.

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 distributing water).

scanner

A scanner acts as the originator of I/O connection requests for implicit messaging in EtherNet/IP, and message requests for Modbus TCP.

scanner class device

A scanner class device is defined by the ODVA as an EtherNet/IP node capable of originating exchanges of I/O with other nodes in the network.

service port

A dedicated Ethernet port on the M580 RIO modules. The port may support these major functions (depending on the module type):

- port mirroring: for diagnostic use
- access: for connecting HMI/Control Expert/ConneXview to the CPU
- extended: to extend the device network to another subnet
- disabled: disables the port, no traffic is forwarded in this mode

SFC

(*sequential function chart*) An IEC 61131-3 programming language that is used to graphically represent in a structured manner the operation of a sequential CPU. This graphical description of the CPU's sequential behavior and of the various resulting situations is created using simple graphic symbols.

SFP

(*small form-factor pluggable*). The SFP transceiver acts as an interface between a module and fiber optic cables.

simple daisy chain loop

Often referred to as SDCL, a simple daisy chain loop contains RIO modules only (no distributed equipment). This topology consists of a local rack (containing a CPU with Ethernet I/O scanner service), and one or more RIO drops (each drop containing an RIO adapter module).

SMTP

(*simple mail transfer protocol*) An email notification service that allows controller-based projects to report alarms or events. The controller monitors the system and can automatically create an email message alert with data, alarms, and/or events. Mail recipients can be either local or remote.

SNMP

(*simple network management protocol*) Protocol used in network management systems to monitor network-attached devices. 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.

SNTP

(*simple network time protocol*) See NTP.

SOE

(*sequence of events*) SOE software helps users understand a chain of occurrences that can lead to unsafe process conditions and possible shutdowns. SOEs can be critical to resolving or preventing such conditions.

ST

(*structured text*) An IEC 61131-3 programming language that presents structured literal language and is a developed language similar to computer programming languages. It can be used to organize a series of instructions.

sub-ring

An Ethernet-based network with a loop attached to the main ring, via a dual-ring switch (DRS) or BMENOS0300 network option switch module on the main ring. This network contains RIO or distributed equipment.

subnet 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.

switch

A multi-port device used to segment the network and limit the likelihood of collisions. Packets are filtered or forwarded based upon their source and destination addresses. Switches are capable of full-duplex operation and provide full network bandwidth to each port. A switch can have different input/output speeds (for example, 10, 100 or 1000Mbps). Switches are considered OSI layer 2 (data link layer) devices.

T

T->O

(*target to originator*) See target and originator.

target

In EtherNet/IP, a device is considered the target when it is the recipient of a connection request for implicit or explicit messaging communications, or when it is the recipient of a message request for un-connected explicit messaging.

TCP

(*transmission control protocol*) A key protocol of the internet protocol suite that supports connection-oriented communications, by establishing the connection necessary to transmit an ordered sequence of data over the same communication path.

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 two 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.

TFTP

(*trivial file transfer protocol*) A simplified version of *file transfer protocol* (FTP), TFTP uses a client-server architecture to make connections between two devices. From a TFTP client, individual files can be uploaded to or downloaded from the server, using the user datagram protocol (UDP) for transporting data.

TIME_OF_DAY

See `TOD`.

TOD

(*time of day*) The `TOD` type, encoded in BCD in a 32-bit format, contains this information:

- the hour encoded in an 8-bit field
- the minutes encoded in an 8-bit field
- the seconds encoded in an 8-bit field

NOTE: The eight least significant bits are not used.

The TOD type is entered in this format: xxxxxxxx: **`TOD#`**`<Hour>:<Minutes>:<Seconds>`

This table shows the upper/lower limits of each field:

Field	Limits	Comment
Hour	[00,23]	The leading 0 is displayed; it can be omitted during data entry.
Minute	[00,59]	The leading 0 is displayed; it can be omitted during data entry.
Second	[00,59]	The leading 0 is displayed; it can be omitted during data entry.

Example: `TOD#23:59:45`.

TR

(*transparent ready*) Web-enabled power distribution equipment, including medium- and low-voltage switch gear, switchboards, panel boards, motor control centers, and unit substations. Transparent Ready equipment allows you to access metering and equipment status from any PC on the network, using a standard web browser.

trap

A trap is an event directed by an SNMP agent that indicates one of these events:

- A change has occurred in the status of an agent.
- An unauthorized SNMP manager device has attempted to get data from (or change data on) an SNMP agent.

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. Unlike TCP, UDP does not include preliminary communication to establish data paths or provide data ordering and checking. However, by avoiding the overhead required to provide these features, UDP is faster than TCP. 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.

UMAS

(*Unified Messaging Application Services*) UMAS is a proprietary system protocol that manages communications between Control Expert and a controller.

UTC

(*coordinated universal time*) Primary time standard used to regulate clocks and time worldwide (close to former GMT time standard).

V

variable

Memory entity of type `BOOL`, `WORD`, `DWORD`, etc., whose contents can be modified by the program currently running.

VLAN

(*virtual local area network*) A local area network (LAN) that extends beyond a single LAN to a group of LAN segments. A VLAN is a logical entity that is created and configured uniquely using applicable software.

Index

E

F

H

I

L

M

N

O

P

Q

R

S

T

U

W