Quantum NOE 771 xx
Ethernet Modules
User Guide

840 USE 116 00   Version 5.0

31001913 04

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, 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 an imminently hazardous situation, which, if not avoided, **will result** in death, serious injury, or equipment damage.

WARNING

WARNING indicates a potentially hazardous situation, which, if not avoided, **can result** in death, serious injury, or equipment damage.

CAUTION

CAUTION indicates a potentially hazardous situation, which, if not avoided, **can result** in injury or equipment damage.

PLEASE NOTE Electrical equipment should be serviced only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material. This document is not intended as an instruction manual for untrained persons.

About the Book

At a Glance

Document Scope

This manual describes the functionality of the Quantum NOE 771 xx product line. The following Programmable Logic Controllers (PLC) modules are covered.

- **140 NOE 771 00**
- **140 NOE 771 01**
- **140 NOE 771 10**
- **140 NOE 771 11**

This manual should provide you with the knowledge to begin using a Quantum PLC to communicate with devices over an Ethernet network. This manual includes information about:
- Hardware architecture of a Quantum Ethernet TCP/IP module designed to fit into a single slot on the standard Quantum backplane
- Capabilities of the NOE 771 xx modules
- Installation of the NOE 771 xx module on to a Quantum backplane

This manual describes the procedures for:
- Configuring the module from your programming panel using the *Concept* software
- Setting up the modules to transfer data using one of three methods:
  - Communication Blocks
    Use either the special MSTR instruction from the 984 Ladder Logic instruction set or instructions from the IEC Logic functions.
  - Global Data (Publish / Subscribe) Utility
  - I/O Scanner
    The I/O scanner modules (NOE 771 -00, -01, -11 only) include configuration procedures for the I/O scan list using either Concept, ProWORX, or Modsoft.
- Using an embedded Web server to access diagnostics and online configurations for the module and its associate controller
- Using the FatoryCast Web server to customize your configuration via embedded Web pages (140 NOE 771 -10, -11)

- Using the NOE in a Hot Standby solution that provides fault tolerance for the remote I/O and communications
- Using the Network Options Ethernet Tester with a Windows-based PC to monitor the network

Nomenclature

The following table describes the naming scheme.

140 NOE 771		Model Numbers
xx	refers to	-00, -01, -10, -11
x0	refers to	-00, -10
x1	refers to	-01, -11
0x	refers to	-00, -01
1x	refers to	-10, -11

Who Should Use This Manual?

This manual is intended to support anyone using a Quantum PLC that needs to communicate with devices over an Ethernet network. You are expected to have some knowledge about the use of PLC systems and possess of a working knowledge of either the Concept, ProWORX NxT, or Modsoft programming tools. You also need to understand the use of an Ethernet network and TCP/IP.

Validity Note

The data and illustrations found in this book are not binding. We reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be construed as a commitment by Schneider Electric.

Related Documents

Title of Documentation	Reference Number
Concept 2.6 User's Manual	840 SPU 503 00
BOOTP Lite User Documentation	31002087
FactoryCast User Guide	31002839.00
Hot Standby User Guide	840 USE 106 00
Ladder Logic Library User Guide	31002840 00
MODBUS Protocol Reference Guide	31002841 00
Open MODBUS Specification	www.modicon.com/openmbus
ProWORX NxT User Guide	372 SPU 680 01 NMAN
RIO Manual	31002842 00

Related Electronic Documentation

Product Related Warnings

Schneider Electric assumes no responsibility for any errors that may appear in this document. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

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

User Comments

We welcome your comments about this document. You can reach us by e-mail at TECHCOMM@modicon.com

Product Description

1

At a Glance

Introduction

This chapter provides product overviews of the Quantum 140 NOE 771 xx modules.

What's in this Chapter?

This chapter contains the following topics:

NOE 771 xx Module Overview

Overview

The following information provides overviews of all Quantum 140 NOE 771 xx modules.

General Description

The Quantum 140 NOE 771 00,10/100 Ethernet module, shown below, is one of the latest models in a line of Quantum Ethernet TCP/IP modules designed to make it possible for a Quantum Programmable Logic Controller (PLC) to communicate with devices over an Ethernet network. The electronics for the NOE 771 xx modules are contained in a standard Quantum single width housing that takes up one slot in a Quantum backplane. The module, which is capable of being hot swapped, can be plugged into any available slot in the backplane.
The NOE 771 00 provides real-time peer-to-peer communications as well as I/O scanning and a MODBUS/TCP server. The included HTTP services provide maintenance and configuration utilities to the module.
The NOE 771 10 provides all the services of the -00 except the I/O Scanner. It also has the following additional features.

- User programmable web pages
- FactoryCast application, including
  - Creating and viewing of graphic real-time templates using Java beans
  - Creating and viewing of text real-time templates in spreadsheet format
  - Using Concept symbols or direct addresses

The following figure shows the Quantum 140 NOE 771 00,10/100 Ethernet module.

Key Features

The key features of the **140 NOE 771 (-00, -01, -10, -11)** models are listed below:

	-00	**-01**	**-10**	**-11**
HTTP Server	X	X	X	X
BOOTP Client	X	X	X	X
BOOTP Server	X	X	X	X
SNMP V2 Agent	X	X	X	X
Flash File System	X	X	X	X
FTP Server	X	X	X	X
MODBUS Messaging	X	X	X	X
MODBUS I/O Scanning	X	X		X
Hot Standby	X	X	X	X
Global Data - Publish / Subscribe		X		X
Bandwidth Monitoring		X		X
Faulty Device Replacement (DHCP Server)		X		X
Enhanced Web Diagnostics		X		X
Schneider Private MIB		X		X
FactoryCast Application			X	X
User Programmable Web Pages			X	X

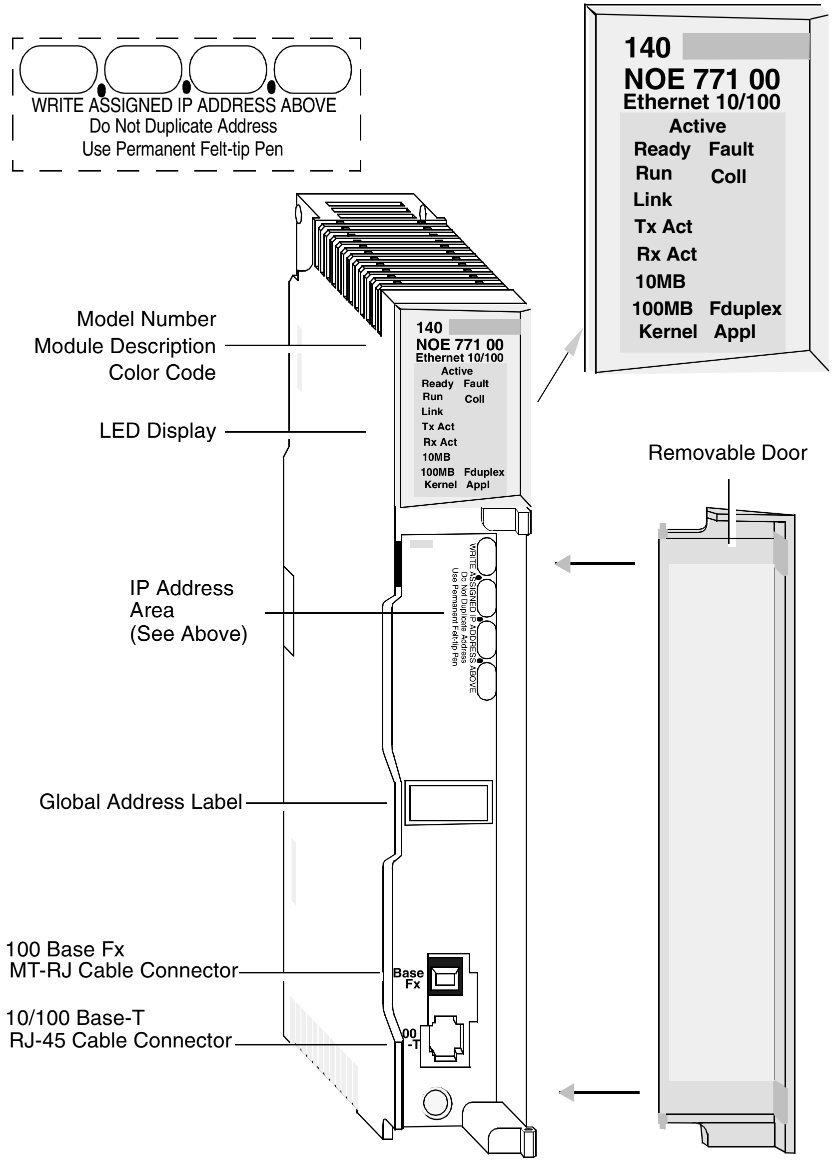
Front Panel Components

The front panel of the NOE 771 xx modules contain identification markings, color codes, and LED displays. A writable area for an Internet Protocol (IP) address, a global address label, and two Ethernet cable connectors is located behind the removable front panel door.
The following table provides a description of the front panel components which are shown in front view figure.

Component	**Description**
LED indicator Panel	Indicates the operating status of the module, and the fiber optic and MODBUS communications networks it is connected to.
IP Address Area	Provides a writable area to record the module's assigned IP address.
Global Address Label	Indicates the module's global Ethernet MAC address assigned at the factory.
100 BASE-FX Connector	Provides an MT-RJ receptacle for connection to a 100 megabit fiber optic Ethernet cable.
10/100BASE-T Connector	Provides an RJ-45 receptacle for connection to a shielded, twisted pair Ethernet cable.

Front View

The following figure shows the front of the NOE 771 00 Ethernet module. The 140 NOE 771 xx modules are identical, with the exception of the module description.



LED Indicators

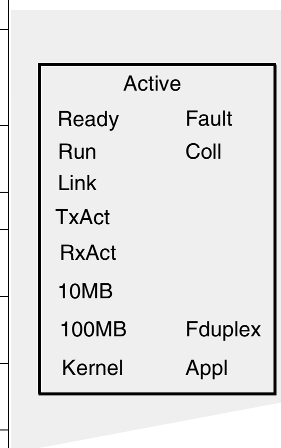
Overview

The following information describes the LED indicator panel.

LED Indicator Panel

The LED indicator panel provides continuous operating information about the NOE 771 xx modules and their connection to the network.
The following table describes the function of each LED indicator on the LED indicator panel.

LED	Color	Description
Active	Green	Indicates the backplane is operating.
Ready	Green	Indicates module is healthy.
Fault	Red	Indicates when the NOE is in a crash state.
Run	Green	Flashes to indicate diagnostic code, as described in "Run LED Status" (below).
Coll	Red	Flashes when Ethernet collisions occur.
Link	Green	On when Ethernet link is active.
TxAct	Green	Flashes to indicate Ethernet transmission.
RxAct	Green	Flashes to indicate Ethernet reception.
Kernel	Amber	On when in Kernel Mode. Flashing while in download mode.
10MB	Green	On when the module is connected to a 10 Megabit network.
100MB	Green	On when the module is connected to a 100 Megabit network.
Fduplex	Green	On when Ethernet is operating in the full duplex mode.
Appl	Green	On when crash log entry exists.



Run LED Status

The following table lists each available state of the *Run LED* indicator.

Indicator State	Status
On (steady)	Normal operation: The NOE module is ready for network communication.
Number of flashes in sequence	
one	Not used
two	Not used
three	No Link: the network cable is not connected or is defective
four	Duplicate IP address: The module will stay off-line.
five	No IP address: The module is attempting to obtain an IP address from a BOOTP server.
six	Using default IP address
seven	No valid executive NOE present
eight	Invalid IP configuration (Likely cause: Default gateway is not on the same subnet mask as the NOE.)

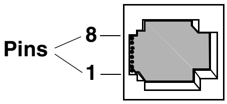
Connectors and Cabling

Overview

The following information describes the 10/100 BASE-T and 100 BASE-FX connectors.

10/100 BASE-T Twisted Pair Connector

The NOE 771 xx modules' 10/100 BASE-T connector (shown below) is a standard RJ-45 twisted pair receptacle.
The following figure shows the 10/100 BASE-T connector.



Schneider Automation recommends that you use Category 5 STP cabling, which is rated to 100 Mbps, with an RJ-45 connector.
The eight pins are arranged vertically and numbered in order from the bottom to the top. The RJ-45 pinout used by this module is:
- Receive Data (+)3
- Receive Data (-)6
- Transmit Data (+)1
- Transmit Data (-)2

100 BASE-FX

The NOE 771 xx modules' 100 BASE-FX connector is a MT-RJ receptacle or a mating fiber optic cable connector. (See the figure in the Front View block in *NOE 771 xx Module Overview, p. 16*).
For the NOE 771 xx, you may need an MT-RJ to SC (Duplex) Multimode fiber optic cable assembly 62.5/125mm. Schneider Electric recommends Cable Number 490NOC00005 to connect to fiber hubs/switches.

Note: The NOE 771 xx is a one channel device. It is capable of communicating over either a 10/100BASE-T or a 100BASE-FX Ethernet network at any given time, **but not over both at the same time**.

I/O Scanner (140 NOE 771 00, -01, -11 only)

Overview

The following information describes the MODBUS I/O Scanner.

Introduction

The functionality of your NOE 771 00, -01, or -11 module is further enhanced by the addition of a MODBUS I/O Scanner that you can configure with either the Modsoft, Concept, or the ProWORX programming panel. This functionality allows you to transfer data between network nodes without using the communication instruction. You can configure the NOE 771 MODBUS I/O Scanner by either one of the following two methods:

- Peer Cop (available on NOE 771 00 only)
- Ethernet I/O Scanner

Note: It is recommended that you use the enhanced MODBUS I/O Scanner for all new installations. Peer Cop functionality is provided only as an easy migration path for an existing installation. The enhanced MODBUS I/O Scanner provides greater functionality than the Peer Cop based I/O scanner.

Peer Cop Based I/O Scanner

The following table lists the characteristics of the Peer Cop based MODBUS I/O Scanner.

Parameter	Value
Max. No. of Devices	128
Max. No. of Input Words	500
Max. No. of Output Words	500
Health Timeout Value	Global Setting (20 Msec to 2 Secs in 20 mSec increments)
Input Timeout State	Global Setting (Zero or Hold)
IP Address	Derived from MODBUS Address (must be on NOE's Subnet)
Remote Register Reference	Not configurable - 400001 is used

Enhanced MODBUS I/O Scanner

The following table lists the characteristics of the Enhanced based MODBUS I/O Scanner.

Parameter	Value
Max. No. of Devices	64: 140 NOE 771 -00 only 128: 140 NOE 771 -01 and -11 only
Max. No. of Input Words	4000
Max. No. of Output Words	4000
Health Timeout Value	Individual Setting (1 Msec to 2 Secs in 1 mSec increments)
Input Timeout State	Individually set
IP Address	Individually set
Remote Register Reference	Configurable
Min. Update Rate	Set

Refer to the chapter *Transferring Data with the I/O Scanner 140 771- 00, -01, and -11* to learn how to configure the MODBUS I/O Scanner.

Performance

Refer to the Appendix *NOE 771 00 Module I/O Scanner Performance Statistics* for detailed performance data.

MODBUS Messaging

Overview

The following information describes the functionality of the MODBUS/TCP Server.

Introduction - Client

All NOE 771 xx Quantum Ethernet TCP/IP modules provide the user with the capability of transferring data to and from nodes on a TCP/IP network through the use of a communication instruction. All PLCs that support networking communication capabilities over Ethernet can use the MSTR ladder logic instruction to read or write controller information or can also use IEC communication blocks.

Introduction - Server

All NOE 771 xx Quantum Ethernet TCP/IP modules provide the user with the ability to access data from the controller using the standard MODBUS/TCP protocol. Any device: PC, HMI package, another PLC, or any MODBUS/TCP compliant device can access data from the PLC. The MODBUS/TCP Server also allows Programming Panels to log into the controller over Ethernet.

Limitations

The NOE 771 xx supports up to 64 simultaneous MODBUS/TCP Server connections. The NOE 771 xx allows only one Programming Panel to be logged in at a time to guarantee consistency of changes to the controller configuration.
The following MODBUS/TCP commands are supported by the NOE:
- Read Data
- Write Data
- Read/Write Data
- Get Remote Statistics
- Clear Remote Statistics
- MODBUS 125 Commands (used by Programming Panels to download a new Exec to the NOE)

Performance

The following table shows the performance characteristics of the NOE 771 xx's MODBUS/TCP Server.

Parameter	Value
Typical Response Time (mSec)	0.6
Number of MODBUS Connections (Client and Server)	64 (-01, -11) 16 (Client -00) 32 (Server -10)
Number of Simultaneous Login Channels	1

Note: NOE 771 xx MODBUS/TCP performance measurements are made with 140 CPU 534 14.

FTP and HTTP Server

Overview

The following information describes the services provided by the FTP and HTTP servers.

FTP Server

The NOE 771 xx's File Transfer Protocol (FTP) server is available as soon as the module receives an IP address. Any FTP client can log on to the module, if the client uses the correct user name and password.

The FTP server provides the following services:

- Update the NOE's firmware by downloading a new Exec
- Provides error log visibility by uploading error log files
- Upload/download BOOTP server and SNMP configuration files

The default user name is USER, and the default password is USER. Both the user name and password are case sensitive. Refer to *Installing the Module, p. 35* for instructions about how to change the password, and how to add or delete user names to the FTP server.

There should be only one FTP client per module.

HTTP Server

The NOE 771 xx's HyperText Transport Protocol (HTTP) server is available as soon as the module receives an IP address. It can be used with version 4.0 or greater of either the Internet Explorer or Netscape browser.

The NOE 771 xx's HyperText Transport Protocol (HTTP) server allows you to view the following information:

- Module's Ethernet statistics
- Controller and I/O information
- BOOTP/DHCP/FDR server information
- Global Data (Publish / Subscribe)

The HTTP server's HTML pages allow you to configure the module's BOOTP/DHCP/FDR server and SNMP Agent.

The HTTP server is protected with a default name and password. The default name and password are both USER, and both are case sensitive. They can both be changed via the Configuration page on the NOE 771 0x's Web Embedded Pages (see *Installing the Module, p. 35*).

For the NOE 771 1x modules, they can be changed via the FactoryCast Configurator.

The NOE 771 xx supports a maximum of 32 HTTP instantaneous connections.

Note: Browsers may open multiple connections so 32 HTTP connections does not indicate 32 simultaneous users.

Note: The NOE 771 0x module does not support user downloaded Web pages. You will need to purchase the 140 NOE 771 1x module to support that requirement.

Address Server

Overview

The following information describes the services provided by the Address servers.
- BOOTP server
- DHCP server

BOOTP Server

Note: The BOOTP server is available on the 140 NOE 771 -00 and -10 models.

The BOOTstrap Protocol (BOOTP) software, compliant with RFC 951, is used to assign IP addresses to nodes on an Ethernet network. Devices (hosts) on the network issue BOOTP requests during their initialization sequence, and a BOOTP server that receives the request will extract the required IP address information from its database and place it in BOOTP response messages to the requesting devices. The devices will use the assigned IP addresses, received from the BOOTP server, for all communication occurring on the network.

Your NOE BOOTP Server

Your NOE x0 module comes supplied with a BOOTP server. This feature allows you to provide IP addresses to all the I/O devices being serviced by the NOE 771 x0. Providing a BOOTP server that is built into your NOE 771 x0 module eliminates the need for you to have a dedicated PC on your I/O network acting as a BOOTP server.

Note: The NOE 771 x0's BOOTP server cannot be used to provide its own IP address.

You can configure your NOE 771 x0's BOOTP server from the module's HTTP Web page. Using this feature allows you to add, remove, and edit devices to the BOOTP server's database, which is maintained on the modules non-volatile memory.

DHCP Server

Note: The DHCP server is available on the 140 NOE 771 -01 and -11 models. Dynamic Host Configuration Protocol (DHCP) is a superset of the BOOTP Protocol. Your 140 NOE 771 x1 has a DHCP Server. The DHCP Server is compliant with RFC 1531. The DHCP Server can be used to provide the IP Configuration to devices using BOOTP or devices using DHCP.
The DHCP Server has entries that use the MAC Address to serve the IP Configuration and entries in the server that use the Role Name to serve the IP Configuration. See the *Address Server Configuration / Faulty Device Replacement* chapter for details on configuring your NOE's Address Server.
If you are migrating a BOOTP Configuration from a 140 NOE 771 x0 module to the new 140 NOE 771 x1 module, see the *Address Server Configuration / Faulty Device Replacement* chapter for details on automatic upgrade of your configuration for the new DHCP Server.

Note: OPERATING ON A CORPORATE NETWORK
Before placing the NOE on a corporate network, Schneider Automation recommends that you discuss the installation with your MIS department. It is likely that your company's corporate network has at least one DHCP Server running already. If the NOE's DHCP server is running on the same network, it may disturb the network.
To avoid any possible problem related to the NOE's DHCP server on the corporate network, you must ensure that the DHCP server is not running in the NOE by not having Address Entries in the Configuration. If there are no configured devices in the Address Server Configuration page, then the NOE will not start the DHCP server.

Global Data

Overview

Global Data service is a real time Publisher/Subscriber mechanism providing the most efficient data exchange for PLC application coordination.
Devices supporting Global Data are grouped in a distribution group for the purpose of application variable exchange and synchronization. Each Global Data device can Publish up to one network (application) variable and Subscribe up to 64 Network (application) variables.
The Quantum NOE's embedded Web Global Data Configuration page provides a configuration screen to determine which and how many application variables are exchanged with this service. After configuration, the exchanges between all stations belonging to the same distribution group are done automatically.
The Global Data Service uses the 4x Register Space for Global Data Exchanges.

Key Features of Global Data

The main features for Global Data are:

- One Publisher and many Subscribers
- A Device can publish one Network Variable of up to 512 Registers
- A Device can subscribe to several Network Variables of up to 2048 4x Registers
- A Device subscribes to the complete Network Variable
- One distribution group per network IP address
- Application defined Publication rate
- Up to 64 Global Data Network Variables (numbered from 1 to 64) can be part of the data distribution group
- An NOE has only one multicast address; consequently, it can only Publish and Subscribe inside the group
- A Device can participate in several Distribution Groups by using multiple NOEs in the Rack

Global Data has an advantage over Client / Server services when more than one Subscriber is receiving the same data since only one transaction is necessary for all Subscribers to receive the data.
This advantage offers two benefits:

- Reduce overall network traffic
- Ensure tighter synchronization of multiple subscribers

Bandwidth Monitoring

Overview

Bandwidth Monitoring allows you to monitor the NOE's CPU allocation for each of the following services: Global Data, I/O Scanning, and Messaging. The Bandwidth Monitoring service retrieves workload data and returns one of two pieces of information: whether the module has free resources or whether the module is working at capacity. Knowing the resource allocation helps you:

- Decide about allocating your resources
- Determine the number of NOEs needed in a system

Available Services

The services accessed and monitored are:

- Global Data
- I/O Scanner
- MODBUS Messaging

If you use Bandwidth Monitoring, you do not need to develop a new set of access functions. The actual NOE CPU load is computed each second.

Bandwidth Monitoring Load Rates

The Bandwidth Monitoring service checks once a second and computes four (4) values in private data:
- Percentage of NOE's CPU allocated to **Global Data**
- Percentage of NOE's CPU allocated to the **I/O Scanner**
- Percentage of NOE's CPU allocated to **Messaging**
- Percentage of NOE's CPU allocated to other services and idle

Results are returned as percentages. CPU time spent in other services is shown as "Other" or "Free." Bandwidth Monitoring uses the same functions as used by SNMP.
The three service rates, Global Data, I/O Scanner, and Messaging, are computed using the following formula:

```
(Current load * 100) / Maximum Load
```

Table of **Maximum Load Rates**

Diagnostic Service	**Workload Data Returned**	**Maximum load for NOE 771 x1**
Global Data	Number of published variables per second	800
I/O Scanner	Number of transactions per second	4200
Messaging	Number of messages treated per second	410

The current load is computed dynamically.

Note: The loads are dependent on controller scan time. Each application has an expected scan time. Therefore, when evaluating the loads, you should ensure that the controller scan time is set to the expected scan time for the application being modelled.

Web Diagnostics

Overview

Note: These services are available on the 140 NOE 771 x1 modules.

The embedded Web server provides Web pages that you may use to diagnose Transparent Factory / Real Time services.
Those diagnostic services are listed below:
- Global Data diagnostics
  - Status of all Global Data services
  - Status of all subscribed and published variables
  - Publication / Subscription rate
- I/O Scanning diagnostics
  - Status of all I/O Scanning services
  - Status of individual scanned devices
  - Actual I/O scanning rate
- Messaging diagnostics
  - Diagnostic information for Port 502 messaging
- Bandwidth Monitoring
  - Throughput measurement of NOE by service

Note: All these pages are protected by the general HTTP password.

System Requirements and Customer Support

Minimum System Requirements

The following table lists the minimum version requirements for systems used with the NOE 771 x0 modules.

System	Minimum Version Number
Quantum Executive	2.0
Concept	2.2
Modlink	2.0
Modsoft	2.6
ProWORX NxT	2.0 IP Address Configuration 2.1 I/O Scanning

Note: If you are unable to configure your NOE 771 01 or NOE 771 11 modules in your version of Concept, please configure using NOE 771 00 part numbers to represent the NOE 771 01 and NOE 771 11 modules.

Customer Support

If you have any problems, please first consult the documentation listed above or the MS-Windows documentation. If you still have a question or need assistance, help is available from our Schneider hotline.

- Tel: USA and Canada 800-468-5342
- Tel: International 978-975-9557
- Fax: All 978-975-9301
- BBS: Bulletin Board 978-975-9779

When calling the Schneider 800 telephone number, you will get a recording asking you to enter a one-digit code for the type of service you request, provided you use a touch tone telephone.

Visit Our Web Site: Please access the Schneider Web site, *www.modicon.com* or schneider.com for the most up-to-date NOE Ethernet Controller information, such as resolutions to product issues, and product announcements. When you access the Web site, look under technical information, and choose Quantum from the list of cross-product families. Then access Resolutions for resolutions to product issues, Product Manuals for the most recently published user documentation, and so on.

Installing the Module

2

At a Glance

Introduction

This chapter contains installation and configuration information for the NOE 771 xx modules.

What's in this Chapter?

This chapter contains the following topics:

Before You Begin

Overview

The following information describes how to install the NOE 771 xx module.

Initial Checks

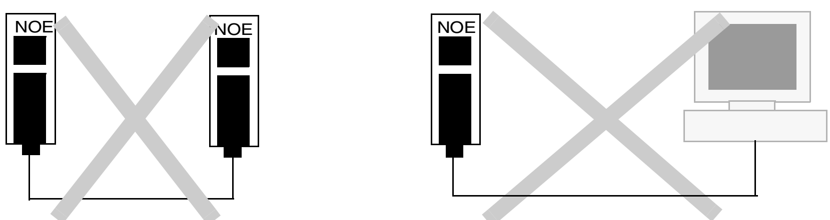
Before you install your module, you need to complete the following checks.
- Determine how the NOE 771 xx module will be assigned its Ethernet address parameters (the default method is BOOTP)
- Verify that your Ethernet network is properly constructed

Determining the Appropriate Ethernet Address Parameters

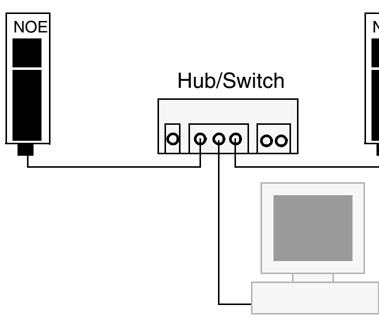
Consult your system administrator to determine if you must configure a new IP address and appropriate gateway and subnet mask addresses, or whether the module will obtain its Ethernet address parameters from a BOOTP server. If the administrator assigns new address parameters, you will need to configure the module from your programming panel. Follow the directions in the *Configuring the Module with Concept* chapter.

Verifying the Network Topology

You should not use a length of cable to connect an Ethernet web embedded server module directly to another device. For the network to operate properly, you must route the cable for each device through an Ethernet hub/switch. Hubs/switches are widely available and can be purchased from many suppliers.
The following figure shows two incorrect network topologies.



The following figure shows a correct network topology.



Cabling Schemes

Overview

The following information describes how to connect devices in a standard Ethernet cabling.

Introduction

In a standard Ethernet cabling scheme, each device connects via a cable to a port on a central Ethernet hub/switch.

Twisted Pair Length

The purpose of the following table is to show that the maximum length of cable between devices depends on the type of device.

Type of Device	Max. Cable from Device to Hub	Max. Hubs Between Any Two Nodes	Max. Cable Between Most Distant Nodes on Network
Hub	100 m	4	500 m
Switch	100 m	Unlimited	Unlimited

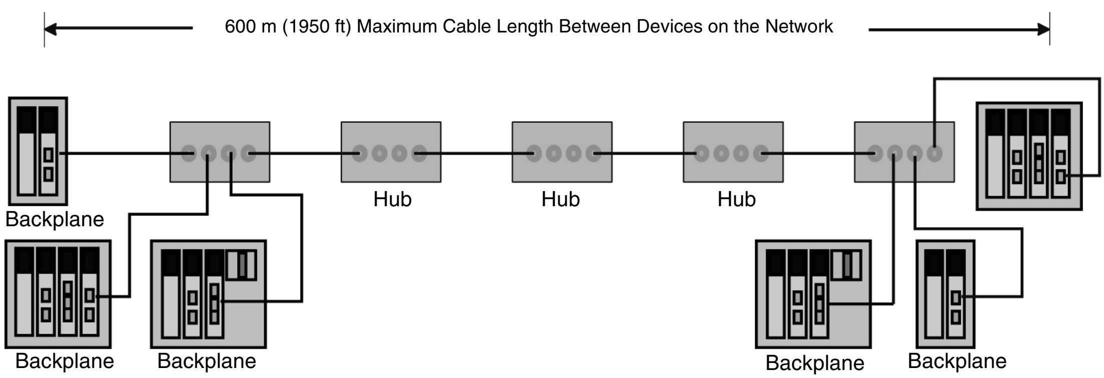
For Fast Ethernet (100 Base-T) specifications, please refer to the IEEE 802.3u Standard available from the IEEE (*www.IEEE.org*).

Cabling with Traditional Hubs

The figures and tables that follow show the maximum number of hubs and the maximum cable length between devices allowed if using hubs.

10 BASE-T Cable Distances

The following figure is for 10 BASE-T cable



100 BASE-T Cable Distances

The 100 BASE-T cabling allows for two hubs with a link maximum distance of 100 m (325 ft), and a total network diameter of 205 m (665 ft).
The following table provides information about the maximum distance parameters with 100 BASE-T.

Model	Length max. in Twisted pair TX-T2-T4
DTE-DTE (no repeater)	100 m (325 ft)
One Class I repeater	200 m (650 ft)
One Class II repeater	200 m (650 ft)
Two Class II repeaters	205 m (665 ft)

100 BASE-FX Cable Distances

The 100 BASE-FX cabling allows for two hubs with a link maximum distance of 412 m (1339 ft), and a total network diameter of 205 m (665 ft).
The following table provides information about the maximum distance parameters with 100 BASE-FX and 100 BASE-TX-FX.

Model	Length max. Twisted pair TX and Fiber FX	Length max. Fiber FX
DTE-DTE (no repeater)	n.a.	412 m (1339 ft)
One Class I repeater	260.8 m (*1*)	272 m (884 ft)
One Class II repeater	308.8 m (*1*)	320 m (1040 ft)
Two Class II repeaters	216.2 m (*2*)	228 m (741 ft)
(1) Mixed twisted pairs and fiber assumes a 100 m (325 ft) twisted pair links		
(2) Mixed twisted pairs and fiber assumes a 105 m (340 ft) twisted pair links		

Fiber Length

The maximum length for 850 nm/Multimode cable is 2 KM.

Security

Overview

The following information describes firewalls. A firewall is a gateway that controls access to your network. To restrict access to your Ethernet controller and I/O network, you may want to consider a firewall.

Types of Firewalls

There are two types of firewalls.
- Network-level firewalls
- Application-level firewalls

Network-Level Firewalls

Network-level firewalls are frequently installed between the Internet and a single point of entry to an internal, protected network.

Application-Level Firewalls

An application-level firewall acts on behalf of an application. It intercepts all traffic destined for that application and decides whether to forward that traffic to the application. Application-level firewalls reside on individual host computers

Port Numbers Used by NOE

The following table contains the port numbers used by NOE

Protocol	Port Number
Modbus/TCP	TCP 502
HTTP	TCP 80
SNMP	UDP 61
FTP	TCP 21

You may need to provide the information in this table to your system administrator so that the firewall configuration will allow access to your PLC from outside of your facility.

Installing the Module

Overview

The following information describes how to install the NOE 771 module.

Before You Begin

Locate the backplane in which you will mount the NOE 771 xx module. Ensure that an open slot is available in which to mount the module.

Note: The NOE 771 xx module can be installed only in a local backplane.

Note: Power Requirements
- Ensure when installing the NOE that the NOE does not exceed the Quantum backplane requirements.

Backplane Slot Placement

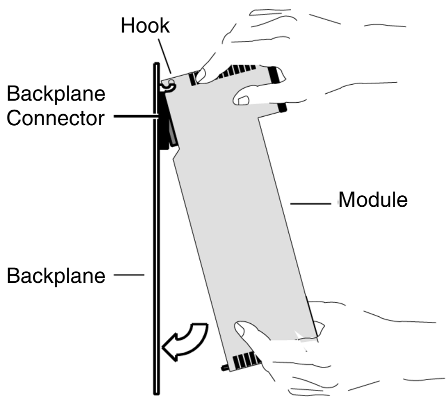
The modules may be placed in any slot on the backplane. They do not have to be placed next to each other.

Tools Required

You will need one medium-size, Phillips-head screw driver.

Mounting the Module in the Backplane

Perform the following steps to mount the NOE 771 xx module on to a Quantum backplane.

Step	Action
1	Holding the module at an angle, mount it on the two hooks located near the top of the backplane. The following figure shows the correct way to hold the module. 
2	Swing the module down so its connector engages the backplane connector.
3	Using a Phillips-head screw driver, tighten the screw at the bottom of the module between 2 and 4 in-lbs or between .22 and .45 Newton meters of torque.

Connecting the Cable

Overview

The following information pertains to cabling.

Note: The 140 NOE 771 xx is capable of communicating over either a 10/100BASE-T or a 100BASE-FX Ethernet network at any given time, **but not both at the same time**.

Accessories

The following are switches that are sold by Schneider Automation.

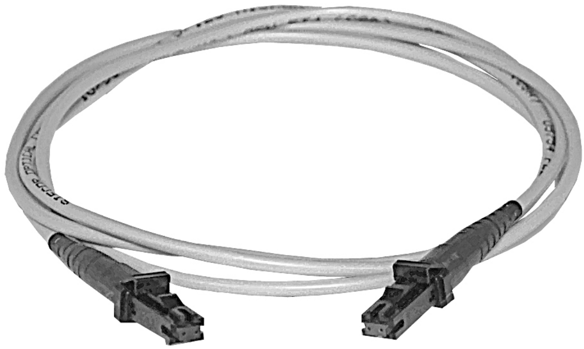
Hub or Switch	Description
499NEH10410	Hub with 4 ports 10BASE-T
499NOH10510	Hub with 3 ports 10BASE-T and 2 ports 10BASE-FL
499NTR10010	Transceiver 10BASE-T / 10BASE-FL
499NEH14100	Hub with 4 ports 100BASE-TX
499NTR10100	Transceiver 100BASE-TX
499NES18100	Switch with 8 ports 10/100BASE-TX
499NES17100	Managed Switch with 7 ports 10/100BASE-TX
499NOS17100	Managed Switch with 5 ports 10/100BASE-TX and 2 ports 100BASE-FX

The following are Schneider Automation cables that support Multicast Filtering

Cable	Description
490NTW000 02/05/12/40/80 U	StraightThru cable
490NTC000 05/15/40/80 U	Crossover cable

Fiber Optic

Remove the protective plug from the module's MT-RJ connector port and the protective cap from the tip of the black connector on the MT-RJ fiber optic cable (as shown in the following figure). Note the position of the keyway on the module's connector port and the matching key on the cable connector, and then insert the connector into the port. It should snap into place.
The following figure shows MT-RJ fiber optic cable.



Assigning Ethernet Address Parameters

Overview

The following information describes how to assign IP address parameters.
As shipped from the factory, the NOE 771 xx module does not contain an IP address. If you have not programmed the unit with an Ethernet configuration extension, the module does not contain an IP address. When the module starts up without an IP address, the module will attempt to obtain an IP address from the network's BOOTP server.
You can use Concept or ProWORX to assign an IP address, default gateway and sub network mask. See the section, *Configuring the Ethernet Address Parameters*.
You can also assign IP address parameters using the BOOTP Lite software utility. See Using BOOTP Lite to Assign Address Parameters in this chapter.

Note: You can not configure the IP address using Web pages.

Using a BOOTP Server

A BOOTP server is a program that manages the IP addresses assigned to devices on the network. Your system administrator can confirm whether a BOOTP server exists on your network and can help you use the server to maintain the adapter's IP address.

How an Unconfigured ("as shipped") Module Obtains an IP Address

On startup, an unconfigured NOE 771 xx module will attempt to obtain an IP address by issuing BOOTP requests. When a response from a BOOTP server is obtained, that IP address is used. If no BOOTP response is received within two minutes, the module uses the default IP address.

Setting the Default IP Address

To set the default IP address using your PC, set up an active route from your PC. To do this with either Windows 95/98/2000/ME/XP or Windows NT, use the following procedure.

Step	Action
1	Be sure the NOE module is running.
2	Obtain the default IP address of the NOE.
3	Open an MS-DOS Window.
4	Print the currently active routes by typing `C:>route print`
5	Add an active route for the local NOE by typing `C:\>route add 84.0.0.0 mask 255.0.0.0 205.217.193.205` The result is that Windows 95/98/ NT will now talk to any address that starts with an 84 that is directly connected to a hub or switch directly accessible to your machine, or that can be seen by the route/gateway specified.
6	Confirm that there is a new entry in the active route table by typing `C:>route print`: The following figure confirms that the new entry was added to the active route table. Active Routes: Network Address / Netmask / Gateway Address / Interface / Metric 0.0.0.0 / 0.0.0.0 / 205.217.193.250 / 205.217.193.205 / 1 84.0.0.0 / 255.0.0.0 / 205.217.193.205 / 205.217.193.205 / 1 127.0.0.0 / 255.0.0.0 / 127.0.0.1 / 127.0.0.1 / 1
7	Verify that a connection is made by typing `C:\>ping 84.0.0.2` The following figure shows that the connection is verified. Reply from 84.0.0.2: bytes=32time=1msTTL=32 Reply from 84.0.0.2: bytes=32time=1msTTL=32 Reply from 84.0.0.2: bytes=32time=1msTTL=32 Reply from 84.0.0.2: bytes=32time=1msTTL=32

Specifying Address Parameters

Consult your system administrator to obtain a valid IP address and an appropriate gateway and a subnet mask, if required. Then follow the instructions in the section, *Configuring the Ethernet Address Parameters*.

Assigning an IP Address via Concept's or ProWORX's "Specify IP Address" Option

You can select the NOE 771 xx module's Specify IP Address mode via Concept or ProWORX to assign an IP address (as well as default gateway and sub network mask) to the module.

Assigning an IP Address via Concept's or ProWORX's "Use BOOTP Server" Option

You can select the NOE 771 xx module's Use BOOTP Server mode via Concept or ProWORX to instruct the module to obtain its IP address from a network BOOTP server. In this mode, only an address obtained from a BOOTP server will be accepted by the module.

If BOOTP Server Responds

If the server responds with address parameters, the NOE 771 xx module will use those parameters as long as power remains applied to the module.
If the server does not respond, the module will retry its request for two minutes.

If BOOTP Server Does Not Respond

If no BOOTP response is received, the NOE 771 xx module will use the default IP Address.
During this time the Run indicator will display a pattern of five flashes for a BOOTP attempt and six flashes for using the default IP.

NOE 771 xx Duplicate IP Address Test

In all cases, when the NOE 771 xx module receives an IP address, it will test for duplicate addresses by sending broadcast ARP requests three times at 5 second intervals.
If a Duplicate IP Address is found on the network, the NOE 771 xx will stay off-line to avoid a network disruption. It will display a pattern of four flashes to indicate a Duplicate IP Address detection.

Gratuitous ARP

If there are no replies to its requests, the NOE 771 xx will send gratuitous ARP three times at 2 second intervals to announce its presence on the network.

Establishing the FTP Password

Overview

The following information describes how to set the FTP Password.

Establishing the FTP Password

The FTP Password is established using the Embedded Web Server. This section contains information about initially accessing the web server. The first thing the system administrator should do upon accessing the web server is change the FTP password. Doing this restricts access to the web server functionality to the system administrator.

This section contains information on how to access the web server for purposes of changing the FTP and HTTP passwords. See the chapter *Embedded Web Pages* for detailed information about the web server pages and their functionality.

Introduction to Accessing the Web Server

Each Quantum 140 NOE 771 xx module contains an embedded Web server, which allows you to access diagnostics and online configurations for the module and its associated controller (PLC).

The web pages can only be viewed across the World Wide Web using version 4.0 or greater of either Netscape Navigator or Internet Explorer, both of which support JDK 1.1.4 or higher.

For information about the additional functionality provided by the FactoryCast system in the 140 NOE 771 1x modules, see the *FactoryCast Manual*, 890 USE 152 00.

How to Access the Web Server

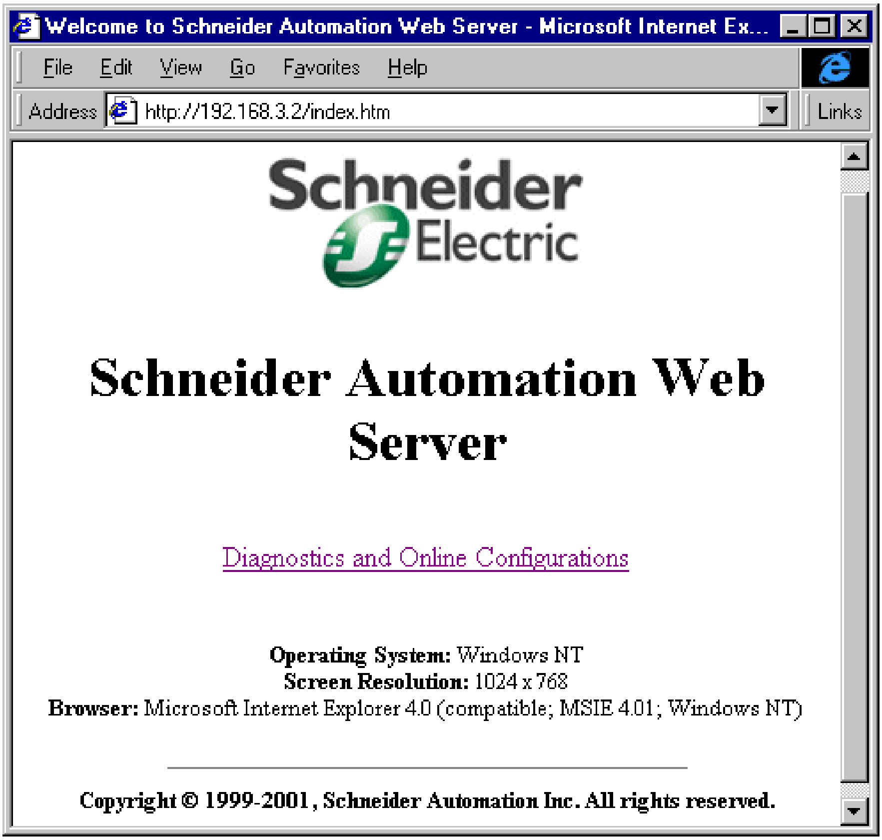
Before you can access the module's home page, you must enter the full IP address or URL in the Address or Location box in the browser window.

For example: *http://hostname* (*hostname* is full IP address or DNS host name.)

After you do this, the Schneider Automation Web Utility home page displays.

Schneider Web Utility Home Page

The following figure shows the Schneider Automation Web Utility home page

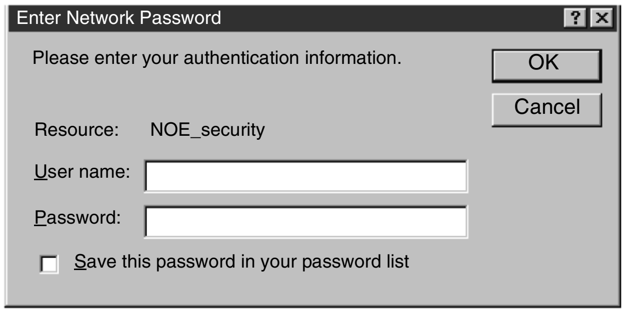


From this page, you can access the following pages.

- Access the pages to change the FTP password
- Access the pages to change the HTTP password
- Access the pages for diagnostic and configuration information, see *Embedded Web Pages, p. 157* for further information.

Modifying the FTP Server Password

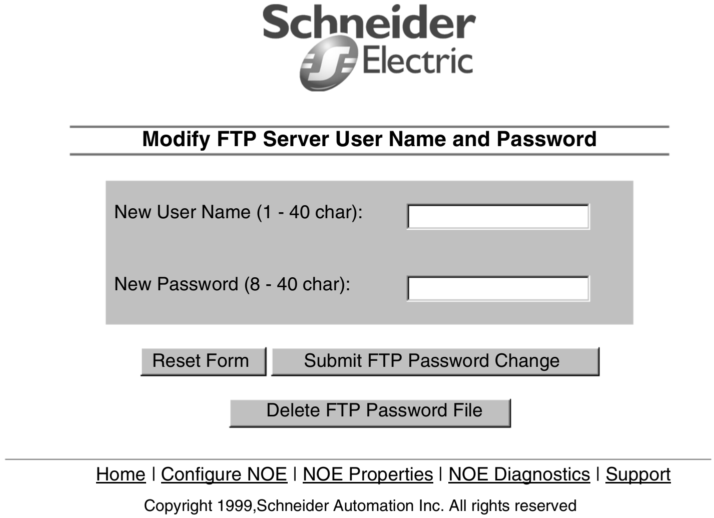
The following steps detail how to link to the correct web page for modifying the FTP Password

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Enter the URL, for example, http://hostname/secure/embedded/ftp_passwd_config.htm</td></tr>
<tr><td>2</td><td>The result of step 1 is that the user is requested to supply a user name and password, as shown in the figure that follows.
The following figure shows the Enter Network Password dialog box.


</td></tr>
<tr><td>3</td><td>Upon supplying the user name, password, and clicking the <OK> button, the Modify FTP Server User Name and Password Page displays.
NOTE: The default User Name is USER, and the default Password is USER. Both should be changed by the system administrator during module installation.</td></tr>
</table>

FTP Username and Password Modify Page Overview

The following figure shows the page used for modifying the FTP user name and password.



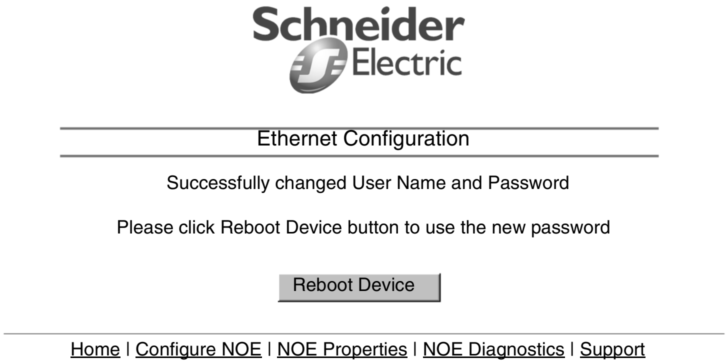
Change the Username and Password

At this point, the system administrator should change the Username and Password to restrict future access to the system. The following steps should be used.

Step	Action
1	Type in the new Username in the New User Name block
2	Type in the new Password in the New Password block
3	Click on the **Submit FTP Password Change** button.

Modify FTP Server User Name and Password Message

The following figure shows the message that is generated if you click on the Submit FTP Password Change button



Clicking the **Reboot Device** button will reset the Username and Password for the NOE 771 xx board.

> **Note:** The Reboot requires about 40 seconds (with large applications reboot may require up to one (1) minute).
> During the reboot all services provided by the NOE 771 xx are not available.

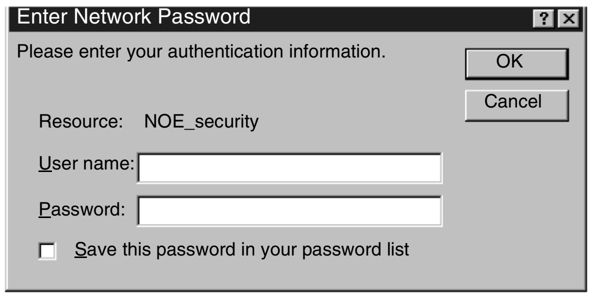
Establishing the HTTP Password

Overview

The following information describes how to set the HTTP Password for the NOE 771 0x only.

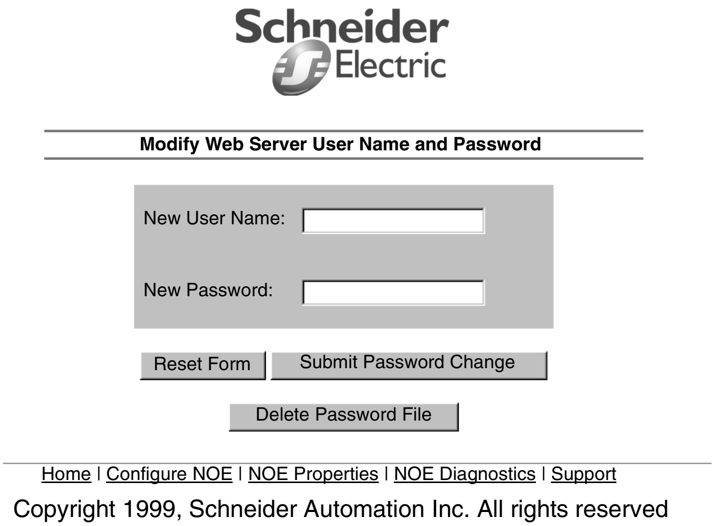
Modifying the HTTP Password

The following steps detail how to link to the correct Web page for modifying the HTTP Password

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Enter the URL, for example, http://hostname/secure/embedded/http_passwd_config.htm</td></tr>
<tr><td>2</td><td>When you enter the URL, you will be requested to supply a user name and password.
The following figure displays the Enter Network Password dialog box.

</td></tr>
<tr><td>3</td><td>Upon supplying the user name, password, and clicking the OK button, the Modify HTTP Server User Name and Password page displays.
NOTE: The default User Name is USER, and the default Password is USER. Both should be changed by the system administrator during module installation.</td></tr>
</table>

Modify Web Server User Name and Password Page Overview

The following figure shows the Schneider Automation Web page used to modify the HTTP user name and password.



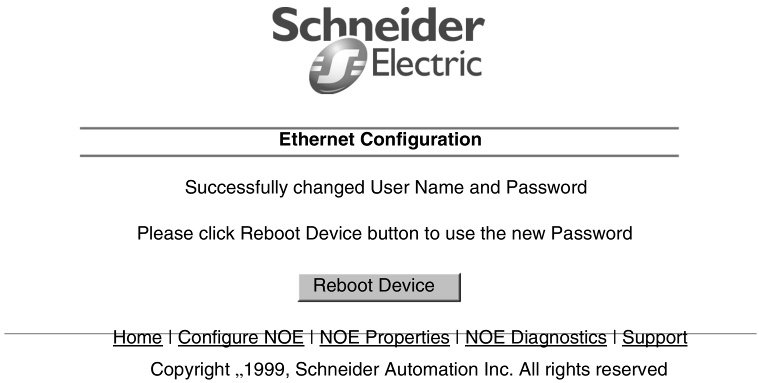
Change the Username and Password

At this point, the system administrator should change the Username and Password to restrict future access to the system. The following steps should be used.

Step	Action
1	Type in the new Username in the New User Name block
2	Type in the new Password in the New Password block
3	Click on the **Submit Password Change** button.

Modify Web Server User Name and Password Message

The following figure shows the message that is generated if you click on the **Submit Password Change** button.



Clicking the **Reboot Device** button will reset the Username and Password for the NOE 771 0x board.

> **Note:** The Reboot will take approximately 40 seconds (with large applications reboot may require one (1) minute).
> During the reboot all services provided by the NOE 771 0x are not available.

Establishing the SNMP Community Strings

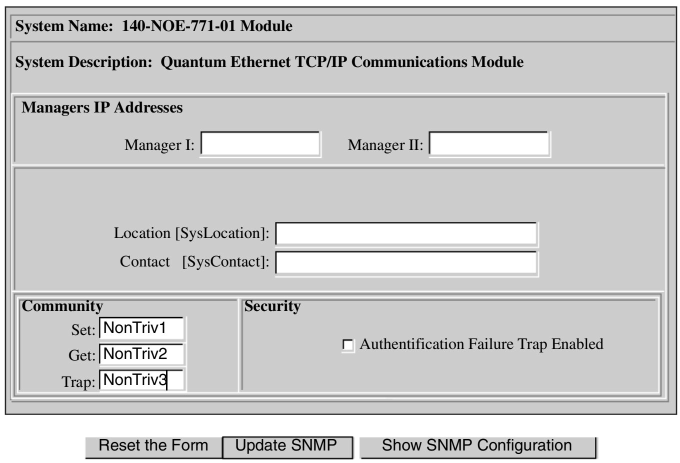
Overview

SNMP Community Strings are used to restrict access to the SNMP Agent. These strings should be set to Non-Trivial names during module installation.

Modifying the SNMP Community Strings

The following steps should be used to establish the SNMP Community Strings:

Step	Action
1	Enter the URL into your browser: http://hostname/secure/embedded/builtin?submit=Configure+SNMP or navigate to the **SNMP Configuration** Web page
2	Enter the **Community** names for **Set**, **Get**, and **Trap** into the **SNMP Configuration** Web page as shown below.
3	Click **Update SNMP**. The **SNMP Configuration** Web page.



Using "Bootp Lite" to Assign Address Parameters

Overview

The following information describes how to use the BOOTP Lite utility software.

	CAUTION
⚠	**UNINTENIONAL OPERATION — INCORRECT MAC ADDRESS** If you do not enter the correct parameters of the target controller, it will run in its old configuration. An incorrect MAC address may also result in an unwanted change to another device and cause unexpected results. ● Verify the MAC address of the target device before invoking Bootp Lite Server Software. **Failure to follow this precaution can result in injury or equipment damage.**

Bootp Lite Utility

Instead of a BOOTP server, Schneider Electric's *Bootp Lite Server Software* utility can be used to provide the IP address, subnet mask, and default gateway to the NOE 771 xx module.
Refer to the Bootp Lite Server Software user documentation for instructions

Note: Bootp Lite Server Software and user documentation are available for download at www.modicon.com.

Configuring the Module with Concept

3

At a Glance

Introduction

This chapter describes how to configure the NOE 771 module from your programming panel using Concept. This is used to configure the module's IP parameters using Concept. The module can function as a network interface to the CPU without I/O services, as long as the IP parameters are provided by a BOOTP server, or with the module's default IP address.

What's in this Chapter?

This chapter contains the following topics:

Selecting Your PLC

Overview

The following information describes how to start to configure the NOE 771 using Concept.

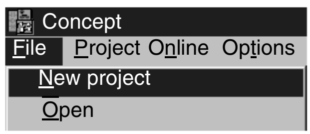
Initial Setup

After the NOE 771 module has been installed in a Quantum backplane (refer to the section *Installing the Module*), you can begin to configure it using Concept. To begin configuring the NOE 771, first select your CPU (PLC).

> **Note:** For detailed information about how to use Concept, refer to the set of manuals shipped with that software.

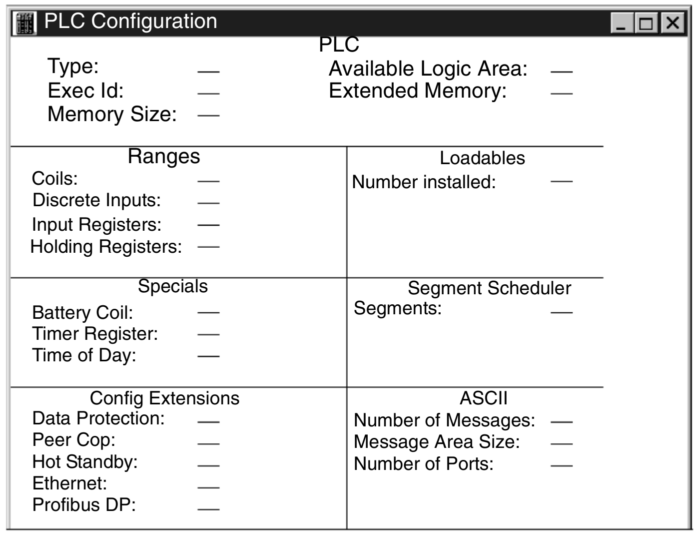
Procedure for Selecting a CPU

Perform the following steps to select a CPU..

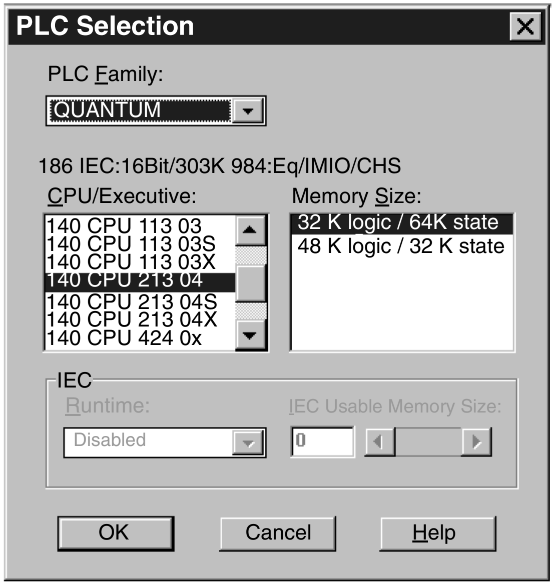
Step	Action
1	Open Concept on your programming panel (PC)
2	As shown in the following figure, from the File menu, select **New project**.  **Result**: As shown in the following figure, a new project is opened and the file name (untitled) appears over the menu bar. 

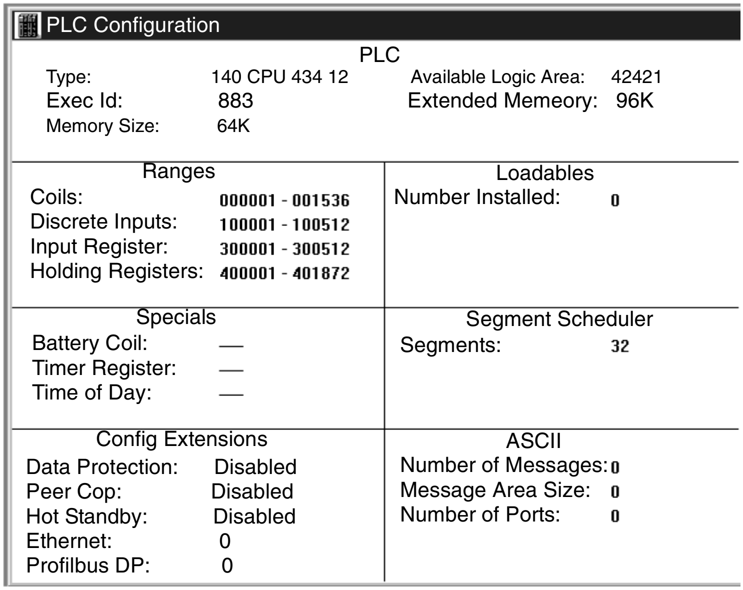
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>3</td><td>As shown in the following figure, from the Project menu, select Configurator.

Result: As shown in the following figure, the PLC Configuration screen displays.



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

Step	Action
4	Double click on the **Type** field in the PLC section of the Configure menu. **Result**: As shown in the following figure, the PLC Selection dialog box displays. The default selection is Quantum. 
5	From the **CPU/Executive** scroll box, select the CPU that is installed in your Quantum backplane NOTE: Depending on the CPU selected, you may need to select the correct memory size applicable to it in the **Memory Size** dialog box.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>6</td><td>Click the <OK> button.
Result: As shown in the following figure, your PLC type and default configuration parameters are displayed in the PLC Configuration screen.


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

Next

Next, you must configure the number of Ethernet modules that your system will contain, as shown in the *Cabling Schemes* procedure.

Setting the Number of NOEs

Overview

The following information describes how many NOEs you can configure in a single controller and how to configure that number.

Introduction

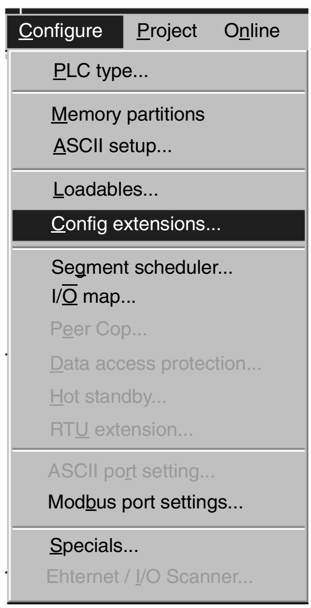
You may configure from two to six Ethernet modules in a single controller, depending on the model. A 140 CPU 113 or 213 will accept a total of two network option modules, including NOE, NOM, NOP, and CRP 811. A 140 CPU 424, 434, 534. 434A, or 534A will accept six. Refer to the table in *I/O Scanner Concepts* section regarding the mix of I/O scanners and NOE modules per CPU.

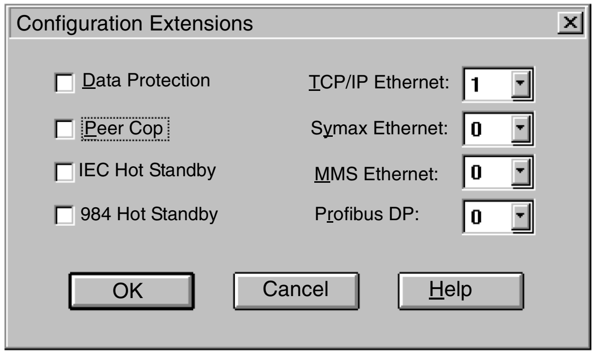
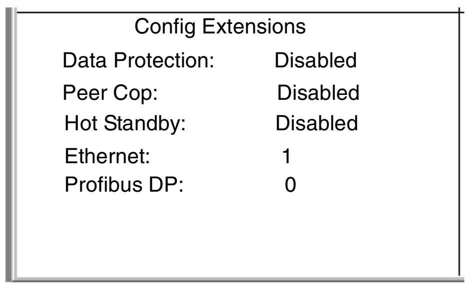
Memory Requirements

The first Ethernet TCP/IP module configured requires 20 words of memory. Each additional module requires an additional 16 words of memory.

Procedure for Setting the Number of NOEs

From the PLC Configuration screen, follow the steps below to select the number of NOE modules.

Step	Action
1	As shown in the following figure, from the Configure menu, select Config extensions, or double-click anywhere in the **Config Extensions** region of the screen.  **Result**: The Configuration Extension dialog box displays.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2</td><td>As shown in the following figure, in the TCP/IP Ethernet scroll box, select the number of NOE modules to be configured.

</td></tr>
<tr><td>3</td><td>Click on the <OK> butt
Result: As shown in the following figure, the Ethernet status changes from 0 to the number selected in Step 2.

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

Next

Next, you need to create an I/O map for the NOEs in your configuration, as shown in the *Accessing and Editing the I/O Map* section.

Accessing and Editing the I/O Map

Overview

The following information describes how to create an I/O map for the NOEs in your system.

Introduction

This procedure is required to determine the number of NOEs in the system and the slot numbers in which they reside.
As part of the configuration process, you need to create an I/O Map for the local backplane including the NOE 771 xx module. This step is required to determine the number of NOEs in the system and their slot assignments.

Procedure for Accessing and Editing an I/O Map

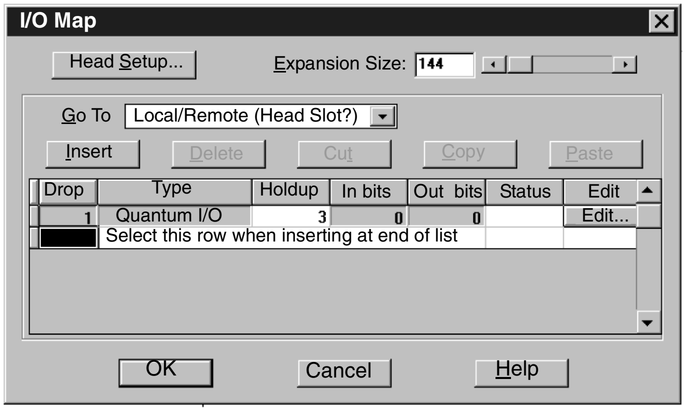
Perform the following steps to access and edit an I/O Map from the PLC. Configuration screen.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>As shown in the following figure, from the Configure menu, select I/O map.

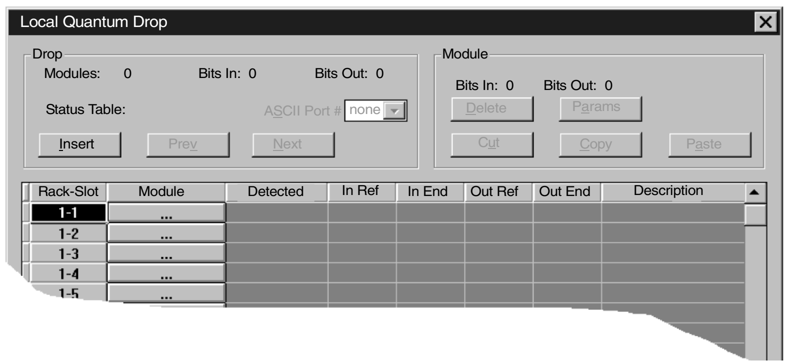
Result: As shown in the following figure, the I/O Map dialog box displays.</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2</td><td>

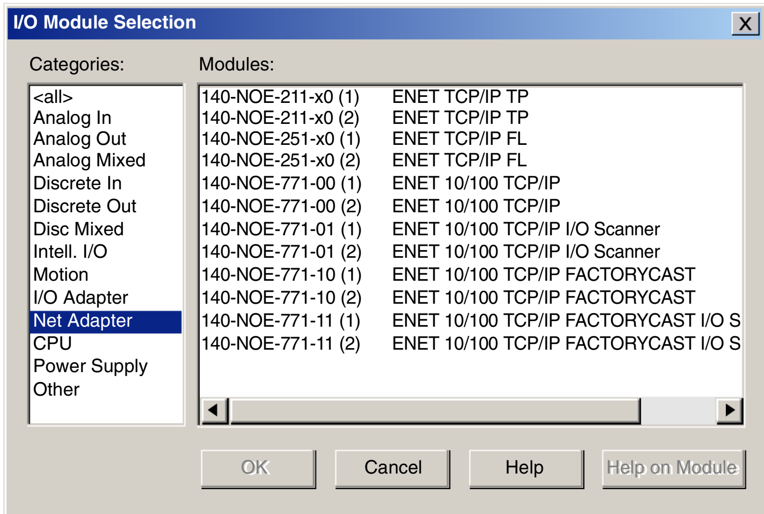
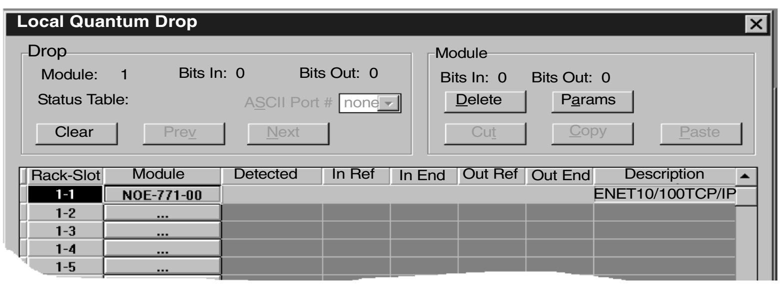
Click the <**Edit**> button at the end of the Quantum I/O row.



Result: As shown in the following figure, the Local Quantum Drop I/O dialog box displays.



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

Step	Action
3	Click on the **...** button under **Module**. **Result**: As shown in the following figure, the I/O Module Selection drop-down menu displays. 
4	Click on **NOE-771-00**, which appears in the Special column, and then click the <**OK**> button. **Result**: The Local Quantum Drop I/O dialog box updates and the **NOE-771-00** is now listed under Module and described in the Description column. 
5	Repeat Steps 3 and 4 if other modules need to be added to the I/O map.
6	Click the <**OK**> buttons to return to the PLC Configuration screen.

Next

Next, you will configure the Ethernet address parameters from the Ethernet/ I/O Scanner screen as shown in the *Configuring the Ethernet Address Parameters* section.

Configuring the Ethernet Address Parameters

Overview

The following information describes how to configure Ethernet address parameters for the NOE 771 xx.

Introduction

The NOE 771 xx module's Ethernet address parameters, consisting of Internet, Subnet mask, and Gateway addresses, are accessible from the Ethernet/ I/O Scanner dialog box. Prior to performing the following procedure, consult your system administrator to determine if you must configure new Ethernet address parameters, or whether the module will obtain them from the BOOTP server.

CAUTION

UNINTENTIONAL OPERATION — DUPLICATE IP ADDRESS

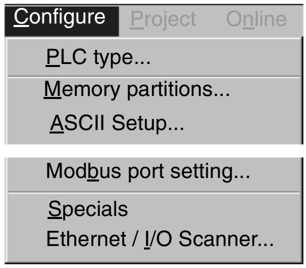
Having two devices with the same IP address can cause unpredictable operation of your network.

- Ensure that this device will receive a unique IP address.
- Always obtain your IP addresses from your system administrator to avoid the possibility of duplicate addresses.

Failure to follow this precaution can result in injury or equipment damage.

Procedure for Configuring Ethernet Address Parameters

Perform the following steps to configure the Ethernet Address Parameters..

Step	Action
1	As shown in the following figure, from the Configure menu, select **Ethernet/ I/O Scanner**  **Result**: The Ethernet/ I/O Scanner dialog box displays.
2	Click on the **Specify IP Address** radio button.
3	Type in the new IP, Subnet Mask, and Gateway addresses in the applicable text boxes.
4	Select the correct Internet frame type from the **Frame Type** scroll box.
5	If the module's BOOTP server will assign Ethernet address parameters, click on the Use BOOTP Server radio button Note that if you select this option, the address parameter text boxes will be grayed out and will not display the addresses.

How the Module Derives Its IP Address

During initialization, the NOE 771 module attempts to read the address parameter information from the PLC and determines its IP Address in the following fashion

- If the PLC has the IP Address and the BOOTP server is not selected, the module will use the configured IP address that you assigned in Step 2 of the above procedure.
- If the BOOTP server was selected in Step 5 of the above procedure, the module will send BOOTP requests to receive its IP Address.
- If no Configuration Extension exists, the NOE sends out BOOTP requests. If the module does not receive its IP Address from the BOOTP server after 2 minutes, it will then use the IP Address derived from its MAC address.

Note: The MAC address is assigned at the factory and is recorded on a label on the front panel, above the cable connector. This is a unique 48-bit global assigned address. It is set in PROM. The Ethernet address is recorded on the label in hexadecimal, in the form 00.00.54.xx.xx.xx.

Transferring Data using Communication Blocks

4

At a Glance

Introduction

This chapter describes how to transfer data to and from nodes on a TCP/IP network using communication blocks. You transfer the data using either a special MSTR instruction (the Master instruction of the 984 Ladder Logic instruction set) or an IEC Logic function. Included in this chapter are the operational statistics and error codes for reading and writing the controller information.

What's in this Chapter?

This chapter contains the following sections:

4.1 Using 984 Ladder Logic Communication Blocks

At a Glance

Introduction

This section contains information on the MSTR instruction of the 984 Ladder Logic instruction set.

What's in this Section?

This section contains the following topics:

MSTR Description

Overview

The following information describes MSTR operations.

Introduction

All NOE 771 x0 Quantum Ethernet TCP/IP modules provide the user with the capability of transferring data to and from nodes on a TCP/IP network through the use of a special MSTR (master instruction). All PLCs that support networking communication capabilities over Modbus Plus and Ethernet can use the MSTR ladder logic instruction to read or write controller information.

MSTR Operations

The following table lists each of the 12 possible MSTR network communications operations and indicates whether a TCP/IP Ethernet network supports it. Each operation is designated by a code. The following table lists the 12 operations and indicates those that are supported on an Ethernet TCP/IP network.

MSTR Operation	Operation Type	TCP/IP Ethernet Support
Write data	1	supported
Read Data	2	supported
Get local statistics	3	supported
Clear local statistics	4	supported
Write global database	5	not supported
Read global database	6	not supported
Get remote statistics	7	supported
Clear remote statistics	8	supported
Peer Cop health	9	supported
Reset Option Module	10	supported
Read CTE (config extension)	11	supported
Write CTE (config extension)	12	supported

No. of MSTR Instructions Allowed

Up to16 MSTR instructions can be simultaneously serviced in a ladder logic program per NOE. More than 16 MSTRs may be programmed to be enabled by the logic flow as one active MSTR block releases the resources it has been using and becomes deactivated, the next MSTR operation encountered in logic can be activated.

MSTR Block for TCP/IP in Concept

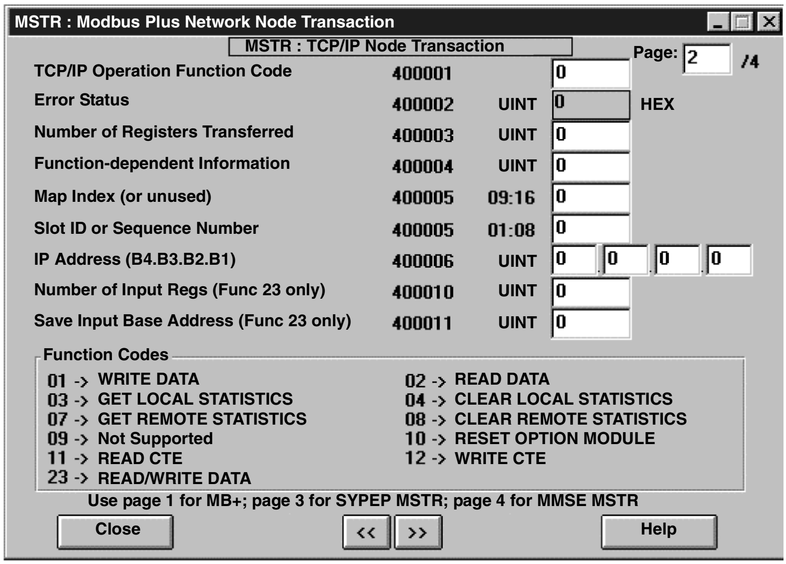
Overview

This following information describes how to complete installing the MSTR block.in a TCP/IP network.

Procedure for Installing the MSTR Block in TCP/IP

This is the MSTR Block as used for TCP/IP in Concept. After the MSTR Block is inserted in the network, perform the following steps.

Step	Action
1	Place cursor on the MSTR block.
2	Press <Ctrl D> (actually the dx zoom screen access)
3	Go to page 2 for TCP/IP specific MSTR block and complete the required information.



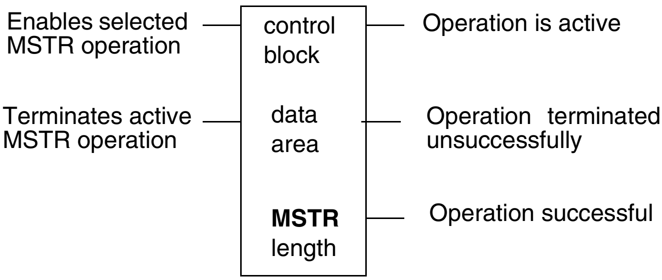
MSTR Ladder Logic Representation

Overview

The following information describes the Ladder Logic representation for MSTR.

Ladder Logic Diagram

As the following figure shows, the MSTR Block is represented in Ladder Logic diagrams.



Inputs

The MSTR instruction has the following two control inputs:
- the input to the top node enables the instruction if it is ON
- the input to the middle node terminates the active operation if it is ON

Outputs

The MSTR instruction can produce the following three possible outputs:
- the output from the top node echoes the state of the top input - it goes ON while the instruction is active
- the output from the middle node echoes the state of the middle input - it goes ON if the MSTR operation is terminated prior to completion or if an error occurs in completing the operation
- the output from the bottom node goes ON if an MSTR operation has been completed successfully

All outputs are zero is an indication that four MSTR instructions are already in progress.

Top Node Content

The 4x register entered in the top node is the first of several (network dependent) holding registers that comprise the network *control block*. The *control block* structure differs according to the network in use.
In the case of the Ethernet Read and Write CTE operations (see *Read and Write MSTR Operations, p. 78*), the middle node stores the contents of the Ethernet configuration extension table in a series of registers.
The following table shows the *control block* structure for the TCP/IP Ethernet network.

Register	Content
Displayed	Identifies one of ten MSTR operations legal for TCP/IP (1... 4 and 7... 12).
First implied	Displays error status.
Second implied	Displays length (number of registers transferred).
Third implied	Displays MSTR operation-dependent information.
Fourth implied	Low byte: Destination index.
	High byte: Quantum backplane slot address of the NOE module.
Fifth implied	Byte 4 of the 32-bit destination IP Address.
Sixth implied	Byte 3 of the 32-bit destination IP Address.
Seventh implied	Byte 2 of the 32-bit destination IP Address
Eight implied	Byte 1 of the 32-bit destination IP Address

Middle Node Content

The 4x register entered in the middle node is the first in a group of contiguous holding registers that comprise the *data area*. For operations that provide the communication processor with data such as a Write operation, the *data area* is the source of the data. For operations, such as Read, that acquire data from the communication processor, the *data area* is the destination for the data
In the case of the Ethernet Read and Write CTE operations (see *Read and Write MSTR Operations, p. 78*), the middle node stores the contents of the Ethernet configuration extension table in a series of registers.

Bottom Node Content

The integer value entered in the bottom node specifies the *length* - the maximum number of registers in the *data area*. The *length* must be in the range 1... 100.

MSTR Function Error Codes

Overview

The following information describes the error codes for MSTR operations.

Where Displayed

If an error occurs during an MSTR operation, a hexadecimal error code is displayed in the first implied register in the *control block* (the top node). Function error codes are network-specific.

TCP/IP Ethernet Error Codes

The following table describes the errors that can occur in the MSTR *control block* if the MSTR routine runs over TCP/IP Ethernet.

Hex Error Code	Meaning
1001	User has aborted the MSTR element.
2001	An unsupported operation type has been specified in the *control block*.
2002	One or more *control block* parameters has been changed while the MSTR element is active (applies only to operations that take multiple scans to complete). *Control block* parameters may be changed only when the MSTR element is not active.
2003	Invalid value in the length field of the *control block*.
2004	Invalid value in the offset field of the *control block*.
2005	Invalid values in the length and offset fields of the *control block*.
2006	Invalid slave device data area.
2008	Invalid slave device network routing
3000	Generic MODBUS fail code.
30ss*	MODBUS slave exception response.
4001	Inconsistent MODBUS slave response.
F001	Option Module not responding
* ss = subfield	

The following table lists the ss subfield values in error code 30ss.

ss Hex value	Meaning
01	Slave device does not support the requested operation.
02	Nonexistent slave device registers requested.
03	Invalid data value requested.
04	Reserved
05	Slave has accepted long-duration program command.
06	Function cannot be performed now; a long-duration command is in effect.
07	Slave rejected long-duration program command.

TCP/IP Ethernet Network Errors

The following table describes the errors that can occur in the MSTR *control block* as a result of an error on the TCP/IP Ethernet network.

Hex Error Code	Meaning
5004	Interrupted system call
5005	I/O error
5006	No such address
5009	The socket descriptor is invalid
500C	Not enough memory
500D	Permission denied
5011	Entry exists
5016	An argument is invalid
5017	An internal table has run out of space
5020	The connection is broken
5028	Destination address required
5029	Protocol wrong type for socket
502A	Protocol not available
502B	Protocol not supported
502C	Socket type not supported
502D	Operation not supported on a socket
502E	Protocol family not supported
502F	Address family not supported
5030	Address already in use
5031	Cannot assign requested address
5032	Socket operation on a non-socket
5033	Network is unreachable
5034	Network dropped connection on reset
5035	Network caused connection abort
5036	Connection reset by peer
5037	No buffer space available
5038	Socket is already connected
5039	Socket is not connected
503A	Cannot send after socket shutdown
503B	Too many references, cannot splice
503C	Connection timed-out (See note below.)
503D	Connection refused

503E	Network is down
503F	Text file busy
5040	Too many levels of links
5041	No route to host
5042	Block device required
5043	Host is down
5044	Operation now in progress
5045	Operation already in progress
5046	Operation would block
5047	Function not implemented
5050	No Network Resource
5051	Length Error
5052	Addressing Error
5053	Application Error
5054	Client in Bad State for Request
5055	No Remote Resource (Note: May indicate no path to Remote Device) (See note below.)
5056	Non-Operational TCP Connection
5057	Incoherent Configuration

Note: Error Codes 503C and 5055
- Error 5055 can occur before a 503C error.

No remote device takes precedence over a timeout.

CTE Error Codes

The following table lists the error codes that are returned if there is a problem with the Ethernet configuration extension table (CTE) in your program configuration.

Hex Error Code	Meaning
7001	There is no Ethernet configuration extension.
7002	The CTE is not available for access.
7003	The offset is invalid.
7004	The offset + length is invalid.
7005	Bad data field in the CTE.

Read and Write MSTR Operations

Overview

This section describes the MSTR Read and Write operations.

Introduction

An MSTR Write operation (operation type 1 in the displayed register of the top node) transfers data from a master source device to a specified slave destination device on the network. An MSTR Read operation (operation type 2 in the displayed register of the top node) transfers data from a specified slave source device to a master destination device on the network. Read and Write use one data master transaction path and may be completed over multiple scans.

> **Note:** TCP/IP Ethernet routing must be accomplished via standard third-party Ethernet IP router products

Control Block Utilization

The following table describes the registers in the MSTR *control block* (the top node) that contain the Read or Write information,

<table>
<tr><th>Register</th><th>Function</th><th colspan="2">Content</th></tr>
<tr><td>Displayed</td><td>Operation Type</td><td colspan="2">1 = Write, 2 = Read</td></tr>
<tr><td rowspan="4">First implied</td><td rowspan="4">Error status</td><td colspan="2">Displays a hex value indicating an MSTR error.</td></tr>
<tr><td>Exception response, where response size is incorrect.</td><td>Exception code +3000</td></tr>
<tr><td>Exception response where response size is incorrect.</td><td rowspan="2">4001</td></tr>
<tr><td>Read Write</td></tr>
<tr><td>Second implied</td><td>Length</td><td colspan="2">Write = number of registers to be sent to slave. Read = number of registers to be read from slave.</td></tr>
<tr><td>Third implied</td><td>Slave device data area</td><td colspan="2">Specifies starting 4x register in the slave to be read from or written to (1 = 4001, 49 =40049).</td></tr>
<tr><td>Fourth implied</td><td>Low byte</td><td colspan="2">Quantum backplane slot address of the NOE module.</td></tr>
<tr><td>Fifth ... Eighth implied</td><td>Destination</td><td colspan="2">Each register contains one byte of the 32-bit IP address.</td></tr>
</table>

Read/Write MSTR Operation (FC 23)

Overview

This section describes the MSTR Read/Write operation, Function Code (FC) 23.

Introduction

In a single transaction, the MSTR read and write operation transfers data from a master source device to a specified stave destination device and then transfers data from this specified slave source to the master.

Note: FC 23 can only be used with these two models:
- NOE 771 **01**, version 2.0 and higher
- NOE 771 **11**, version 2.0 and higher

Control Block Utilization

The following table describes the registers in the MSTR control block (the top node), registers that contain the Read or Write information.

Register	Function	Content
Displayed	Operation Type	23 = Read/Write
1st Implied	Error Status	Displays a hex value indicating a MSTR error
2nd Implied	Write Length	Number of registers to be sent to slave
3rd Implied	Slave device writes data area	Specifies starting 4x register in the slave to be written to (1 = 400001, 49 = 40049)
4th Implied	Slot ID	Slot where the NOE is located
5th Implied	IP - 34	Byte 4 (MSB) of the 32 bit destination IP address
6th Implied	IP - 33	Byte 3 of the 32 bit destination IP address
7th Implied	IP - 32	Byte 2 of the 32 bit destination IP address
8th Implied	IP - 31	Byte 1 (LSB) of the 32 bit destination IP address
9th Implied	Read Length	Number of registers to be read from slave
10th Implied	Slave device reads data area	Specifies starting 4x register in the slave to be read to

Get Local Statistics MSTR Operation

Overview

This section describes the Get Local Statistics operation.

Introduction

The Get Local Statistics operation (operation type 3 in the display register of the top node) obtains information related to the local node where the MSTR has been programmed. (See *TCP/IP Ethernet Statistics, p. 91*.)

Control Block Utilization

The following table describes the registers in the MSTR *control block* (the top node). These registers contain the Get Local Statistics information.

Register	Function	Content
Displayed	Operation Type	3
First implied	Error status	Displays a hex value indicating an MSTR error, if relevant.
Second implied	Length	Starting from *offset*, the number of words of statistics from the local processor's statistics table; the *length* must be > 0 < *data area*.
Third implied	Offset	An offset value relative to the first available word in the local processor's statistics table. If the offset is specified as 1, the function obtains statistics starting with the second word in the table.
Fourth implied	Low byte	Quantum backplane slot address of the NOE module.
Fifth ... Eighth implied	Not applicable	

Clear Local Statistics MSTR Operation

Overview

The following information describes the Local Statistics operation.

Introduction

The Clear Local Statistics operation (operation type 4 in the displayed register of the top node) clears statistics relative to the local node where the MSTR has been programmed.

Control Block Utilization

The following table describes the registers in the MSTR *control block* (the top node). These registers contain the Clear Local Statistics information.

Register	Function	Content
Displayed	Operation Type	4
First implied	Error status	Displays a hex value indicating an MSTR error, if relevant.
Second implied	Not applicable	
Third implied	Not applicable	
Fourth implied	Low byte	Quantum backplane slot address of the NOE module.
Fifth ... Eighth implied	Not applicable	

Get Remote Statistics MSTR Operation

Overview

The following information describes the Get Remote Statistics operation.

Introduction

The Get Remote Statistics operation (operation type 7 in the displayed register of the top node) obtains information relative to remote nodes on the network. This operation may require multiple scans to complete and does not require a master data transaction path. (See *TCP/IP Ethernet Statistics, p. 91*.)
The remote Ethernet module always returns its complete statistics table if a request is made, even if the request is for less than the full table. The MSTR instruction then copies only the amount of words you have requested to the designated 4x registers.

> **Note:** TCP/IP Ethernet routing must be accomplished via standard third-party Ethernet IP router products.

Control Block Utilization

The following table describes the registers in the MSTR *control block* (the top node). These registers contain the Get Remote Statistics information.

Register	Function	Content
Displayed	Operation Type	7
First implied	Error status	Displays a hex value indicating an MSTR error, if relevant.
Second implied	Length	Starting from an *offset*, the number of words of statistics from the local processor's statistics table; the length must be > 0 < *data area*.
Third implied	Offset	Specifies an offset value relative to the first available word in the local processor's statistics table. If the *offset* is specified as 1, the function obtains statistics starting with the second word in the table.
Fourth implied	High byte	Destination index
Fifth ... Eighth implied	Destination	Each register contains one byte of the 32-bit IP address.

Clear Remote Statistics MSTR Operation

Overview

The following information describes the Clear Remote Statistics operation.

Introduction

The Clear Remote Statistics operation (operation type 8 in the displayed register of the top node) clears statistics relative to a remote network node from the *data area* in the local node. This operation may require multiple scans to complete and uses a single data master transaction path.

Control Block Utilization

The following table describes the registers in the MSTR *control block* (the top node). These registers contain the Clear Remote Statistics information.

Register	Function	Content
Displayed	Operation Type	8
First implied	Error status	Displays a hex value indicating an MSTR error, if relevant.
Second implied	Not applicable	
Third implied	Not applicable	
Fourth implied	High byte	Destination index
Fifth ... Eighth implied	Destination	Each register contains one byte of the 32-bit IP address.

Peer Cop Health MSTR Operation

Overview

The following information describes the Peer Cop Health operation.

Introduction

The Peer Cop Health operation (operation type 9 in the displayed register of the top node) reads selected data from the Peer Cop communications health table and loads that data to specified 4x registers in state RAM. The Peer Cop communications health table is 12 words long, and the words are indexed via this MSTR operation as words 0 ... 11.

Note: The Peer Cop Health MSTR block is operational only if a Peer Cop based I/O Scanner has been configured.

Control Block Utilization

The following table describes the registers in the MSTR control block (the top node). These registers contain information for a Peer Cop Health operation.

Register	Function	Content
Displayed	Operation Type	9
First implied	Error status	Displays a hex value indicating an MSTR error, if relevant.
Second implied	Data Size	Number of words requested from Peer Cop table (range 1 ... 12).
Third implied	Index	First word from the table to be read (range 0 ... 11, where 0 = the first word in the Peer Cop table and 11 = the last word in the table).
Fourth implied	Low byte	Quantum backplane slot address of the NOE module.
Fifth ... Eighth implied	Destination	Each register contains one byte of the 32-bit IP address.

Peer Cop Communications Health Status Information

The following information describes the structure of the Peer Cop health table: Each bit in each table word is used to represent an aspect of communications health relative to a specific node on the TCP/IP network:

- The bits in words 0 ... 3 represent the health of the global input communication expected from nodes 1... 64. Since global input is not supported these bits are set to zero.
- The bits in words 4 ... 7 represent the health of the output from a specific node.
- The bits in words 8 ... 11 represent the health of the input to a specific node.

The following table shows the 12 contiguous registers used by the health table and the words to which they are indexed. Each row that is configured is mapped to a bit position.

Word 1 Bit Positions															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Word 2 Bit Positions															
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

Word 3 Bit Positions															
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48

Word 4 Bit Positions															
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64

Peer Cop Communications Health Bit State

The state of a Peer Cop health bit reflects the current communication status of its associated node. The following list provides more detailed information:

- A health bit is set if data is successfully exchanged with its corresponding node.
- A health bit is cleared if no communication has occurred with the corresponding node within the configured Peer Cop health time-out period.
- All health bits are cleared at PLC start time. The health bit for a given node is always zero if its associated Peer Cop entry is null.
- All global health bits are always reported as zero.

Reset Option Module MSTR Operation

Overview

The following information describes the Reset Option Module operation.

Introduction

The Reset Option Module operation (operation type 10 in the displayed register of the top node) causes a Quantum NOE option module to enter a reset cycle to reset its operational environment.

Control Block Utilization

The following table describes the registers in the MSTR *control block* (the top node). These registers contain the Reset Option Module information.

Register	Function	Content
Displayed	Operation Type	10
First implied	Error status	Displays a hex value indicating an MSTR error, if relevant.
Second implied	Not applicable	
Third implied	Not applicable	
Fourth implied	Low byte	Quantum backplane slot address of the NOE module.
Fifth ... Eighth implied	Not applicable	

Read CTE (Config Extension Table) MSTR Operation

Overview

The following information describes the Read CTE operation.

Introduction

The Read CTE operation (operation type 11 in the displayed register of the top node) reads a given number of bytes from the Ethernet configuration extension table to the indicated buffer in PLC memory. The bytes to be read begin at a byte offset from the beginning of the CTE. The content of the Ethernet CTE table is displayed in the middle node of the MSTR block.

Control Block Utilization

The following table describes the registers in the MSTR *control block* (the top node). These registers contain the Read CTE information.

Register	Function	Content
Displayed	Operation Type	11
First implied	Error status	Displays a hex value indicating an MSTR error, when relevant.
Second implied	Not applicable	
Third implied	Not applicable	
Fourth implied	Low byte	Quantum backplane slot address of the NOE module.
Fifth ... Eighth implied	Not applicable	

CTE Display Implementation

The values in the Ethernet configuration extension table (CTE) are displayed in a series of registers in the middle node of the MSTR instruction when a Read CTE operation is implemented. The middle node contains the first of 11 contiguous 4x registers.
The following table describes the CTE data contained in the registers.

Parameter	Register	Content	
Frame type	Displayed	1 = 802.3	
		2 = Ethernet	
IP Address	First implied	First byte of the IP address	
	Second implied	Second byte of the IP address	
	Third implied	Third byte of the IP address	
	Fourth implied	Fourth byte of the IP address	
Subnetwork mask	Fifth implied	Hi word	
	Sixth implied	Low word	
Gateway	Seventh implied	First byte of the gateway	
	Eighth implied	Second byte of the gateway	
	Ninth implied	Third byte of the gateway	
	Tenth implied	Fourth byte of the gateway	
	Eleventh implied	**High byte**	**Low byte**
		Software defined Module Type (Ignored by M1 and NOE modules) 0 = NOE211 1 = NOE251 2 = NOE77100 3 = NOE77110 4 = M1 5 = NOE77101 6 = NOE77111	IP Address Algorithm 0: Take IP Address from above definition (default) (All modules support this functionality) 1: Always take IP Address from BOOTP Server (M1 and NOE 771 x0 support this functionality) 2: Disable Ethernet functionality (M1 only)

Note: Module type only used by the p-unit during an upload to determine module.

Write CTE (Config Extension Table) MSTR Operation

Overview

The following information describes the Write CTE operation.

CTE Write Implementation

The Write CTE operation writes an indicated number of bytes from PLC memory, starting at a specified byte address, to an indicated Ethernet configuration extension table at a specified offset. The content of the Ethernet CTE table is contained in the middle node of the MSTR block.

Network Implementation

The Write CTE operation (type 12 in the displayed register of the top node) can be implemented for TCP/IP Ethernet networks via the appropriate network adapter.

Note: Modbus Plus networks do not use this operation.

Control Block Utilization

In a Write CTE operation, the registers in the MSTR *control block* (the top node) differ according to the network in user.
The following table describes the registers in the MSTR *control block* (the top node). These registers contain the Write CTE information.

Register	Function	Content
Displayed	Operation Type	12
First implied	Error status	Displays a hex value indicating an MSTR error, if relevant.
Second implied	Not applicable	
Third implied		
Fourth implied	Low byte	Either a value displayed in the high byte of the register or not used.
	Slot Index	Number displayed in the low byte, in a range 1 ... 16 indicating the slot in the local backplane where the option resides.
Fifth ... Eighth implied	Not applicable	

CTE Display Implementation

The values in the Ethernet configuration extension table (CTE) are displayed in a series of registers in the middle node of the MSTR instruction if a Write CTE operation is implemented. The middle node contains the first of 11 contiguous 4x registers.
The following table describes the CTE data contained in the registers.

<table>
<tr><th>Parameter</th><th>Register</th><th colspan="2">Content</th></tr>
<tr><td rowspan="2">Frame type</td><td rowspan="2">Displayed</td><td colspan="2">1 = 802.3</td></tr>
<tr><td colspan="2">2 = Ethernet</td></tr>
<tr><td rowspan="4">IP Address</td><td>First implied</td><td colspan="2">First byte of the IP address</td></tr>
<tr><td>Second implied</td><td colspan="2">Second byte of the IP address</td></tr>
<tr><td>Third implied</td><td colspan="2">Third byte of the IP address</td></tr>
<tr><td>Fourth implied</td><td colspan="2">Fourth byte of the IP address</td></tr>
<tr><td rowspan="2">Subnetwork mask</td><td>Fifth implied</td><td colspan="2">Hi word</td></tr>
<tr><td>Sixth implied</td><td colspan="2">Low word</td></tr>
<tr><td rowspan="4">Gateway</td><td>Seventh implied</td><td colspan="2">First byte of the gateway</td></tr>
<tr><td>Eighth implied</td><td colspan="2">Second byte of the gateway</td></tr>
<tr><td>Ninth implied</td><td colspan="2">Third byte of the gateway</td></tr>
<tr><td>Tenth implied</td><td colspan="2">Fourth byte of the gateway</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">Eleventh implied</td><td>High byte</td><td>Low byte</td></tr>
<tr><td>Software defined Module Type (Ignored by M1 and NOE modules)
0 = NOE211
1 = NOE251
2 = NOE771 00
3 = NOE771 10
4 = M1
5 = 140 NOE 771 01
6 = 140 NOE 771 11</td><td>IP Address Algorithm
0: Take IP Address from above definition (default) (All modules support this functionality)
1: Always take IP Address from BOOTP Server (M1 and NOE 771 x0 support this functionality)
2: Disable Ethernet functionality (M1 only)</td></tr>
</table>

TCP/IP Ethernet Statistics

Overview

The following information describes the available TCP/IP Ethernet Statistics.

Introduction

A TCP/IP Ethernet board responds to the "Get Local Statistics" and "Set Local Statistics" commands with the following information.

Word	Meaning
00 ... 02	MAC address
03	Board Status (see the Board Status Bit Definition table in this map)
04 and 05	Number of receiver interrupts
06 and 07	Number of transmitter interrupts
08 and 09	Transmit _ timeout error count
10 and 11	Collision_detect error count
12 and 13	Missed packets
14 and 15	Memory error
16 and 17	Number of times driver has restarted
18 and 19	Receive framing error
20 and 21	Receiver overflow error
22 and 23	Receive CRC error
24 and 25	Receive buffer error
26 and 27	Transmit buffer error
28 and 29	Transmit silo underflow
30 and 31	Late collision
32 and 33	Lost carrier
34 and 35	Number of retries
36 and 37	IP address

Board Status Word Bit Definition

The following table describes the word bit definitions for Board Status.

Bit #	Definition
15 ... 12	Module Type (See Module type table below)
11	(Reserved)
10	0 = half duplex 1 = full duplex
9	0 = not configured 1 = configured
8	0 = PLC not running 1 = PLC/NOE running
7	0 = Link LED off 1 = Link LED on
6	0 = Appl LED off 1 = Appl LED on
5	0 = twisted pair 1 = fiber
4	0 = 10 Mbit 1 = 100 Mbit
3 ... 0	(Reserved)

Board Status Word Bit Definition by Module Type

The following table describes the values of the Module Types.

Value of Bits 15 ... 12	Module Type
0	NOE 2x1
1	ENT
2	M1E
3	NOE 771 00
4	ETY
5	CIP
6	(Reserved)
7	(Reserved)
8	(Reserved)
9	(Reserved)
10	NOE 771 10
11	NOE 771 01
12	NOE 771 11
13 ... 15	(Reserved)

For bit level detail for the Momentum 170ENT1101 and Momentum 170ENT11000 see the *Momentum Ethernet Communication Adapter 170ENT11001 and 170ENT11000 User Guide*, 870USE11400.
For bit level detail for the 140NOE211xx see the *TCP/IP Module User Guide*, 840USE10700.

4.2 Using IEC Logic Communication Blocks

At a Glance

Introduction

This section contains information on the IEC Logic communication blocks used for transferring data.

What's in this Section?

This section contains the following topics:

CREAD_REG

Function Description

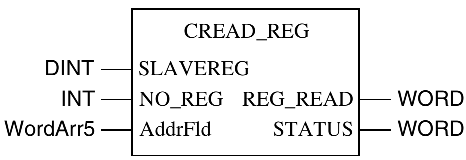
This Function block reads the register area continuously. It reads data from an addressed node via TCP/IP-Ethernet.
EN and ENO can be projected as additional parameters.

> **Note:** When programming a CREAD_REG function, you must be familiar with the routing procedures used by your network.

> **Note:** For technical reasons, this function block does not allow the use of programming languages ST and IL.

Representation

Block Representation



Parameter Description

Description of Parameters

Parameter	Data Type	Meaning
SLAVEREG	DINT	Offset address of the first 4x register in the slave to be read from
NO_REG	INT	Number of registers to be read from slave
AddrFld	WordArr5	Data structure describing the TCI/IP address
REG_READ	WORD	First 4x area register for read values
STATUS	WORD	Error code, see Runtime errors

Elementary Description for WordArr5 with TCP/IP Ethernet

Elementary description for WordArr5 with TCP/IP Ethernet

Element	Data Type	Meaning
WordArr5[1]	WORD	Low value byte: MBP on Ethernet Transporter (MET) mapping index High value byte: Slot of the NOE module
WordArr5[2]	WORD	Byte 4 (MSB) of the 32-bit destination IP address
WordArr5[3]	WORD	Byte 3 of the 32-bit destination IP address
WordArr5[4]	WORD	Byte 2 of the 32-bit destination IP address
WordArr5[5]	WORD	Byte 1 (LSB) of the 32-bit destination IP address

Function Mode of the CREAD_REG Block

Although a large number of CREAD_REG function blocks can be programmed, only four read operations may be active at the same time. In such a case it is insignificant whether they are the result of this function block or others (e.g. MBP_MSTR, MSTR, READ_REG). All function blocks use one data transaction path and require multiple cycles to complete a job.

Note: A TCP/IP communication between a Quantum PLC (NOE 771 xx) and a Momentum PLC (all TCP/IP CPUs and all TCP/IP I/O modules) is only possible, when only one read or write job is carried out in every cycle. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

Note: A TCP/IP communication between a Quantum PLC (NOE 211 00) and a Momentum PLC (all TCP/IP CPUs and all TCP/IP I/O modules) is only possible, when only oneread or write job is carried out in every cycle.. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

The entire routing information is contained in data structure WordArr5 of input AddrFld. The type of function block connected to this input and thus the contents of the data structure depends on the network used.
Please use:
- TCP/IP EtherNet the function block TCP_IP_ADDR

Note: For experts: The WordArr5 data structure can be used with constants as well.

> **Note:** This function block puts a heavy load on the network; therefore the network load must be carefully monitored. If the network load is too high, the program logic should be reorganized in order to work with the READ_REG function block, a variation of this function block that does not operate in a continuous mode, but under command control.

SLAVEREG

Start of the area in the addressed slave from which the source data is read. The source area always resides within the 4x register area. SLAVEREG expects the source reference as offset within the 4x area. The leading "4" must be omitted (e.g. 59 (contents of the variables or value of the literal) = 40059).
The parameter can be specified as direct address, located variable, unlocated variable, or literal.

NO_REG

Number of registers to be read from the addressed slave (1 ... 100). The parameter can be entered as a Direct address, Located variable, Unlocated variable, or Literal. The parameter can be entered as a Direct address, Located variable, or Unlocated variable.

REG_READ

This word parameter addresses the first register in a series of NO_REG registers, listed one after the other, which are used as a destination data area. The parameter must be entered as a Direct address or located Variable.

STATUS

Error code, see Runtime errors.

CWRITE_REG

Function Description

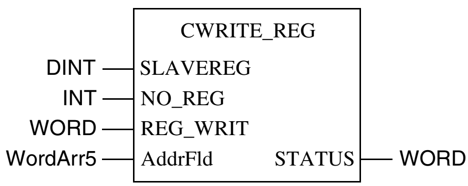
The purpose of this Function block is to write the register area continuously. It transfers data from the PLC via TCP/IP Ethernet to an addressed slave.
EN and ENO can be configured as additional parameters.

Note: You must be familiar with the routing procedures of the network when programming a CWRITE_REG function.

Note: For technical reasons, this function block does not allow the use of ST and IL programming languages.

Symbol

Block Representation



Parameter Description

Description of Parameters

Parameter	Data Type	Meaning
SLAVEREG	DINT	Offset address of the first 4x register in the slave to be written to
NO_REG	INT	Number of registers to be written to slave
REG_WRIT	WORD	First 4x register of the source data area
AddrFld	WordArr5	Data structure for transferring the TCI/IP address
STATUS	WORD	MSTR error code, see Runtime errors

Elementary Description for WordArr5 with TCP/IP Ethernet

Elementary description for WordArr5 with TCP/IP Ethernet

Element	Data Type	Meaning
WordArr5[1]	WORD	Low value byte: MBP on Ethernet Transporter (MET) mapping index High value byte: Slots of the NOE module
WordArr5[2]	WORD	Byte 4 (MSB) of the 32-bit destination IP address
WordArr5[3]	WORD	Byte 3 of the 32-bit destination IP address
WordArr5[4]	WORD	Byte 2 of the 32-bit destination IP address
WordArr5[5]	WORD	Byte 1 (LSB) of the 32-bit destination IP address

CWRITE_REG Block Function Mode

Although a large number of CWRITE_REG function blocks can be programmed, only four write operations may be active at the same time. It makes no difference whether these operations are performed using this function block or others (e.g. MBP_MSTR, MSTR, WRITE_REG). All function blocks use one data transaction path and require multiple cycles to complete a job.
If several CWRITE_REG function blocks are used within an application, they must at least differ in the values of their NO_REG or REG_WRITE parameters.

Note: A TCP/IP communication between a Quantum PLC (NOE 771 xx) and a Momentum PLC (all TCP/IP CPUs and all TCP/IP I/O modules) is only possible, when only one read or write job is carried out in every cycle. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

The entire routing information is contained in data structure WordArr5 of input AddrFld. The type of function block connected to this input and thus the contents of the data structure depend on the network used.
Please use:
- TCP/IP Ethernet: the function block TCP_IP_ADDR

Note: For experts: The WordArr5 data structure can also be used with constants.

Note: This function block puts a heavy load on the network. The network load must therefore be carefully monitored. If the network load is too high, the program logic should be reorganized to work with the WRITE_REG function block, which is a variant of this function block that does not operate in continuous mode but is command driven.

SLAVEREG Start of the area in the addressed slave to which the source data are written. The destination area always resides within the 4x register area. SLAVEREG expects the destination address as offset within the 4x area. The initial "4" must be omitted (e.g. 59 (contents of the variables or value of the literal) = 40059).
The parameter can be specified as direct address, located variable, unlocated variable, or Literal.

NO_REG Number of registers to be written to slave processor (1 ... 100). The parameter can be specified as direct address, located variable, unlocated variable, or Literal.

STATUS Error code, see Runtime errors.
The parameter can be specified as direct address, located variable or unlocated variable.

REG_WRIT This word parameter addresses the first register in a series of NO_REG Successive registers used as source data area.
The parameter must be entered as a direct address or located variable.

READ_REG

Function Description

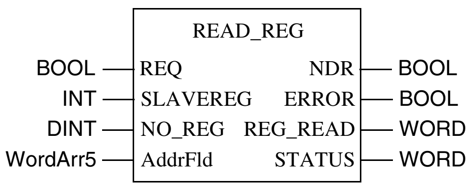
If requested, this function block will read a register area once (rising edge of the REQ input). It reads data from an addressed slave via TCP/IP-Ethernet.
EN and ENO can be projected as additional parameters.

Note: You must be familiar with the routing procedures of your network when programming a READ_REG function.

Note: Note: For technical reasons, this function block does not allow use of the programming languages ST and IL.

Symbol

Block Representation



Parameter Description

Description of Block Parameters

Parameter	Data Type	Meaning
REQ	BOOL	Start read operation once
SLAVEREG	DINT	Offset address of the first 4x register in the slave to be read from
NO_REG	INT	Number of registers to be read from slave
AddrFld	WordArr5	Data structure describing the TCP/IP address
NDR	BOOL	Set to "1" for one cycle after reading new data
ERROR	BOOL	Set to "1" for one scan in case of error
STATUS	WORD	Error code, see Runtime errors
REG_READ	WORD	First 4x area register for read values

Elementary Description for WordArr5 with TCP/IP Ethernet

Elementary description for WordArr5 with TCP/IP Ethernet

Element	Data Type	Meaning
WordArr5[1]	WORD	Low value byte: MBP on Ethernet Transporter (MET) mapping index High value byte: Slot of the NOE module
WordArr5[2]	WORD	Byte 4 (MSB) of the 32-bit destination IP address
WordArr5[3]	WORD	Byte 3 of the 32-bit destination IP address
WordArr5[4]	WORD	Byte 2 of the 32-bit destination IP address
WordArr5[5]	WORD	Byte 1 (LSB) of the 32-bit destination IP address

Function Mode of READ_REG Blocks

Although a large number of READ_REG function blocks can be programmed, only four read operations may be active at the same time. In such a case it is insignificant whether they are the result of this function block or of other read operations (e.g. MBP_MSTR, MSTR, CREAD_REG). All function blocks use one data transaction path and require multiple cycles to complete a job.

Note: A TCP/IP communication between a Quantum PLC (NOE 771 xx) and a Momentum PLC (all TCP/IP CPUs and all TCP/IP I/O modules) is only possible, when only one read or write job is carried out in every cycle. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

The entire routing information is contained in data structure WordArr5 of input AddrFld. The type of function block connected to this input and thus the contents of the data structure depends on the network used.
Please use:
- TCP/IP EtherNet the function block TCP_IP_ADDR

Note: For experts: The WordArr5 data structure can be used with constants as well.

REQ

A rising edge triggers the read transaction.
The parameter can be specified as direct address, located variable, unlocated variable, or Literal.

SLAVEREG

Start of the area in the addressed slave from which the source data is read. The source area always resides within the 4x register area. SLAVEREG expects the source reference as offset within the 4x area. The leading "4" must be omitted (e.g. 59 (contents of the variables or value of the literal) = 40059).
The parameter can be specified as direct address, located variable, unlocated variable, or Literal.

NO_REG

Number of registers to be read from the addressed slave (1 ... 100).
The parameter can be specified as direct address, located variable, unlocated variable, or Literal.

NDR

Transition to ON state for one program cycle signifies receipt of new data ready to be processed.
The parameter can be specified as direct address, located variable, or unlocated variable.

ERROR

Transition to ON state for one program cycle signifies detection of a new error.
The parameter can be specified as direct address, located variable, or unlocated variable.

REG_READ

This word parameter addresses the first register in a series of NO_REG registers lying in series used as destination data area.
The parameter must be entered as a direct address or located variable.

STATUS

Error code, see Runtime errors.
The parameter can be specified as direct address, located variable or unlocated variable.

WRITE_REG

Function Description

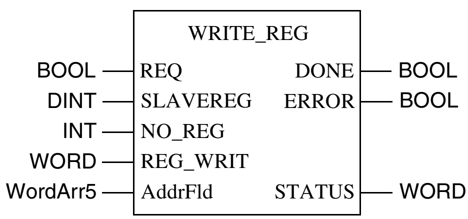
If requested, this Function block will write a register area once (rising edge of the REQ input). It transfers data from the PLC via TCP/IP Ethernet to an addressed slave.
EN and ENO can be configured as additional parameters.

Note: You must be familiar with the routing procedures of your network when programming a WRITE_REG function.

Note: For technical reasons, this function block does not allow the use of ST and IL programming languages.

Symbol

Block Representation



Parameter Description

Description of Parameters

Parameter	Data Type	Meaning
REQ	BOOL	Start write operation once
SLAVEREG	DINT	Offset address of the first 4x register in the slave to be written to
NO_REG	INT	Number of registers to be written from slave
AddrFld	WordArr5	Data structure transferring the TCP/IP address
REG_WRIT	WORD	First 4x register of the source data area
DONE	BOOL	Set to "1" for one scan after writing data
ERROR	BOOL	Set to "1" for one scan in case of error
STATUS	WORD	Error code, see Runtime errors

Elementary Description for WordArr5 with TCP/IP Ethernet

Elementary description for WordArr5 with TCP/IP Ethernet

Element	Data Type	Meaning
WordArr5[1]	WORD	High value byte: Slot of the NOE module Low value byte: MBP on Ethernet Transporter (MET) mapping index
WordArr5[2]	WORD	Byte 4 (MSB) of the 32-bit destination IP address
WordArr5[3]	WORD	Byte 3 of the 32-bit destination IP address
WordArr5[4]	WORD	Byte 2 of the 32-bit destination IP address
WordArr5[5]	WORD	Byte 1 (LSB) of the 32-bit destination IP address

Function Mode of the WRITE_REG Module

Although a large number of WRITE_REG function blocks can be programmed, only four write operations may be active at the same time. In such a case it is insignificant whether they are the result of this function block or of other write operations (e.g. MBP_MSTR, MSTR, CWRITE_REG). All function blocks use one data transaction path and require multiple cycles to complete a job.
If several WRITE_REG function blocks are used within an application, they must at least differ in the values of their NO_REG or REG_WRITE parameters.

> **Note:** A TCP/IP communication between a Quantum PLC (NOE 771 xx) and a Momentum PLC (all TCP/IP CPUs and all TCP/IP I/O modules) is only possible, when only one read or write job is carried out in every cycle. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

The status signals DONE and ERROR report the function block state to the user program.
The entire routing information is contained in data structure WordArr5 of input AddrFld. The type of function block connected to this input and thus the contents of the data structure depend on the network used.
Please use:
- TCP/IP Ethernet: the function block TCP_IP_ADDR

> **Note:** For experts: The WordArr5 data structure can also be used with constants.

REQ

A rising edge triggers the read transaction.
The parameter can be specified as direct address, located variable, unlocated variable, or Literal.

SLAVEREG Start of the area in the addressed slave from which the source data is read. The source area always resides within the 4x register area. SLAVEREG expects the source reference as offset within the 4x area. The leading "4" must be omitted (e.g. 59 (contents of the variables or value of the literal) = 40059).
The parameter can be specified as direct address, located variable, unlocated variable, or Literal.

NO_REG Number of registers to be read from the addressed slave (1 ... 100).
The parameter can be specified as direct address, located variable, unlocated variable, or Literal.

REG_WRIT This word parameter addresses the first register in a series of NO_REG registers used as source data area.
The parameter must be entered as a direct address or located variable.

DONE Transition to ON state for one program scan signifies data have been transferred.
The parameter can be specified as Direct address, Located variable or Unlocated variable.

ERROR Transition to ON state for one program scan signifies detection of a new error.
The parameter can be specified as Direct address, Located variable or Unlocated variable.

STATUS Error code, see Runtime errors.
The parameter can be specified as direct address, located variable, or unlocated variable.

TCP_IP_ADDR

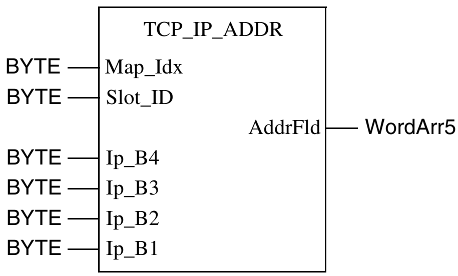
Function Description

This Function Block enables the input of TCP/IP addresses for the READ_REG, CREAD_REG, WRITE_REG and CWRITE_REG Function Blocks. The address is transferred in the form of a data structure.
The parameters EN and ENO can additionally be projected.

Note: You must be familiar with your network when programming the TCP_IP_ADDR Function Block.

Symbol

Block representation



Parameter Description

Description of Parameters

Parameter	Data Type	Meaning
Map_Idx	BYTE	Map-Index MBP on Ethernet Transporter (MET) mapping index
Slot_ID	BYTE	Slot ID Slot of the NOE module
Ip_B4	BYTE	Byte 4 (MSB) of the 32-bit destination IP address
Ip_B3	BYTE	Byte 3 of the 32-bit destination IP address
Ip_B2	BYTE	Byte 2 of the 32-bit destination IP address
Ip_B1	BYTE	Byte 1 (LSB) of the 32-bit destination IP address
AddrFld	WordArr5	Data structure used to transfer the TCP/IP address

Elementary Description for WordArr5

Elementary description for WordArr5

Element	Data Type	Meaning
WordArr5[1]	WORD	High value byte: Slot of the NOE module Low value byte: MBP on Ethernet Transporter (MET) mapping index
WordArr5[2]	WORD	Byte 4 of the 32-bit destination IP address
WordArr5[3]	WORD	Byte 3 of the 32-bit destination IP address
WordArr5[4]	WORD	Byte 2 of the 32-bit destination IP address
WordArr5[5]	WORD	Byte 1 of the 32-bit destination IP address

Map_Idx

The MBP on Ethernet Transporter (MET) mapping index is given at the Map_Idx input, i.e. if MET is 6, the value appears as follows:

0	0	0	0	0	1	1	0

Slot_ID

If an NOE in the rack of a Quantum controller is addressed as destination node, the value at the Slot_ID input represents the physical NOE slot, i.e. if the NOE is plugged in at Slot 7 of the rack, the value appears as follows:

0	0	0	0	0	1	1	1

AddrFld

If an NOE in the rack of a Quantum controller is addressed as a destination node, the value in the High value byte represents the physical slot of the NOE and the Low value byte represents the MBP on Ethernet Transporter (MET) mapping index, i.e. if the NOE is inserted in slot 7 of the rack and the MET mapping index is 6, the first element of the data structure looks as follows:

High value byte								Low value byte							
0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0

High value byte Slots 1 ... 16
Low value byte MBP on Ethernet Transporter (MET) mapping index

MBP_MSTR

Function Block

With this Function block, it is possible to select one of 12 available network communication operations.
Although a large number of MBP_MSTR function blocks can be programmed, only four of them can be active at the same time. All function blocks use one data transaction path and require multiple cycles to complete a job.

> **Note:** A TCP/IP communication between a Quantum PLC (NOE 771 xx) and a Momentum PLC (all TCP/IP CPUs and all TCP/IP I/O modules) is only possible, when only one read or write job is carried out in every cycle. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

> **Note:** In FBD and LD sections, the function block can only be used on the program level, i.e. not in Derived Function Blocks (DFBs).

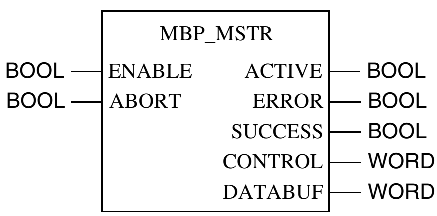
> **Note:** For technical reasons, this function block does not allow the use of programming languages ST and IL.

EN and ENO can be configured as additional parameters.

> **Note:** You must be familiar with the routing procedures of your network when programming an MSTR function.

Symbol

Block representation



Parameter Description

Description of parameters

Parameter	Data Type	Meaning
ENABLE	BOOL	Enable MSTR function
ABORT	BOOL	Cancel active MSTR operation
ACTIVE	BOOL	Operation is active
ERROR	BOOL	Faulty operation
SUCCESS	BOOL	Operation completed successfully
CONTROL	WORD	First 4x register of the MSTR control block
DATABUF	WORD	First 4x register of the data field

Function Mode of MBP_MSTR Blocks

Using the MBP_MSTR block, one of 12 available network communication operations can be triggered via the network. Each operation receives a code. Whether the operations are available depends on the type of network used.

Valid Function Codes

Valid function codes:

Code	Function	TCP/IP Ethernet
1	Write Data	X
2	Read Data	X
3	Get Local Statistics	X
4	Clear Local Statistics	X
5	Write Global Data	-
6	Read Global Data	-
7	Get Remote Statistics	X
8	Clear Remote Statistics	X
9	Peer Cop Status (Peer Cop Health)	-
10	Reset optional module	X
11	Read CTE (Config extension)	X
12	Read CTE (Config extension	X

Legend:

X	Yes
-	No

ENABLE

When ON, the operation specified in the first CONTROL register is enabled.

ABORT

When ON, the currently active operation is aborted.

ACTIVE

ON, if the operation is active.

ERROR

ON, if the operation was aborted without success.

SUCCESS

ON, if the operation concluded successfully.

DATABUF

The 4x register specified is the first in a group of successive output/marker words, making up the data field. For operations providing data, e.g. the write operation, the data field is the data source. For operations receiving data, e.g. the read operation, the data field is the data sink.
In the case of Ethernet CTE Read and Write operations, the middle input stores the contents of the Ethernet configuration extension table in a series of registers.

CONTROL

This word parameter addresses the first of several successive 4x registers. The control block is contained in these registers. The first register displayed contains a number from 1 to 12, which provides the operation code of the MODBUS operation to be performed. The contents of the sequence registers are determined by the operation.
The structure of the control block differs according to the network used:
- TCP/IP Ethernet

Control Block for TCP/IP Ethernet

Control block for TCP/IP Ethernet

Register	Contents
4x	Indicates one of the Operations which are valid for TCP/IP
4x + 1	Indicates the Error status
4x + 2	Indicates the length (number of registers transferred)
4x + 3	Indicates MSTR operation-dependent information
4x + 4	Routing register Low value byte: MBP on Ethernet Transporter (MET) mapping index High value byte: Slot of the NOE module
4x + 5	Byte 4 (MSB) of the 32-bit destination IP address
4x + 6	Byte 3 of the 32-bit destination IP address
4x + 7	Byte 2 of the 32-bit destination IP address
4x + 8	Byte 1 (LSB) of the 32-bit destination IP address

Routing Register (4x + 4) in TCP/IP Ethernet

If a NOE in the rack of a Quantum controller is addressed as destination node, the value in the high value byte represents the physical NOE slot and the value in the low value byte represents the MBP on Ethernet (MET) mapping index, i.e. if the NOE is plugged in at Slot 7 of the rack and the MET mapping index is 6, the first element of the data structure appears as follows:

High value byte								Low value byte							
0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0

High value byte Slots 1 to 16
Low value byte MBP on Ethernet Transporter (MET) mapping index

Transferring Data with the Global Data (Publish / Subscribe) Utility NOEs -01, -11

5

At a Glance

Introduction

The material in this section presents the Global Data (Publish / Subscribe) utility available on two of the 140 NOE 771 modules.
Those two modules are:
- 140 NOE 771 01
- 140 NOE 771 11

For more information on the publish-subscribe model, go to this URL:
`http://www.isa.org/journals/intech/feature/printable/1,1171,596,00.html`

What's in this Chapter?

This chapter contains the following topics:

Planning the Global Data (Publish / Subscribe) System

Overview

Global Data service is a real time Publisher/Subscriber mechanism providing the most efficient data exchange for PLC application coordination.
Devices supporting Global Data are grouped in a distribution group for the purpose of application variable exchange and synchronization. Each Global Data device can Publish up to one network (application) variable and Subscribe up to 64 Network (application) variables.
The Quantum NOE's embedded Web Global Data Configuration page provides a configuration screen to determine which and how many application variables are exchanged with this service. After configuration, the exchanges between all stations belonging to the same distribution group are done automatically.
The Global Data Service uses the 4x Register Space for Global Data Exchanges.

Key Features of Global Data

The main features for Global Data are:
- One Publisher and many Subscribers
- A Device can publish one Network Variable of up to 512 Registers
- A Device can subscribe to several Network Variables of up to 2048 4x Registers
- A Device subscribes to the complete Network Variable
- One distribution group per network IP address
- Application defined Publication rate
- Up to 64 Global Data Network Variables (numbered from 1 to 64) can be part of the data distribution group
- An NOE has only one multicast address; consequently, it can only Publish and Subscribe inside the group
- A Device can participate in several Distribution Groups by using multiple NOEs in the Rack

Global Data has an advantage over Client / Server services when more than one Subscriber is receiving the same data since only one transaction is necessary for all Subscribers to receive the data.
This advantage offers two benefits:
- Reduce overall network traffic
- Ensure tighter synchronization of multiple subscribers

Planning Your System Configuration

The Global Data (Publish / Subscribe) utility is a powerful function incorporated into the NOE product line. Implementing Global Data requires a configuration that spans many PLCs throughout the system. Therefore, we recommend preplanning your installation before implementation. Work spent on preplanning saves time and money by reducing errors and unnecessary debugging time. Preplanning also serves as an aid to ensuring consistency throughout the system.

Go to paper before computer.

We offer the following table to help with your system planning. The table below is a graphic representation of a recommended configuration table for system planning, which we call the **Global Data Planning Spreadsheet**. You may create your own table using the format below or you may download a Microsoft *Excel*™ spreadsheet template which is available on the Schneider public Web site.

Here is the graphic representation of the **Global Data Planning Spreadsheet**.

Parameter Checking	**Variable ID**	**Symbol** [1.]	**Size (Registers)**	**Devices 1 through 64**	
				Device	Device
	Variable ID 1				
	Variable ID 2 ... 63				
	Variable ID 64				
	Device Publication Status				
	Total Publication Size per Node				
	Total Subscription Size per Node				
Group IP Address		239.255.255.0			
Multicast Filtering Enabled		OFF			
Default 4x Address for Health		400100			
Distribution Period		5			
Health Timeout		1000			
Default 4x Address for Global		400100			

Note: The following note relates to the *Symbol* (description).

1. Entries or changes to the symbol (description) do NOT affect or change a variable or the system. The Symbol used in the *Quantum* product line has no relation to the *Concept* product line symbol.

Table of **Global Data Limits**

Parameter	**Limit**
Number of devices allowed in a distribution group	64 Maximum
Maximum size for the Publish Variable	512 Registers = 512 Words (16 bits) = 1024 Bytes (or Octets)
Maximum total Subscription size / device	2k Registers
Maximum number of Subscriptions / device	64 (63 if this device is publishing)
Maximum number of Publications / device	1
Publishers / Variable	1

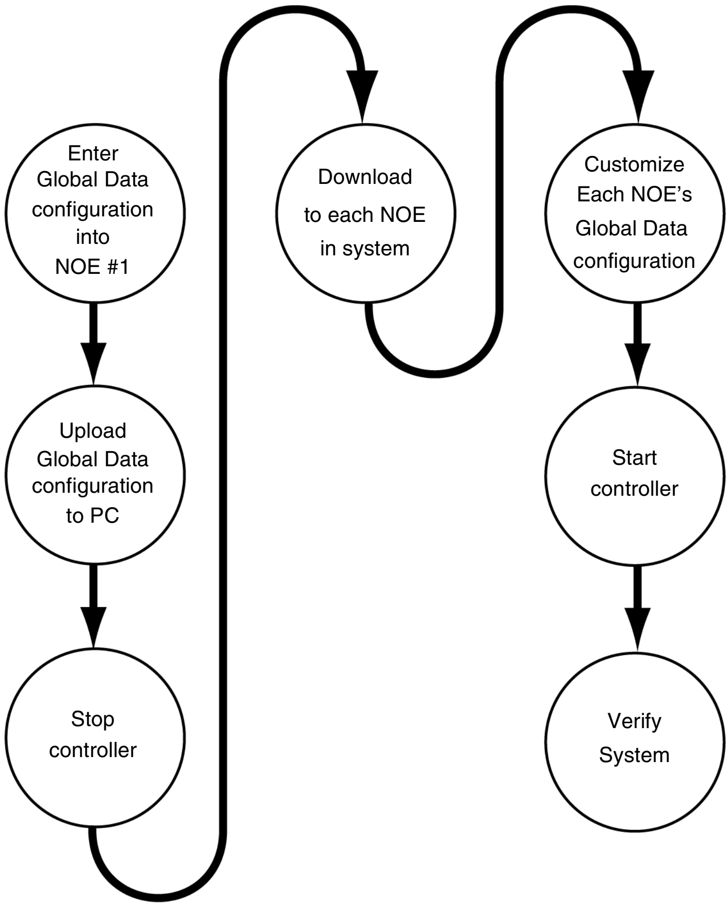
Note: We recommend that you consider the following when planning.
- 10 to 20% Increase margin for growth
  We suggest that you allow for a percentage increase in growth of any variable, a 10 to 20% increase allowance should be sufficient.
- Add at End
  We recommend that you add variables at the end of the configuration because variables added at the end of the configuration do not affect the existing application address. Therefore, you avoid changing the existing addresses in your configuration, which can be a time consuming process.

Modeling the System Configuration

There are two methods of configuring a system.
- Configure Each Device Separately
  Using this method you configure devices via the Global Data Configuration Web page for each device. Repeat for each device in the system We suggest that you do not use this method because entry errors may occur due to the fact that the same information must be reentered on each and every device, possibly as much as 64 times.
- Copy Configuration
  You configure all variable IDs, symbols (descriptions), and lengths on one NOE, then upload to your PC, and download the same configuration to all devices, and finish with a small customization to each node.

Note: We recommend that you use the Copy Configuration method to configure all variable IDs, symbols (description), and lengths.
Choosing this method reduces entry errors, allows for verifying the configuration, and permits you to determine if the system is satisfactory for your needs before implementing the configuration system wide.
By using this method you enter the variable ID, symbol (description), and length only once thus ensuring consistency.

When your planning is complete, follow the *Configuring Your NOE* steps below. We present these steps both as a picture and as instructions.
These are the *Configuring Your NOE* steps displayed as a picture



These are the *Configuring Your NOE* steps written as instructions.

Step	Action
1	Select one NOE.
2	Using your browser, navigate to that NOE's **Global Data Configuration** page. Follow these links:l **Web Server** l **Configure NOE** l **Configure Global Data** l **Global Data Configuration** l
3	Enter the configuration's variable IDs, symbols (description), and lengths.
4	Click the **Update Global Data Configuration** button to update the file. This creates the file `glbdata.ini`. Full Path follows: `ftp://NOE_IP_ADDRESS/wwwroot/conf/glbdata/glbdata.ini` Note: You substitute `NOE_IP_ADDRESS` with an address like 239.255.255.255. You want to check the address with your system administrator.
5	Using the FTP process, upload the `glbdata.ini` file to a PC. See the section below, Uploading a glbdata.ini file.
6	Stop each controller before you do the customization.
7	Using the same path, download the `glbdata.ini` file to the other devices. See the section below, Downloading a glbdata.ini file
8	Connect your Web browser to each device to customize start address and the Publish / Subscribe setting.

Note: The Global Data Configuration page is populated with data the comes from the `glbdata.ini` configuration file.

Uploading a glbdat.ini File

The following procedure describes the steps to upload the glbdata.ini file.

Step	Action
1	At the DOS prompt type **FTP** followed by the IP address and press **Enter**.
2	At the User prompt type **FTP** Username and press **Enter**.
3	At the password prompt enter your FTP Password and press **Enter**.
4	At the FTP prompt type **cd wwwroot/conf/glbdata** and press **Enter**.
5	At the FTP prompt type **get** and press **Enter**.
6	At the local file prompt type **glbdata.ini** and press **Enter**.
7	At the remote file prompt type **glbdata.ini** and press **Enter**.

Downloading a glbdat.ini File

The following procedure describes the steps to download the glbdata.ini file.

Step	Action
1	At the DOS prompt type **FTP** followed by the IP address and press **Enter**.
2	At the User prompt type the FTP Username and press **Enter**.
3	At the password prompt enter your FTP Password and press **Enter**.
4	At the FTP prompt type **cd wwwroot/conf/glbdata** and press **Enter**.
5	At the FTP prompt type **put** and press **Enter**.
6	At the local file prompt type **glbdata.ini** and press **Enter**.
7	At the remote file prompt type **glbdata.ini** and press **Enter**.

Verifying System Operation

To ensure that the system is operational, do the following:

Step	Action
1	Verify all controllers are running.
2	Look at the health of all variables using the **Global Data Diagnostic** page.

Configuring the Global Data (Publish / Subscribe) Utility

Overview

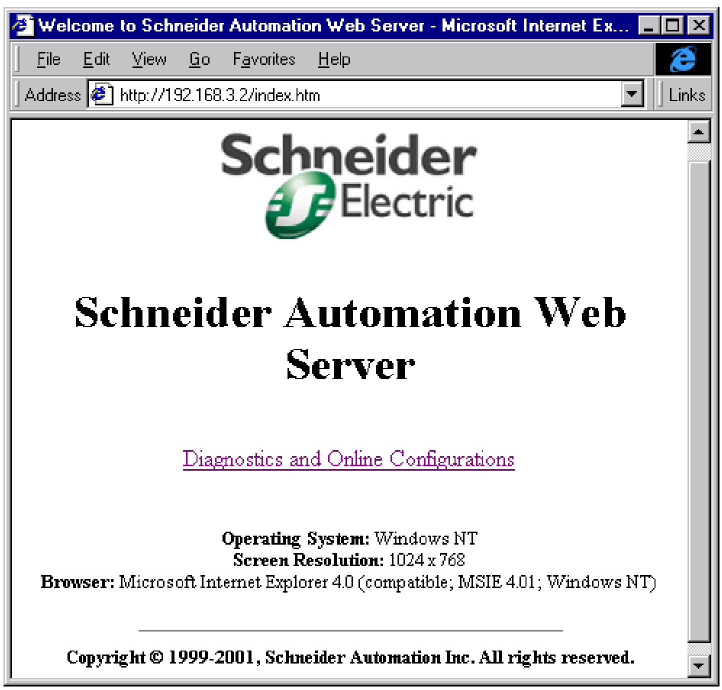
Whether you use the Configure Each Device Separately or the Copy Configuration method, the procedure to configure individual parameters is the same. Therefore, in order to use the Global Data (Publish / Subscribe) utility in the NOE, you need to configure the Global Data parameters including:

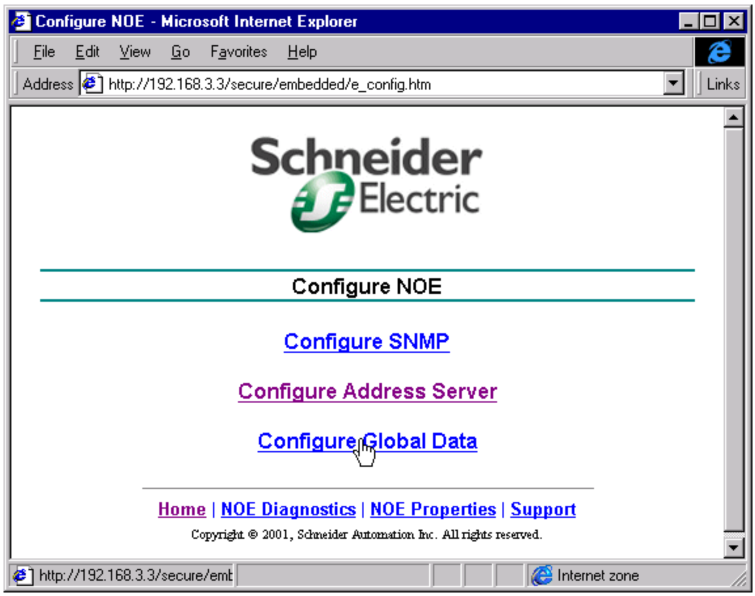
- Distribution period
- Multicast filtering
- Health bit location
- Global Data base address
- Group IP address

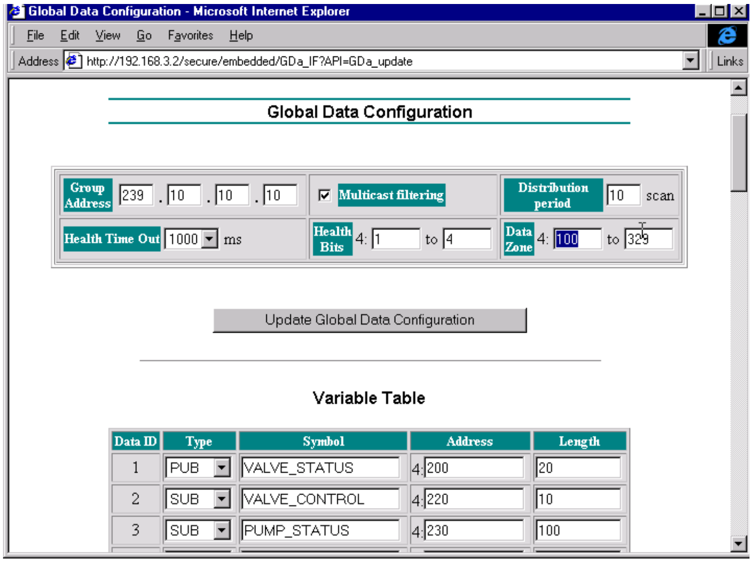
The following sections describe in detail the exact steps to configure each parameter via the **Global Data Configuration** page.

Accessing the Global Data Utility

You access the Global Data utility through the **Global Data Configuration** page.

Step	Action
1	On your screen click the **Diagnostics and Online Configurations** link. 

Step	Action
2	When you click the **Diagnostics and Online Configurations** link, you are requested to enter user name and password
3	Enter your user name and password. The **Configure NOE** page appears.
4	Click the **Configure NOE** link. The **Configure NOE** page appears.
5	Click the **Configure Global Data** link. 

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>6</td><td>The Global Data Configuration page appears.


In this screen you may make changes to the configuration.</td></tr>
</table>

Configuring Global Data

After you have completed the Modelling System Configuration process using the second method, Copy Configuration, then you modify the following parameters:
- Distribution period
- Health Time Out
- Health Bits location
- Start address
- Type: Pub / Sub / None

Please do NOT change Symbol (description), and Length.
To change the Global Data variables of the group box on the **Global Data Configuration** page, follow the instructions below.

Step	Action
1	Adjust the **Distribution Period Cycle**. Enter a value from 1 through 50. **Note: Distribution period** is the minimum number of controller scan times before an update will occur.
2	Before entering a value in the **Group address** field, identify the station's distribution group. The **Group address** entry will be an IP address from `224.0.0.0` through `239.255.255.255`. **Group address:** the Class D Multicast IP address used for a distribution group. All members of this distribution group are configured to use the same group address, and therefore, all members can communicate with each other using Global Data.
3	Set the timeout in the **Health Time Out** field. This value is measured in milliseconds and can be set to a value that ranges from 50 through 1000 ms (increase in units of 50ms). **Note: Validity Time** is the maximum time between received subscriptions before a subscription is declared unhealthy (faulty).
4	In the **4x Starting Address** set the **Data Zone** field.
5	If you are connected to an Ethernet switch that supports multicast filtering, click the **Multicast filtering** check box.
6	Enter 4x Register Location for the Health Bits. This is the register where the health bit will be stored

Note: Health bits run in different directions.
- I/O scanner health bits run left to right.
- Global Data health bits run right to left.

Changing Global Data Variables

To change the Global Data variables that appear in the **Variable Table** area, follow the instructions below.

Step	Action
1	Highlight the identification number in the **Data ID** column.
2	In the Type column select the publish / subscribe variable type from the drop down list. Three options are available publish, subscribe, or none. These options display on the screen as follows: ● **NONE** ● **SUB** ● **PUB**
3	In the **Symbol** column you may enter text to describe the variable.
4	In the **Address** column you see the application address for this variable. **Note:** This is a read only field.
5	In the Length column for each row, type a value, which represents the number of 4x registers. The ending 4x register field is automatically updated. If you are using the second method, **Copy configuration**, you update **Length** the first time only.
6	When you are finished, click the **Update symbols** button.

Verifying System Operation

To ensure that the system is operational, do the following:

Step	Action
1	Verify all controllers are running.
2	Look at the health of all variables using the **Global Data Diagnostic** page.

Multicast Filtering

Overview

Your NOE may offer the multicast filtering functionality.

The global data service synchronizes several stations located in a distribution group. A distribution group is a set of stations identified by using the same IP multicast address for all stations in the group. By using the same IP address for multiple devices, multicast exchanges can be used to distribute global data. Several independent distribution groups can coexist on the same sub-network. Each distribution group posses its own unique IP multicast address.

Early versions of Switches treat multicast packets as a broadcast. Therefore, transmitting broadcasts to all nodes, and thereby suppressing all benefits of both switching and multicasting. Newer version of switches provide automatic Multicast Filtering, and consequently only forward multicast traffic to the ports that are connected to registered end-stations.

Multicast Filtering uses the GARP Multicast Registration Protocol (GMRP) to inform a switch which IP Multicast Addresses are of interest to the attached device GMRP is defined in the IEEE 802.1D-1998 Standard, which is available as a free download at: http://IEEE802.org.

In order to use Multicast Filtering, you need to:
1. Ensure that your Switch supports IEEE 802.1D - 1998
2. Click the **Multicast filtering** check box on the **Global Data Configuration** area on the Web page.

Reducing Traffic

Multicast Filtering helps to reduce the traffic on a network, because broadcasts are sent only to interested, or subscribed, devices.
For distributed applications and one to many communications multicast affords advantages over unicast:
- Utilizes the network bandwidth more efficiently
- Sends a single transmission instead of multiple transmissions.
- Reduces collisions
- Optimizes the performance of Ethernet module processing

Using Multicast Filtering

These ConneXium switches support multicast filtering. Other switches from alternate vendors also support multicast filtering.

Switch	Description
499NES17100	Managed Switch with 7 ports 10/100BASE-TX
499NOS17100	Managed Switch with 5 ports 10/100BASE-TX and 2 ports 100BASE-FX

Transferring Data with the I/O Scanner 140 NOE 771 -00, -01, and -11 only

6

At a Glance

Introduction

This chapter discusses the NOE 771 -00, -01, and -11 modules' I/O scanner capabilities and includes procedures for configuring the I/O scan list using Concept, ProWORX, and Modsoft.

What's in this Chapter?

This chapter contains the following topics:

I/O Scanner Concepts

Overview

The following information describes how to configure the I/O scanner.

Introduction

All NOE 771 00 modules provide an I/O scanner that the user can configure with the Concept, ProWORX, or the programming panel. This way, the user can configure data and transfer it between network nodes without using the MSTR instruction.

I/O Scan List

The I/O Scanner is a feature of the NOE module, which allows repeated reading and/or writing to Input/Output devices.
The I/O scan list is a configuration table that identifies the targets with which repetitive communication is authorized. The list contains enough information to enable each target to construct the MODBUS message addressed to the specified remote device and to designate where on the local controller the input and output data are to be mapped at the end of the scan. While the controller is running, the NOE module transfers data to and from the controller's registers and coils as indicated by the I/O scan list.
The user configures the I/O scan list with the Concept, ProWORX, or the programming panel. There can be multiple instances of the I/O scan list (Peer Cop restrictions apply). The individual scan lists for each module are identified by the Quantum backplane slot number where the NOE is installed.

I/O Scanner Definitions

Note: Health bits run differently.
- I/O Scanner health bits run left to right.
- Global Data health bits run right to left.

The following table lists and defines the terms that are used to describe the I/O Scanner operation.

Term	Definition
Scan List	The list of input and/or output devices that the NOE module is configured to scan.
Specific Input	Input to the controller, on the backplane where the NOE resides.
Specific Output	Output from the controller, on the backplane where the NOE resides.
Peer Cop	Legacy I/O Scanner support to upgrade MODBUS Plus I/O applications to Ethernet.
Ethernet I/O Scanner	Provides high performance cyclic communication service to the controller.

Peer Cop and Enhanced MODBUS/TCP Scanners

The NOE 771 00 module's design provides you with the ability to configure its MODBUS I/O Scanner as either a Peer Cop or Enhanced MODBUS scanner. The determination as to which scanner is used depends on the programming package that is installed on your system. If you presently use Modsoft, you must configure the I/O Scanner for Peer Cop operation; if you presently use Concept or ProWORX, you can configure the I/O Scanner for either Peer Cop or enhanced MODBUS/TCP operation.

Peer Cop I/O Scanner Features

The following table lists the characteristics of the Peer Cop based MODBUS I/O Scanner.

Parameter	Value
Max. No. of Devices	64
Max. No. of Input Words	500
Max. No. of Output Words	500
Timeout Value	Global Setting (20 Msec to 2 Secs in 20 mSec increments)
Input TimeOutState	Global Setting (Zero or Hold)
IP Address	Derived from MODBUS address (must be on NOE's subnet)
Remote Register Reference	Not configurable - 400001 is used
Destination ID	Not settable, set to 0
Operation through a MODBUS Plus to Ethernet bridge	Not supported

Enhanced MODBUS I/O Scanner Features

The following table lists the characteristics of the Enhanced MODBUS I/O Scanner.

Parameter	Value
Max. No. of Devices	64 or 128
Max. No. of Input Words	4,000
Max. No. of Input Words	4,000
Timeout Value	Individual Setting (1 Msec to 2 Secs in 1 mSec increments)
Input TimeOutState	Global Setting (Zero or Hold)
IP Address	Configure from Concept or ProWORX
Destination ID	Not settable, set to 0
Operation through a MODBUS Plus to Ethernet bridge	Not supported
Operation through a MODBUS bridge	Supported

I/O Scanner Support

The following table summarizes the permissible mix of I/O scanners and NOE modules per CPU. A maximum of two NOE modules can be configured as I/O scanners per controller.

Quantum CPU Type	No. of NOEs Supported	Max No. of NOEs Configured as Peer Cop I/O Scanners	Max No. of NOEs Configured as Ethernet I/O Scanners
140 CPU 113 02	2	2	2
140 CPU 113 03	2	2	2
140 CPU 213 04	2	2	2
140 CPU 424 02	6	2	6
140 CPU 434 12	6	2	6
140 CPU 534 14	6	2	6

Configuring the I/O Scan List Using Concept

Overview

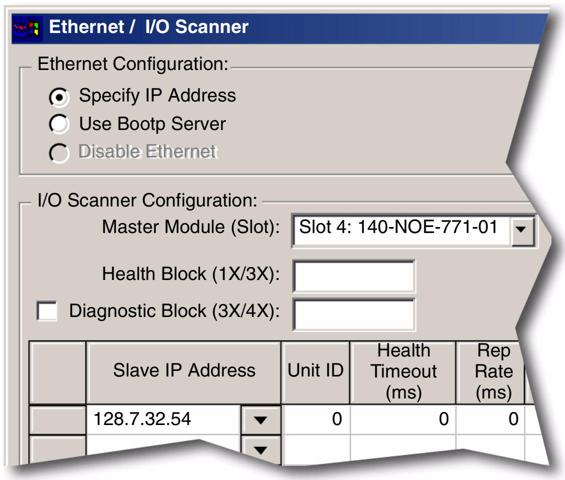
The following information describes how to configure the I/O scan list.

Introduction

After the NOE 771 -00, -01, or -11 has been configured using Concept (see *Configuring the Module with Concept, p. 57*), you can assign parameters for I/O scanning. This involves creating the I/O scan list containing all of the input and output devices that the NOE module will scan.

IP Address

Type the IP address of the slave module in the **Slave IP Address** column.



Unit ID

If the slave module is an I/O device attached to the specified slave module, use the Unit ID column to indicate the device number. The Unit ID is used with the MODBUS Plus to Ethernet bridge to route to MODBUS Plus networks.

Health Timeout

The Health Timeout is used for setting the health bit. If the response arrives before the end of the Health Timeout period, the health bit is set; otherwise it is cleared. If the Health Timeout is zero, the health bit is set to true once communications are established, and it is never cleared.

Rep Rate

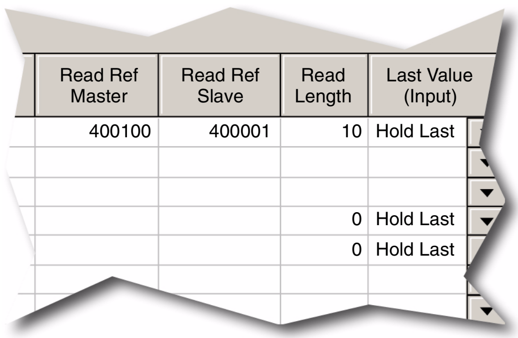
Use this column to specify the lower bound in milliseconds (ms) between transactions to this node. Valid values are 0... 50,000 ms (1 min.). The NOE module takes this value and rounds up to a multiple of 17 ms. Since the update of I/O is synchronized to the CPU scan, if the CPU scan is greater than the configured lower bound, then the actual update rate will be at the rate of the CPU scan. To obtain the maximum rate, specify a zero.

For example, if a user specifies 10 ms, then it is rounded up to 17 ms. If the controller's scan time is 5 ms, then the time between transactions must be greater than or equal to 17 ms. On the other hand, if the controller's scan time is 200 ms, the time between transactions must be greater than or equal to 200 ms.

Read

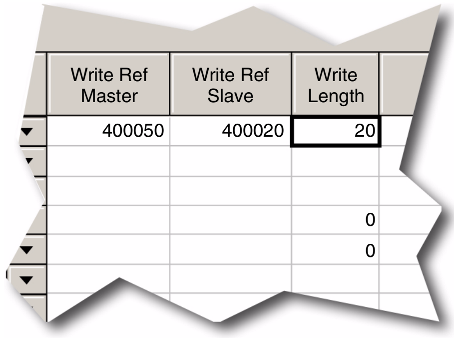
Use the read function to read data from the remote node. The Read Ref Master column specifies the local address for the read response. The Read Ref Slave column specifies the first 4x register of the remote node to be read. The Read Length column specifies the number of registers to read.

The following figure includes sample values for the Read Ref Master, Read Ref Slave, and Read Length parameters.



Write

Use the read function to write data to the remote node. The Write Ref Master column specifies the local address of the write data. The Write Ref Slave column specifies the first 4x register to be written to the remote node. The Write Length column specifies the number of registers to write.
The following figure includes sample values displaying in the Write Ref Master, Write Ref Slave, and Write Length parameters.



Read and Write

You may include both the read and write commands in the same row.

Description

You can type a brief description (up to 32 characters) of the transaction in the Description column.

Configuring the Health Block

The Health Block is located at a block of 3x registers or 1x coils. For 1x coils it must start on a 16-bit boundary. Each device that is configured has a corresponding health bit in the Health Block. If the health bit is one, the remote device is healthy. It the health bit is 0 (zero), the remote device is unhealthy.
As shown in the following tables, each row that is configured is mapped to a bit position.

Word 1 Bit Positions															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

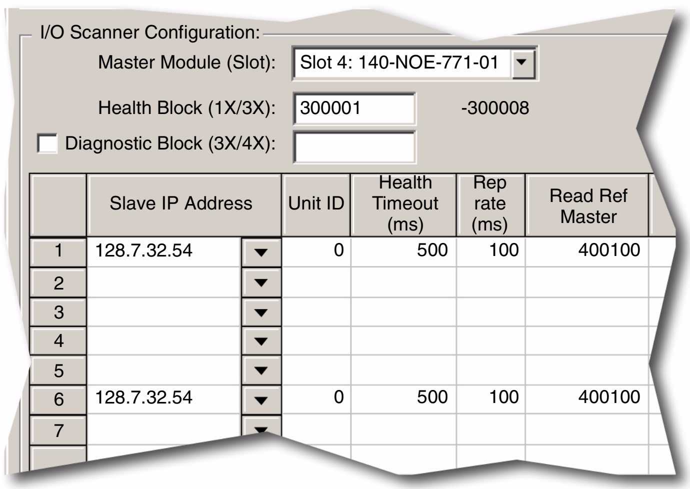
Word 2 Bit Positions															
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

Word 3 Bit Positions															
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48

Word 4 Bit Positions															
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64

Starting Location of Health Block

As shown in the following figure, to specify the starting 1x/3x location of the Health Block, enter the desired address into the Health Block text box



Completing the I/O Configuration

Overview

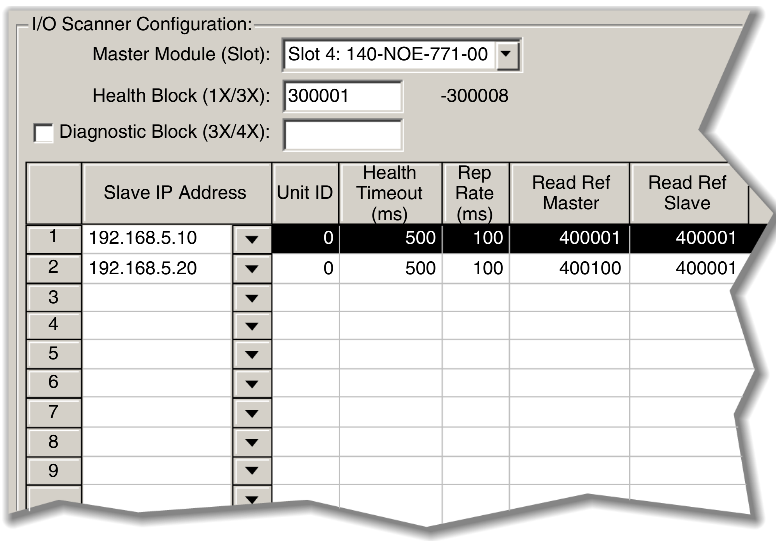
The following information describes the functions to use to complete your Ethernet I/O configuration.

Introduction

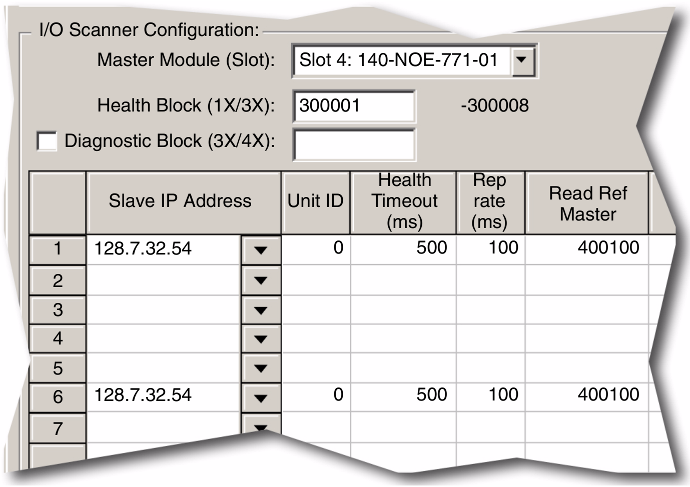
This section describes how to complete your Ethernet I/O configuration using the Copy, Cut, Paste, Delete, Sort, and Fill Down buttons.

Copy and Paste

Use the following procedure to copy and paste entire rows within your configuration, so that you can save time when typing similar read and write commands.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>As the following figure shows, select the row you want to copy by clicking on the row number at the far left.

</td></tr>
<tr><td>2</td><td>Click Copy.</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>3</td><td>Select the row where you would like to paste the data (by clicking on the row number at the far left).



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

Cut and Paste

To move a row within the configuration list, follow the directions for copying, only use the **Cut** button instead of the **Copy** button.

Delete

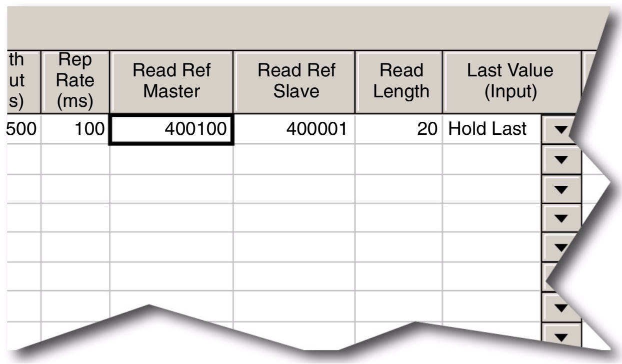
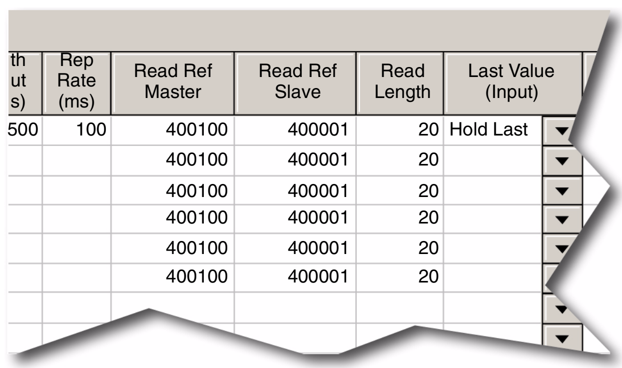
To delete a row from the configuration list, select the row by clicking on the row number at the far left. Then click the **Delete** button.

Sort

To sort the I/O configuration list, select a column by clicking on the column heading (i.e., Read Ref Master). Then click the **Sort** button.

Fill Down

The following procedure shows how to copy part of any row to the next row or to a series of adjoining rows using the Fill Down button

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>The following figure shows how to select the data to copy. Use your mouse to select both the data you would like to copy and the cells you would like to copy the data into. Note that you must select one contiguous block of cells with the data to be copied in the first row. You cannot select two separate blocks.

</td></tr>
<tr><td>2</td><td>Click Fill Down.
Result: As shown in the following figure, the data from the first row is copied into the selected cells.

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

Configuring the I/O Scan List Using ProWORX NxT

Overview

The following information describes how to configure the NOE 771 module from your programming panel using ProWORX NxT program.

Introduction

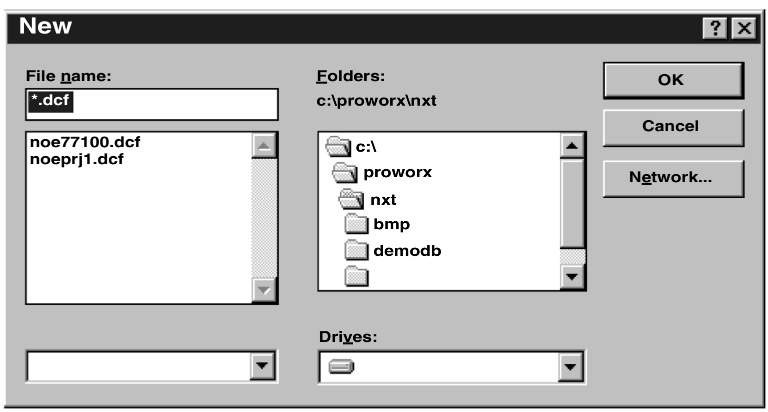
This section discusses how to configure the NOE 771 module from your programming panel using ProWORX NxT program. This process assumes you have switched to an Ethernet network so you can choose I/O Scanner instead of the Peer Cop. This allows you to configure data blocks to be transferred between controllers on a TCP/IP network.

The following three procedures are used in this configuration process:

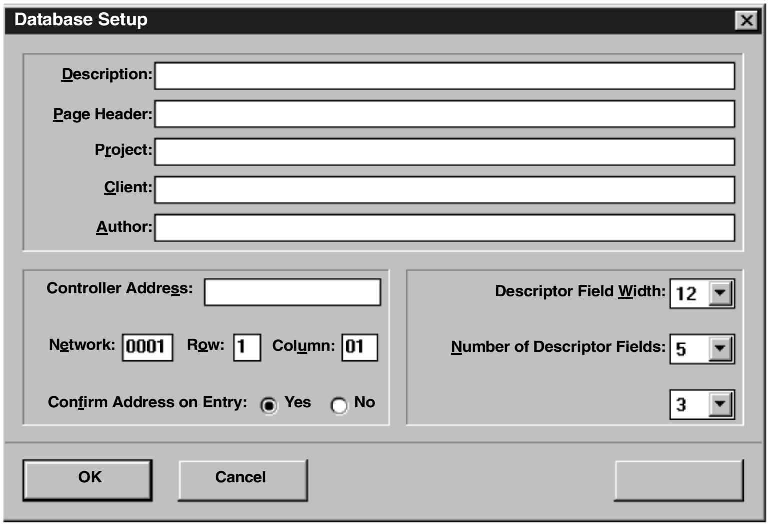
- Selecting Your PLC
- Accessing and Editing the Traffic Cop
- Setting the Number of NOEs and Configuring the Ethernet Address Parameters

Procedure for Selecting Your PLC

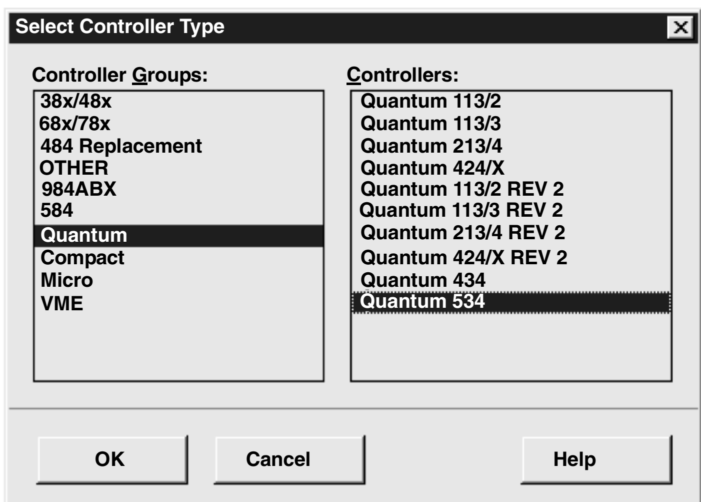
The following steps describe how to select a CPU:

Step	Action
1	Open ProWORX NxT on your programming panel (PC). **Result**: As the following figure shows, the ProWORX NxT initial screen displays. 
2	From the File menu, select **New**. **Result**: The New dialogue box appears. 

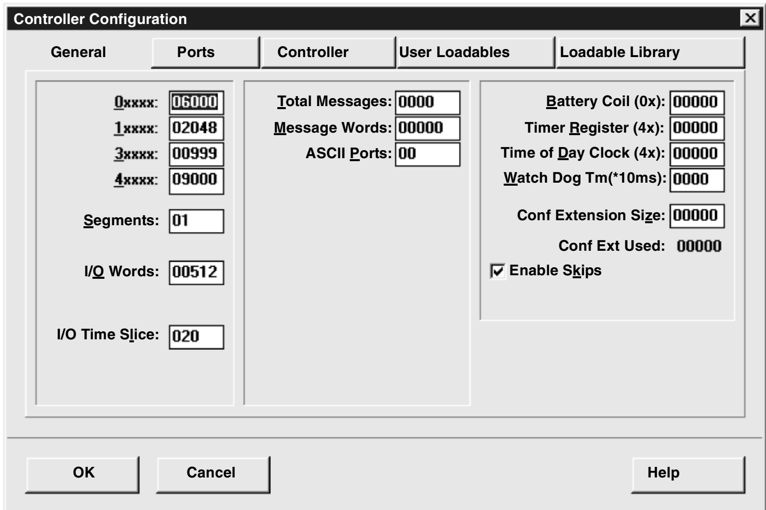
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>3</td><td>Type a file name in the File Name text box. Select the drive from the drop down box labeled "Drivers". Select the file folder where you want to save the new database labeled Folders. Press the <OK> button
Result: As the following figure shows, the Database Setup screen displays.



</td></tr>
<tr><td>4</td><td>Fill in the text boxes as you require. Press the <OK> button.
Result: As the following figure shows, the Select Controller Type dialog box displays.



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

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>5</td><td>From the Controller Groups list box on the left, select the Quantum group. From the Controllers list box on the right, select the CPU that is installed in your Quantum backplane. Click the <OK> button.
Result: As the following figure shows, the Controller Configuration screen displays.


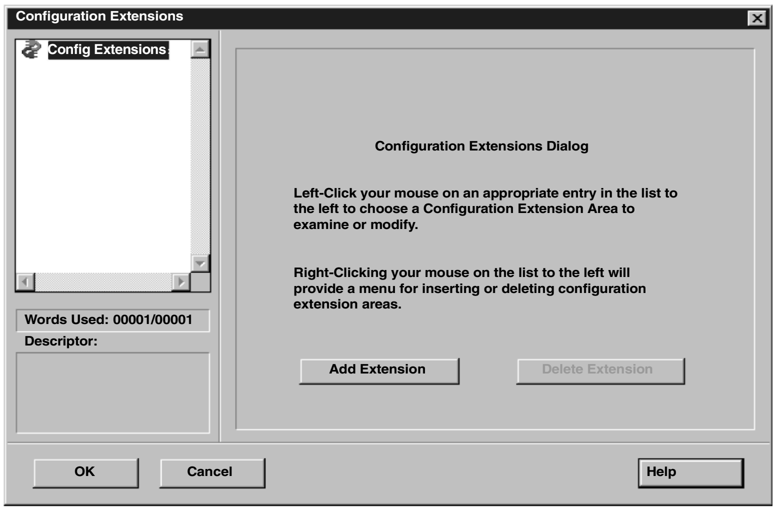
</td></tr>
<tr><td>6</td><td>You must define a value in the Conf Extension Size on the right side of the Controller Configuration screen. This value is the amount of memory you require. Press the <OK> button.</td></tr>
</table>

Procedure for Accessing and Editing the Traffic Cop

The following steps describe how to access and edit the Traffic Cop.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>From the Configuration menu, select Traffic Cop.

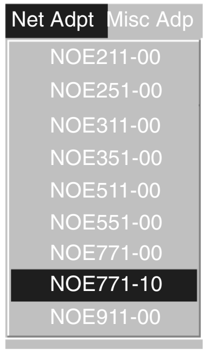
Result: As the following figure shows, the Traffic Cop screen displays.



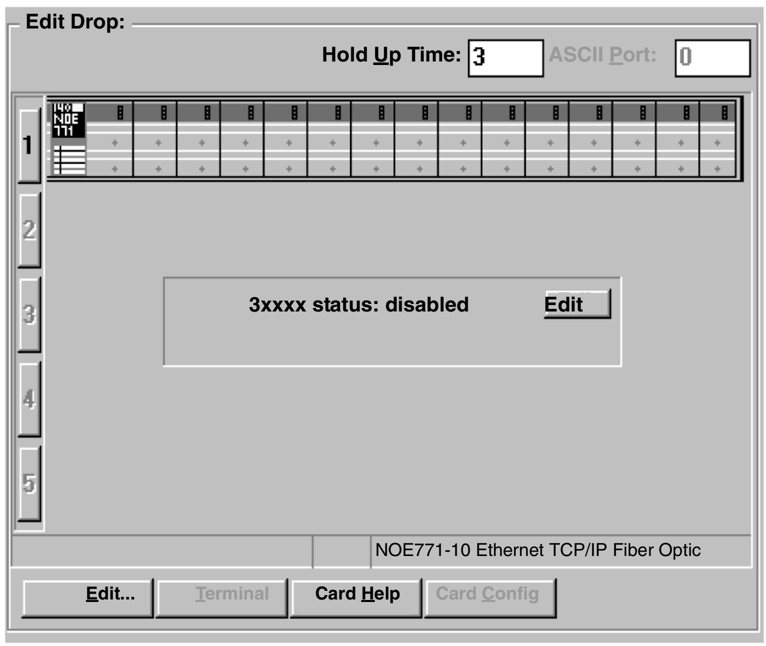
</td></tr>
<tr><td>2</td><td>From the Quantum Traffic Cop menu on the left, click the "+" sign to expand the Traffic Cop tree. Choose the Rack and Slot where you want the NOE 771 module inserted.</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>3</td><td>

As shown in the following figure, from the Net Adpt menu of the Traffic Cop screen, select **NOE 771**.



Result: As the following figure shows, the NOE 771 module is inserted into the specified location, within the Edit Drop field of the Traffic Cop screen.



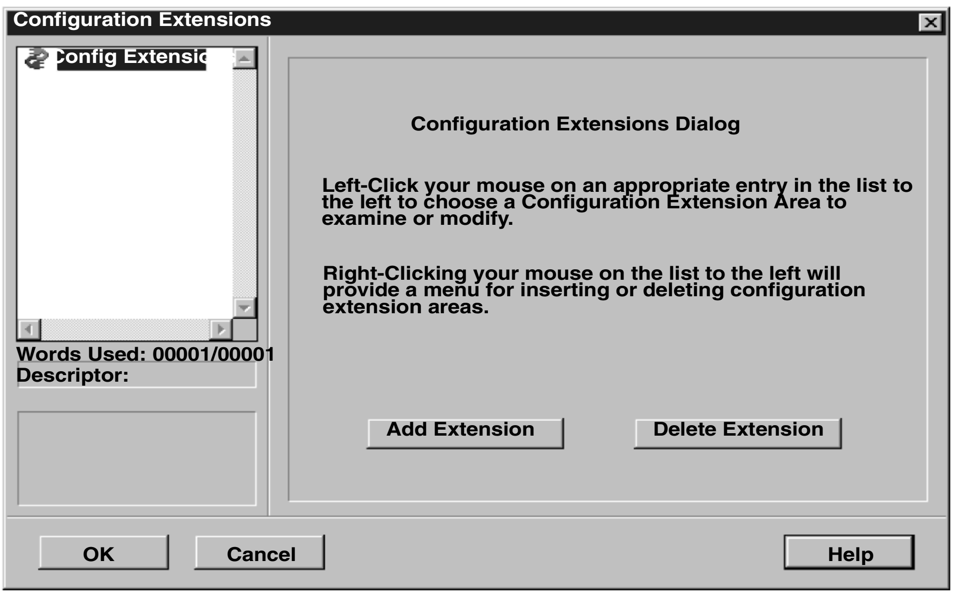
</td></tr>
<tr><td>4</td><td>Press the <OK> button.</td></tr>
</table>

Setting the Number of NOEs and Configuring the Ethernet Address Parameters

You can set values for the following parameters from the Configuration Expansion screen:
- Select the number of NOE 771 modules
- Configure the Ethernet Address Parameters, TCP/IP

Procedure for Setting the Number of NOEs and Configuring the Ethernet Address Parameters

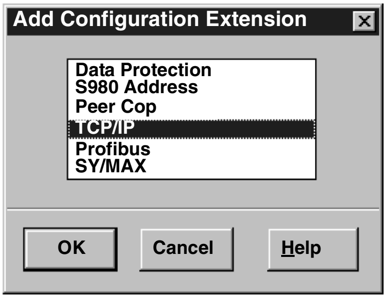
The following steps describe how to select the number of NOE 771 modules and how to configure the Ethernet address parameters.

Step	Action
1	As shown in the following figure, from the Configuration menu, select **Config Extensions**.  **Result**: As shown in the following figure, the Config Extension screen appears. 

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2</td><td>

Click the **Add Extension** button.

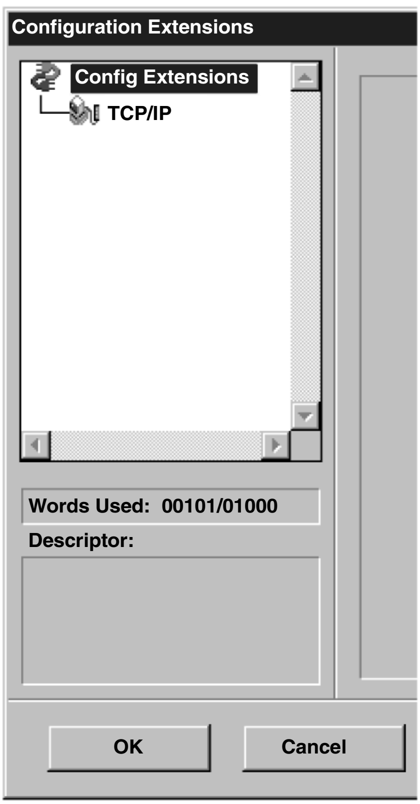
Result: As shown in the following figure, the Add Configuration Extension dialog box displays.



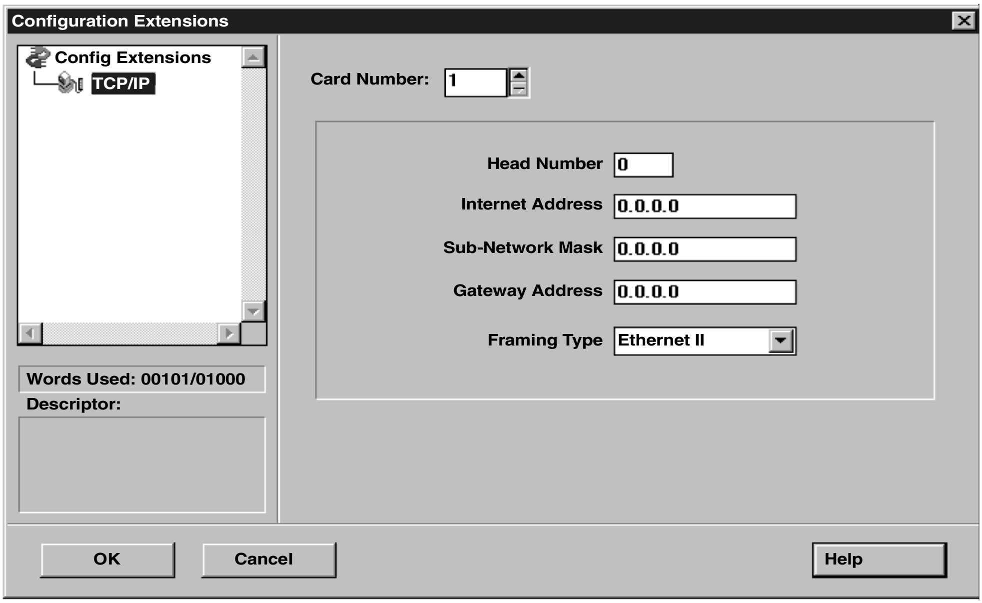
</td></tr>
<tr><td>3</td><td>

Select **TCP/IP** from the list, and click the <**OK**> button.

Result: As shown in the following figure, the TCP/IP Configuration Extension is added to the left menu of the Configuration Extensions screen.



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

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>Click TCP/IP Config Extension in the left menu.
Result: As shown in the following figure, the details of the TCP/IP configuration appear on the right side of the Configuration Extension screen.


</td></tr>
<tr><td>5</td><td>As shown in the following figure, select the Card Number (Rack Number) by pressing the.button in the Card Number combo box.



</td></tr>
<tr><td>6</td><td>Complete the fields for each Card Number. The Head Number text box represents the Rack in which the NOE 771 module is located.
Click the <OK> button when complete.</td></tr>
</table>

Setting Up the I/O Scanner Using ProWORX NxT

At this point, you are ready to set up the I/O Scanner. The I/O Scanner provides data transfer between two or more NOE 771 00 and other Modbus or TCP/IP devices. It allows you to simultaneously configure up to 64 connections.
To configure the I/O Scanner, you need to set values for the following parameters:
- Specify the specific I/O groups to be scanned
- Configure the transaction parameters
- Set the hardware clock for when the data is to be collected

Procedure for Setting Up the I/O Scanner Using ProWORX NxT

The following steps describe how to specify the I/O groups to be scanned.

Step	Action
1	From the Network Editor, on the **Configuration** menu click **Config Extensions**. The Configurations Extensions dialog box appears.
2	In the Config Extensions tree, right-click on **Config Extensions** and select **Add Extension**.
3	Select **Ethernet I/O Scanner**. The parameters for the CDE appear in the details area
4	In the Health Block field, type a 1xxxxx or 3xxxxx address. Note that all 1xxxxx addresses are based on a 16-bit boundary. For example: 100001, 100017, 100033, etc.
5	Double-click on an empty transaction to add a new transaction or double-click on an existing transaction to edit it. The Transaction dialog box appears.

Procedure for Configuring the Transaction Parameters

The following steps describe how to configure the transaction parameters.

Step	Action
1	Double-click on an empty transaction to add a new transaction or double-click on an existing transaction to edit it. The Transaction dialog box appears.
2	Configure the transaction parameters.

Establishing Configuration Extension Memory for Peer Cop

Overview

The following information describes how to set up Peer Cop.

Introduction

By default, the Peer Cop capability is disabled. If you want to use Peer Cop to handle Modbus Plus communications, you need to enable this capability and adjust the amount of configuration extension memory.

Note: If you are upgrading your network to Ethernet, you should consider the option of ignoring Peer Cop and instead, configuring extension memory to use the enhanced Modbus/TCP IO Scanner feature of your NOE 771 00 module. (See *Transferring Data using Communication Blocks, p. 69*.)

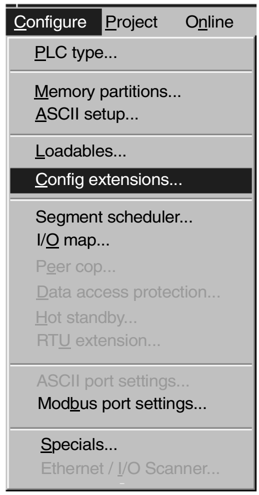
How Much Memory?

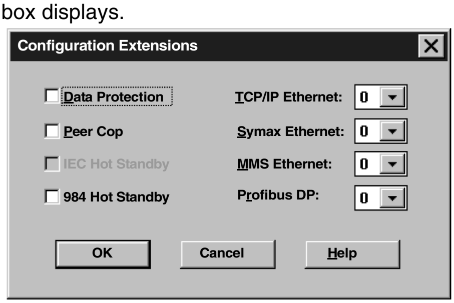
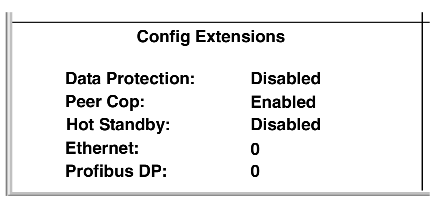
The minimum Peer Cop memory requirement is 20 words; the maximum is 1366 words.
The following table lists the guidelines for estimating the amount of extension memory you will need for your Peer Cop database

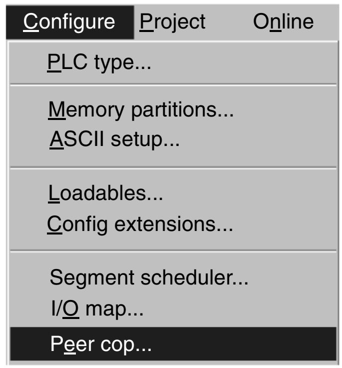
For...	**Add...**	**Up to a maximum of...**
Overhead	9 words	--
Global output	5 words	--
Global input	number of words = number of devices x (1 + 2 x number of device subentries)	1088 words
Specific output	2 words for every device entry in Peer Cop	128 words
Specific input	2 words for every device entry in Peer Cop	128 words

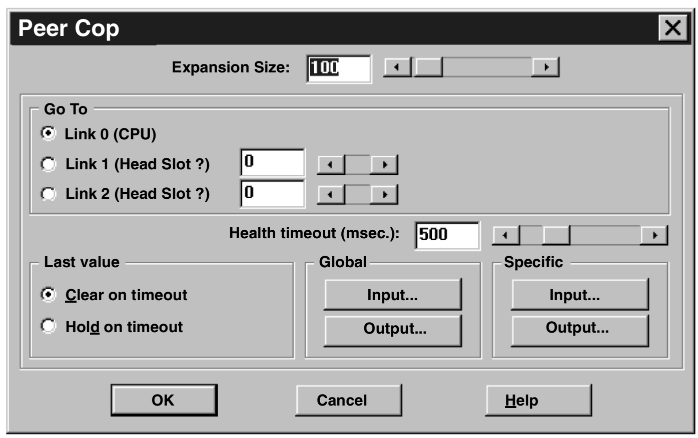
Procedure for Setting Up Peer Cop

The following steps allow you to enable Peer Cop and adjust the amount of Configuration Extension memory from the PLC Configuration screen.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>As the following figure shows, from the Configure menu, select Config extensions, or double-click anywhere in the Config Extensions region of the screen.


Result: As the following figure shows, the Configuration Extension dialog box displays.

</td></tr>
<tr><td>2</td><td>Click on the check box next to Peer Cop, then click the <OK> button.
Result: As the following figure shows, the Peer Cop status changes from Disabled to Enabled in the PLC Configuration screen.

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

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>3</td><td>As the following figure shows, from the Configure menu, select Peer Cop.


Result: As the following figure shows, the Peer Cop dialog box displays.

</td></tr>
<tr><td>4</td><td>Modify the amount of configuration extension memory allocated to Peer Cop by typing a new value in the Expansion Size field, or by adjusting the sliding scale next to the field.</td></tr>
<tr><td>5</td><td>Click the <OK> button.</td></tr>
</table>

Configuring the I/O Scan List Using Modsoft

Overview

The following information describes how to use Peer Cop, which is available in the Modsoft program, to configure the I/O scan list.

Introduction

The Peer Cop input screens in the Modsoft program will be used to configure the I/O scan list.

The Peer Cop configuration extension allows you to configure certain continuous, fixed format communications between the controller (in which it is defined) and all other nodes on the same subnet.

Each Peer Cop configured communication specifies a source data block. The source data block is of fixed location and length and is continuously moved to a fixed destination data block. This data transfer type is useful for transferring state information between controllers and for communicating with slave devices on the Ethernet.

Peer Cop communication is not appropriate for sequence dependent communication that must be performed exactly once. The standard MSTR element is used for those logic dependent requirements with certain restrictions.

Similar to the I/O Map, the Peer Cop can be configured only with the controller stopped. Once the PLC is configured and started, the transfers are performed automatically.

A menu item in the Peer Cop is available to delete the current node on the screen. A warning is given and the node is deleted if (Y) is answered. If the last node is deleted, a window opens to allow entry of a node. This condition is identical to the initial screen of an empty Peer Cop.

Current Limitations

The following table describe the limitations of the operating parameters as well as the recommended settings for other parameters.

Parameter	Limitation/Special Recommendation
Maximum Input Length	32 Words
Maximum Output Length	32 Words
Total I/O Scan Data Length	500 Words
IP Address	An I/O device is currently limited to having an IP address in the form of AAA.BBB.CCC.DDD, where AAA.BBB.CCC are the same as the NOE's IP address and the subnet address of DDD is limited to 1... 64
Operation thorough a Modbus Plus to Ethernet bridge	Not supported
Destination ID	Not user supportable; fixed at 0

Storage Requirements — NOE 771 00

Before selecting Peer Cop from the Cfg Ext pulldown list, you must use ExtSize to set the memory storage requirements.

Note: The remainder of the CfgExt pulldown functions remain disabled until the ExtSize is set.

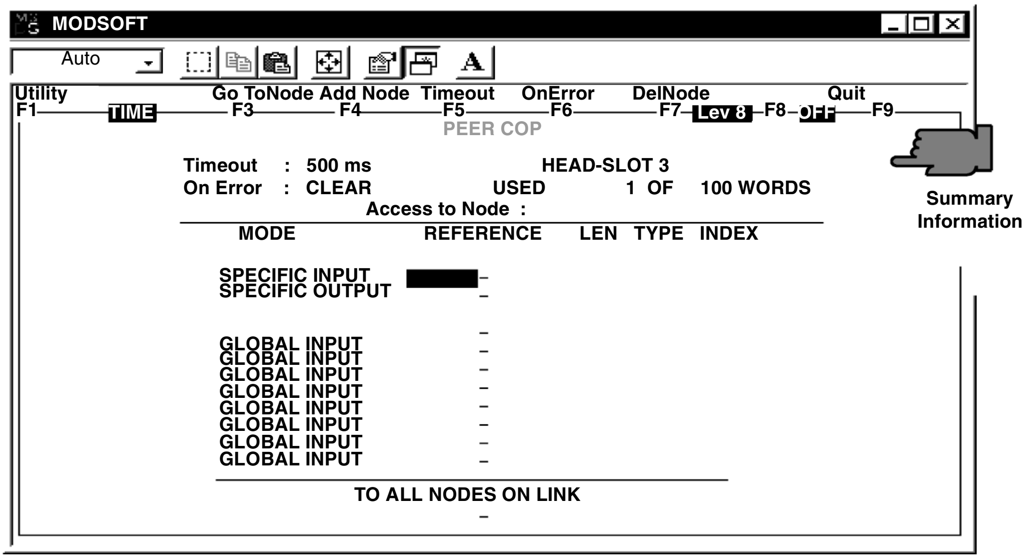
The following list shows the four types of Peer Cop requests
- Peer Cop Global Data input (not supported
- Peer Cop Global Data output (not supported)
- Specific data input
- Specific data output

Note: Peer Cop Global Data Extension
For Ethernet TCP/IP network operations, only specific data input and output are supported.
The NOE ignores Peer Cop Global Data configuration.
- Do not fill in Global input or Global output.

The following items are considerations for estimating your memory needs:
- If Specific Output is configured, add 2 words for each device entry (64 maximum. Maximum is 64 x 2 = 128 words
- If Specific Input is configured, add 2 words for each device entry (64 maximum). Maximum is 64 x 2 = 128 words. Based on the above, the minimum possible size for Peer Cop is 20 words, and the maximum possible size is 1366 words, for each of up to 3 links.

Specific Input/Output Configuration

The following figure shows the default screen for Peer Cop entry (CRX). The screen, which is labeled "Peer Cop, is a data entry template comprising all four data types and providing a summary of settings that apply to the specific link/node as well as timeout, error handling, and statement of memory words used. Initially, the cursor is in the Head Slot field. If you are not editing an initial template, you may press the Esc key which re-positions the cursor to the SPECIFIC INPUT field. To traverse the Heads and Nodes you can re-display the Add Node select box from the Main menu line.



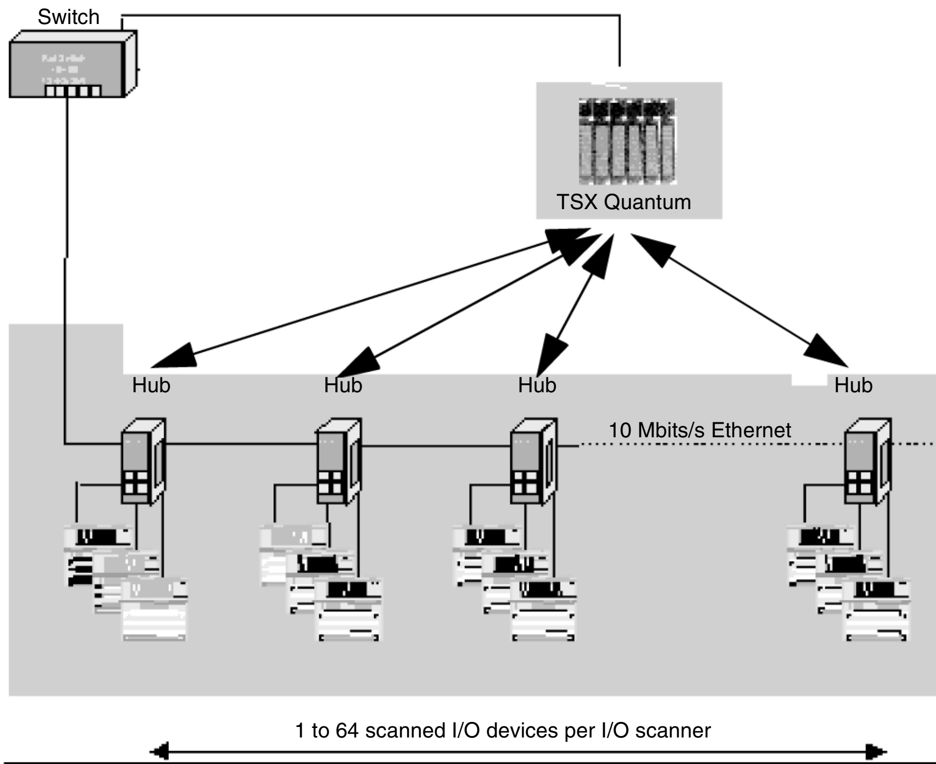
Specific Output

Specific output comes from the Controller located in the rack where the NOE resides. Specific output data can be set from the NOE to the remote node on the subnet by a Modbus Write. The source of each specific output block is a contiguous region of 0x, 1x, 3x or 4x state RAM, which varies from 1 to 32 words in length. If discretes are used, they must start on a word boundary (00001, 00017, 00033, etc.). The Type default (BIN or BCD) is put in by the controller. Where different types can be specified, you make the entry from a display list that is viewed by keying the return key while the cursor is on the TYPE field.

Note: The NOE 771 00 ignores the BIN BCDs setting and always uses a BIN format.

Specific Input

As shown in the following figure, specific input goes to the Controller on which the NOE resides. NOE obtains specific input data from a remote node on the subnet with a Modbus read. You can specify all Specific Input blocks sent to this controller from the other specified stations on the node. The destination of each block of specific input is a contiguous region of 0x, 1x, 3x or 4x state RAM, which varies from 1 to 32 words in length. If discretes are used, they must start on a word boundary (00001, 00017, 00033, etc.).



In the CRX figure, the Quantum NOE is configured to write 1 word from 400050 to the Momentum EIO at IP Address 198.202.137.2, and to read 1 word from the EIO into register 400100.

The following figure shows the Modsoft menu that reflects the above situation. It provides an example of the Length, Source, and Type data fields; and it illustrates the above example completely filling in the template fields.

```
Utility            GoToNode  AddNode  Timeout   OnError                    Quit
F1________F2________F3________F4________F5________F6________F7_Lev 8 _F8_OFF___  ____
                                        PEER COP

          Timeout    :  500 ms                    HEAD-SLOT 2
          On Error   :  CLEAR               USED      17  OF   1000 WORDS
                                   Access to Node  :   2
          _______________________________________________________________
              MODE              REFERENCE     LEN    TYPE    INDEX
           SPECIFIC INPUT        400100-400100    1      BIN
           SPECIFIC OUTPUT       400050-400050    1      BIN
           GLOBAL INPUT               -
           GLOBAL INPUT               -
           GLOBAL INPUT               -
           GLOBAL INPUT               -
           GLOBAL INPUT               -
                                      -
                                      -
                                      -
           ____________________________________________________
                          TO ALL NODES ON LINK
           GLOBAL OUTPUT              -
```

Specific Input/ Output Summary

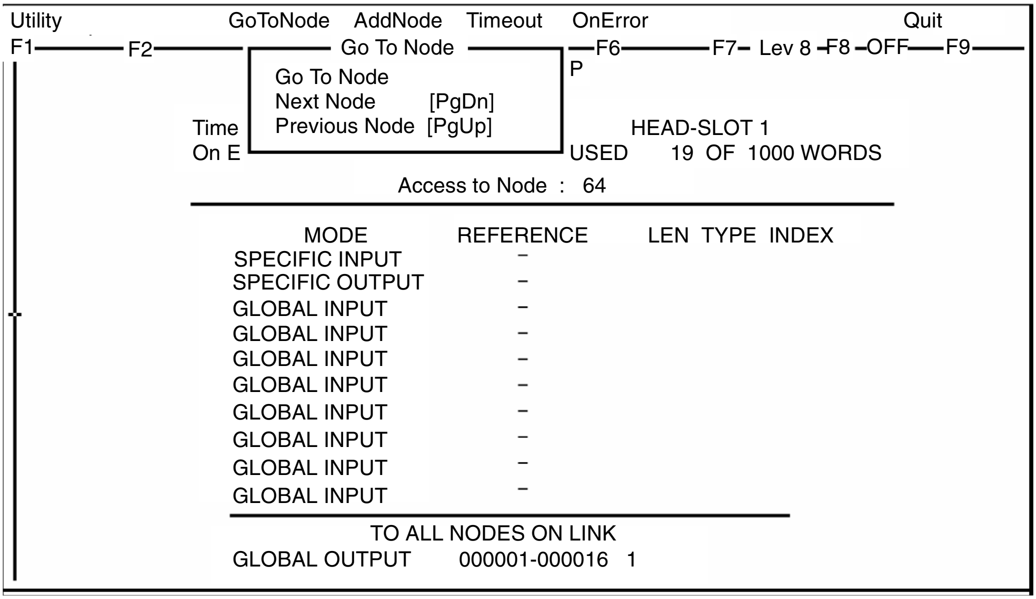
Given the proper configuration, if you complete the template (CRX figure above) for node 2 with a specific input of length of 1 and Specific output length of 1, the results are as follows:
- reading of 1 word from node 2, destination of data is 400100
- writing of 1 word to node 2, source of data is 400050

Other Selectable Support Functions

In addition to the Standard Utility Menu line entry, you have Peer Cop related functions available from the following entries:
- GoToNode
- AddNode
- Timeout
- OnError

The GoToNode Function

GoToNode - Displays the Peer Cop menu that allows you to configure that node. As shown in the following figure, this function has a pulldown. If you select GoToNode and the node number that you enter is not found, you are asked if it should be created for you. You can also transverse the node structure using the PgUp and PgDn keys.



The AddNode Function

This is very similar to GoToNode in that you select the Link and Node number for which you want to add parameters.

The Timeout Function

This field allows you to specify a value for the Health Time–out interval. The default value is 500 Milliseconds. You can change it to any value in the range of 60ms. to 2 seconds. The value that you use specifies the minimum time period that a Peer Cop configured communication must fail before the associated health bit is cleared. You should choose values in 20 ms. increments to account for implementation latency, i.e., the configured time plus the time to assure that the health bit is cleared.
For example, if your choice is 60 ms. the health bit is cleared no sooner than 60 ms. and no later than 79 ms. after communication has been lost.

Health Bits

There is a health bit for each Peer Copped node. If Peer Cop Data is successfully communicated within the set timeout, the associated bit is set to 1. Otherwise, it is set to 0 and all data associated with that group is cleared (to 0). You must use the MSTR element with proper sub-function code (0009) to retrieve the peer cop health information. (See *Peer Cop Health MSTR Operation, p. 84* .)

Note: All configured Specific output health bits are initialized to 1 for the first few scans to allow complete synchronization between controller, health bit time factor, and line latency.

OnError Function

You have the choice of Clearing (CLEAR) the last set of received values or retaining the last set of received values (HOLD) if any error is detected.

Note: DelNode - Once Deleted, you can re-enter node information, or you can exit. Exit with the node deleted removes it. If the DelNode is selected and the Key Verification UPF entry is selected, you are prompted to confirm the intent to clear the node. The default will be "N" for NO. Pressing "Y" for YES and Enter will perform the clear.

Device IP Address Generation

The IP addresses of the I/O devices in the Scan Table are calculated from the Modbus Address entered in the Peer Cop Configuration Extension as well as the IP address of the NOE. Currently, the I/O devices are required to be on the same subnet as the NOE. The device's IP address is calculated by AND'ing the NOE's IP address with the NOE's subnet mask, and then OR'ing the result with the devices's MB address from the configuration extension table.
The following example illustrates the device IP generation.

NOE IP Address:	AAA.BBB.CCC.DDD
Subnet Mask:	255.255.255.0
Device's Modbus Address from Configuration Extension (Range of 1... 64):	MB
Resulting Device IP Address:	AAA.BBB.CCC.MB

Embedded Web Pages

7

At a Glance

Introduction

This chapter presents the contents of the embedded Web pages contained in the Quantum 140 NOE 771 xx modules. These Web pages enable you to access diagnostic information, view configuration information, and change the online configurations for the module.

What's in this Chapter?

This chapter contains the following topics:

Accessing the Web Utility Home Page

Overview

The following information describes how to access the Web utility home page to perform diagnostics and online configuration on the Quantum 140 NOE 771 xx 10/100 Megabit Ethernet module.

Introduction

Each Quantum 140 NOE 771 xx 10/100 Megabit Ethernet module contains an embedded Web server that allows you to access diagnostics and online configurations for the module and its associated controller (PLC).

Pages on the embedded Web site display the following information:

- Configurable menus of the Address Server both BOOTP and DHCP and for SNMP
- Ethernet statistics for the node
- Controller's configuration (**Controller Status** on menu)
- Controller's register values
- Remote I/O status and configuration
- Remote I/O register values
- Remote I/O distributed values

In addition to the pages listed above the 140 NOE 771 1x, FactoryCast / Real Time, modules offer these additional pages:

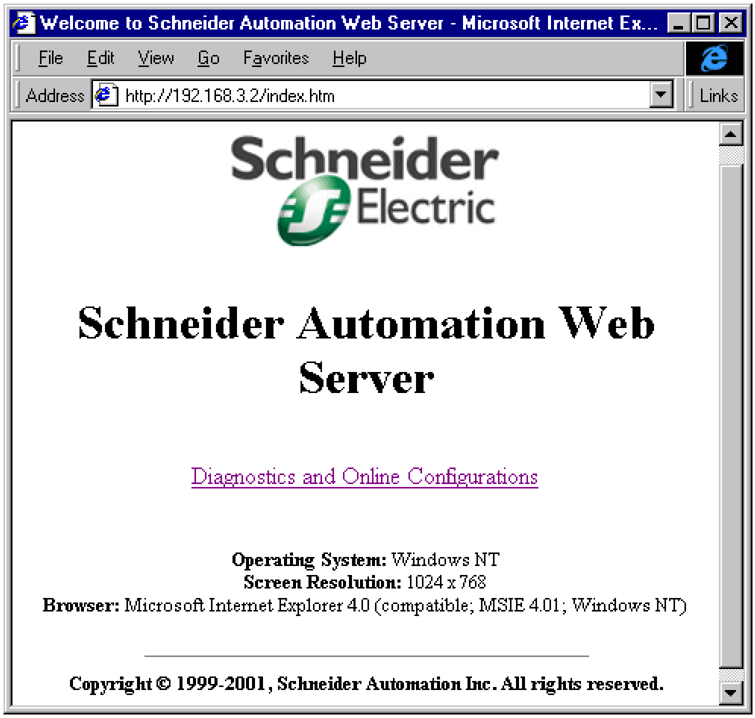
- Configuration and status for Global Data (Publish / Subscribe)
- Bandwidth monitoring
- I/O Scanner status
- MODBUS Messaging status

The Web pages can be viewed with a browser using version 4.0 or greater of either Netscape Navigator or Internet Explorer, both of which support JDK 1.1.4 or higher.

For information about the additional functionality provided by the FactoryCast system in the 140 NOE 771 1x modules, see the *FactoryCast Manual*, 890 USE 152 0X.

Accessing the Module's Home Page

Before you can access the module's home page, you must learn its full IP address or URL from your system administrator. Type the address or URL in the Address or Location box in the browser window. After you do this the Schneider Automation Web Utility home page displays. The following steps describe how to access the Schneider Web Home Utility page.

Step	Action
1	Refer to the following figure, and click **Diagnostics and Online Configurations**. 
2	You will be requested to supply a user name and password.
3	After supplying the user name, password, and clicking **OK**, the Quantum Welcome Page appears. The Quantum Welcome Page provides the links to all the Quantum configuration and diagnostic pages and to the **Run-Time Data Editor**. **NOTE:** The default User Name is USER, and the default password is USER. Both should be changed by the system administrator during module installation.

Quantum Welcome Page

Overview

The following information describes the Quantum Welcome Page.

Quantum Welcome Page Overview and Links

The Quantum Welcome Page provides links to all the Configuration and Diagnostic Pages and to the Run-Time Data Editor.
The following figure shows the Quantum Welcome Page.

Schneider Electric

Web Server for Quantum

Home
Configured Local Rack
Controller Status
Ethernet Statistics
RIO Status
Data Monitor

Configure NOE
NOE Properties
NOE Diagnostics
Support

The following table details the links on the Quantum Welcome Page. To view the pages related to a particular topic, click the link for that topic.

Link	Results
Home	Return to the home page
Configured Local Rack	Displays the Quantum Local Rack with NOE and CPU
Controller Status	Displays the CPU Configuration
Ethernet Statistics	Displays the Ethernet Module Statistics with the Reset Counters link
RIO Status	Displays the Remote I/O Communications Status
Data Monitor	Allows access to the Quantum PLC Data
Configure NOE	Provides the ability to configure and change the NOE through the Ethernet Configuration page
NOE Properties	Provides information about the NOE properties
NOE Diagnostics	Displays the links to Ethernet Statistics and the Crash Log File Diagnostics
Support	Displays contact information for technical assistance, sales, and feedback

Quantum Local Rack Page

Quantum Local Rack Page Overview

The Quantum Local Rack page displays a visual representation of the current configuration.
The following table details the links on the Quantum Local Rack Page. To view the pages related to each of these topics, click on the topic.

Link	Results
Home	Displays the Quantum Welcome Page
Controller Status	Displays the CPU Configuration
Ethernet Statistics	Displays the Ethernet Module Statistics with the Reset Counters link
RIO Status	Displays the Remote I/O Communications Status
Data Monitor	Allows access to the Quantum PLC Data with editing capabilities

CPU Configuration Screen Page

Overview

The following information describes the CPU Configuration Screen Page.

Introduction to the CPU Configuration Screen Page

In the following figure, the top eight fields identify the CPU Configuration. More information about these fields follows.

CPU CONFIGURATION SCREEN

Status:	Stopped	Reference:	140-CPU-113-02
Battery:	OK	Product Type:	Quantum
Rack:	1	Exec ID:	871
Slot:	3	Logged In:	No

DESCRIPTION		REGISTERS		ASCII	
System Memory[Kb]	8 Kb	0xxxxx	000001-001536	Total Words	0
Extended Memory[Kb]	0	1xxxxx	100001-100512	Total Messages	0
Total Memory [Bytes]	8192	3xxxxx	300001-300048	Words Used	0
I/O Map Words	512	4xxxxx	400001-401872	Messages Used	0
Segments	1	6xxxxx	None	Available Words	0
DCP Drop ID	0	Battery Coil	0-----	Available Messages	0
Memory Protect	Off	Timer Register	4-----	# ASCII Ports	0
Constant Sweep	Off	Time of Day Clock	4-----	ASCII Inputs	4-----
Optimize	No	Stopped Codes	0x0200	ASCII Outputs	4-----

Home | Configured Local Rack | Ethernet Statistics | RIO Status | Data Monitor

Description Fields

The following table describes the Description fields on the CPU Configuration Screen Page.

Field	Information Supplied
System Memory [Kb]	Amount of system memory used
Extended Memory [Kb]	Amount of Extended Memory used
Number of I/O words mapped.	Total memory used in bytes
I/O Map Words	Number of I/O words mapped.
Segments	Number of segments
DCP Drop ID	Drop number for Distributed Control
Memory Protect	Position of the Memory Protect Switch
Constant Sweep	Current status of Constant Sweep
Optimize	Current status of Optimization

Register Fields

The following table describes the Register fields on the CPU Configuration Screen Page.

Field	Information Supplied
0xxxxx	Valid Address of 0x
1xxxxx	Valid Address of 1x
3xxxxx	Valid Address of 3x
4xxxxx	Valid Address of 4x
6xxxxx	Valid Address of 6x
Battery Coil	Address of Battery Coil
Timer Register	Address of Timer Register
Time of Day Clock	Address of Timer of Day Clock
Stopped Codes	Reason for controlled stopping

ASCII Fields

The column headed ASCII fields on the CPU Configuration Screen contains information concerning the ASCII fields.

CPU Configuration Screen Page Links

The following table describes the links on the CPU Configuration Screen Page.

Link	Results
Home	Displays the Quantum Welcome Page
Configured Local Rack	Displays the Quantum Local Rack with NOE and CPU
Ethernet Statistics	Displays the Ethernet Module Statistics with the Reset Counters link
RIO Status	Displays the Remote I/O Communications Status
Data Monitor	Allows access to the Quantum PLC Data with editing capabilities

Ethernet Module Statistics Page

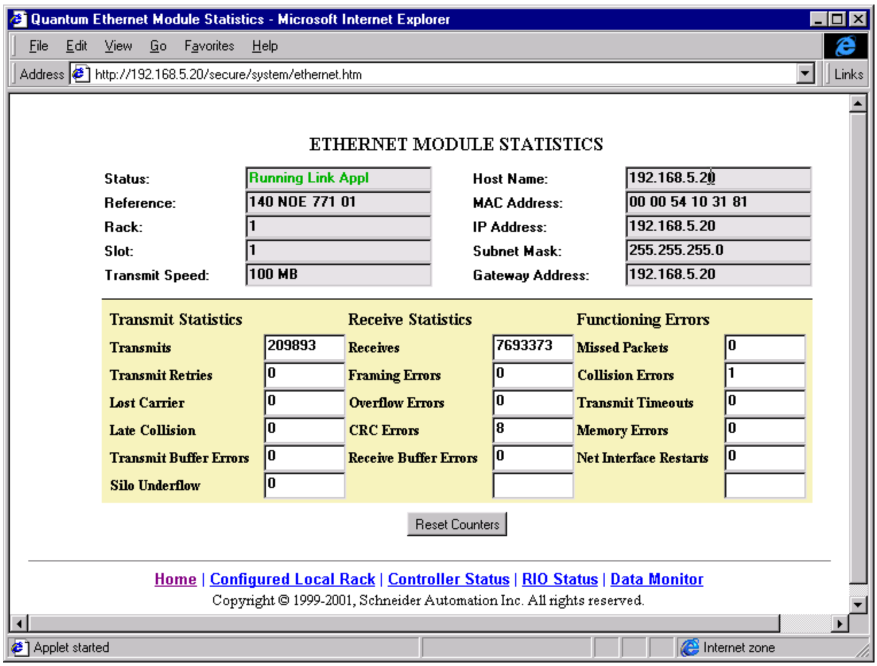
Overview

The following information describes the Ethernet Module Statistics Page.

Introduction to the Ethernet Module Statistics Page

The statistics on Ethernet Module Statistics Page are "information only". To retain any of the information appearing on the Ethernet Module Statistics Page, you must copy it offline (for example, to your hard drive).

The following figure shows the Ethernet Module Statistics Page. The counters may be reset to zero by clicking the Reset Counter button.



Ethernet Module Statistics Page Links

The following table describes the links on the Ethernet Module Statistics Page.

Link	Results
Home	Displays the Quantum Welcome Page
Configured Local Rack	Displays the Quantum Local Rack with NOE and CPU
Controller Status	Displays the CPU Configuration
RIO Status	Displays the Remote I/O Communications Status
Data Monitor	Allows access to the Quantum PLC Data with editing capabilities

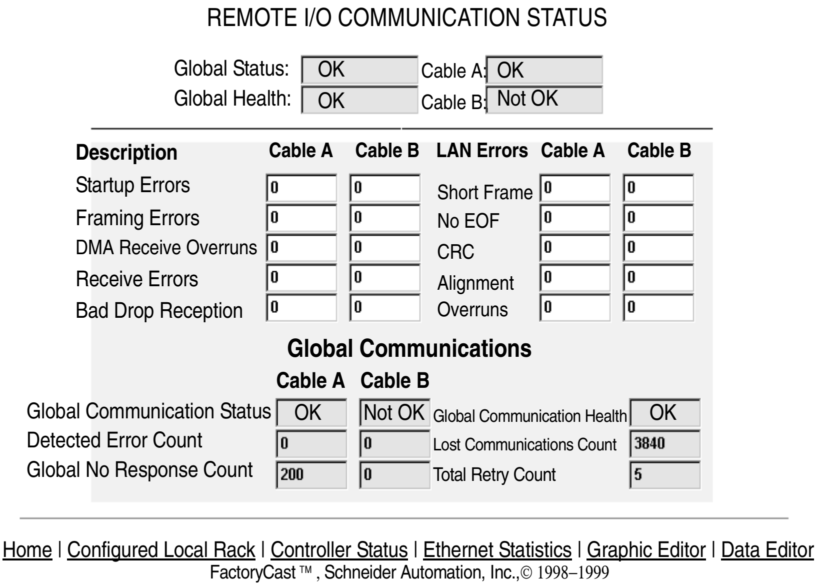
Remote I/O Communication Status Page

Overview

The following information describes the Remote I/O Communication Status Page.

Introduction to the Remote I/O Communications Status Page

In the following figure, the fields are "information only". You cannot change the fields.



> **Note:** The Graphic Editor Link is available only on the 140 NOE 771 1x, the FactoryCast Web server.

For more information about the Remote I/O Communications Status, see the RIO Manual, 890 USE 101 00.

Remote I/O Communications Status Page Links

The following table describes the information that is available for each cable, using the links on the Remote I/O Communication Status Page.

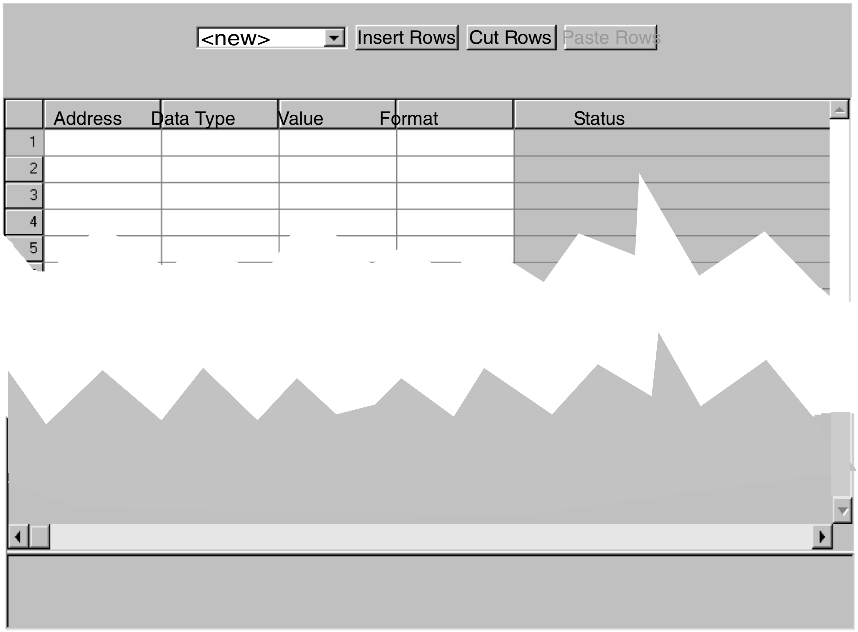
Link	Results
Home	Displays the Quantum Welcome Page
Configured Local Rack	Displays the Quantum Local Rack with NOE and CPU
Controller Status	Displays the CPU Configuration
RIO Status	Displays the Ethernet Module Statistics with the Reset Counters link
Data Monitor	Allows access to the Quantum PLC Data with editing capabilities

Quantum PLC Data Monitor Page

Introduction to the PLC Data Monitor Page

The following figure shows the web page that allows you to display Quantum PLC data.

Quantum PLC Data Monitor



Home | Configured Local Rack | Controller Status | Ethernet Statistics | RIO Status

You can add, delete, and copy Quantum PLC data as described in the following list:
- You can insert additional rows of data by clicking on the Insert Rows button.
- You can delete specific rows of data by clicking on the Cut Rows button.
- You can copy in rows of data by clicking on the Paste Rows button.

PLC Data Monitor PageLinks

The following table describes the links on the Quantum PLC Data Monitor Page.

Link	Results
Home	Displays the Quantum Welcome Page
Configured Local Rack	Displays the Quantum Local Rack with NOE and CPU
Controller Status	Displays the CPU Configuration
RIO Status	Displays the Remote I/O Communications Status

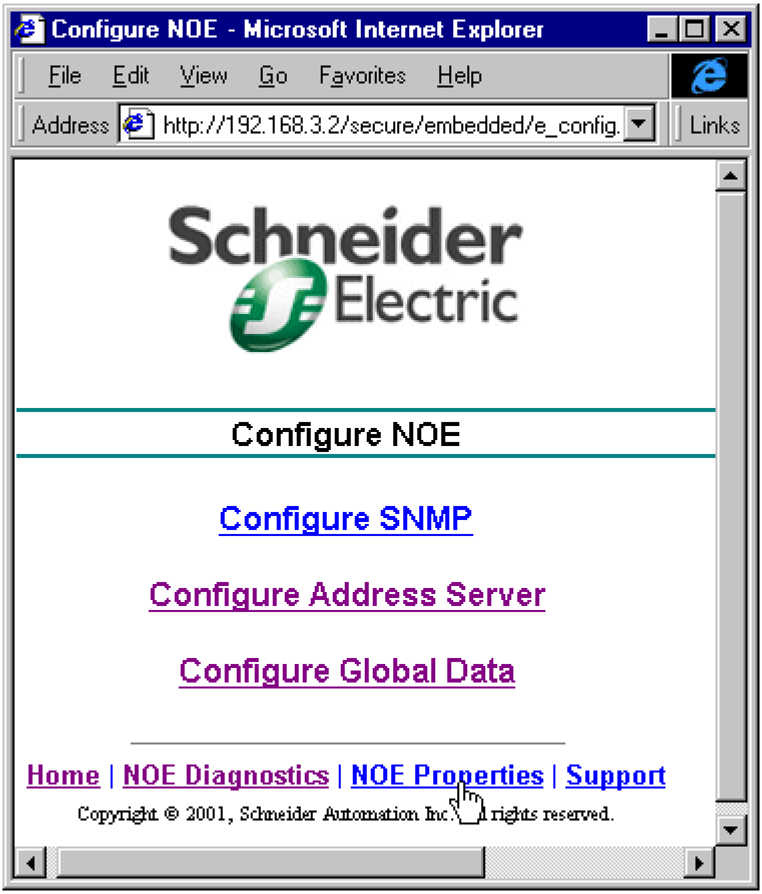
Configure NOE Page

Overview

The following information describes the **Configure NOE** page and the links on that page.

Introduction to the NOE Configuration Pages

The following figure shows the **Configure NOE** page, which provides links to the individual configuration pages for the NOE.



NOE Configuration Page Links

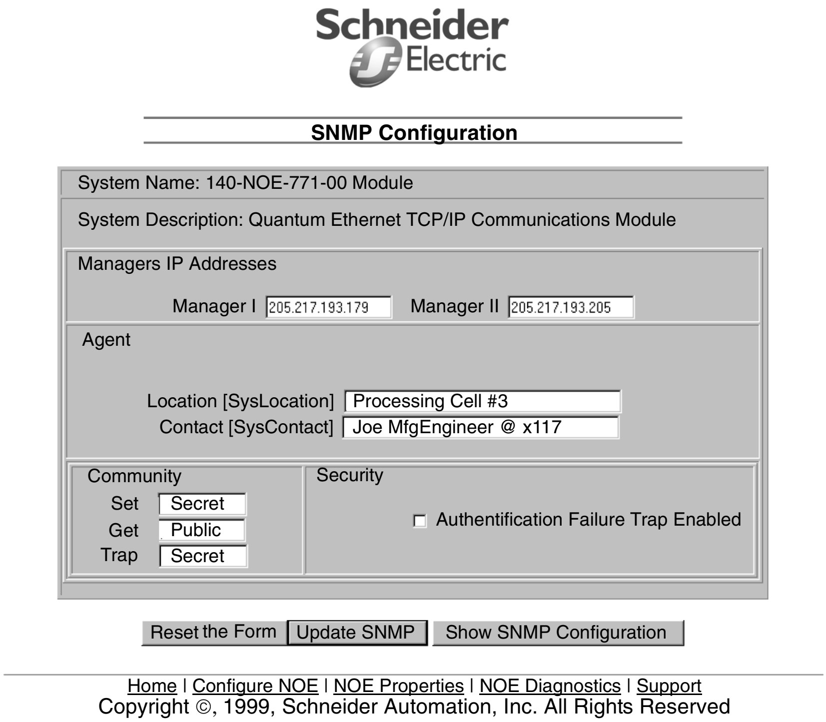
The following table describes the links on the **Configure NOE** page.

Link	**Results**
Home	Returns you to the Quantum Welcome Page
Configure SNMP	Provides the ability to configure the SNMP Agent in the NOE
Configure Address Server	Provides the ability to configure the BOOTP IP assignments, including showing the BOOTP and DHCP database
Configure Global Data	Displays the Global Data Configuration page and provides information about the Group Address, Multicast filtering, Distribution period, Health Time Out, Health Bits, and Data Zones. The Global Data Configuration page also displays a Variable Table.
NOE Properties	Provides information about NOE properties
NOE Diagnostics	Displays links to Ethernet Statistics and the Crash Log File Diagnostics
Support	Provides you with contact information for technical assistance, sales, and feedback

Configure SNMP Page

Introduction to the Configure SNMP Page

SNMP may already be configured on your NOE 771xx. If it is not, complete the SNMP Configuration form, which is shown in the following figure.



The following table shows how to enter the required information for configuring SNMP on your NOE 771 xx.

Task	How To
To display the current SNMP configuration	Click **Show SNMP Configuration**
To clear the fields	Click **Reset the Form**
To change the SNMP configuration	Change the information on the page and click **Update SNMP**

SNMP Page Fields

The following table describes the specific SNMP fields that you can modify.

Field	Information To Be Supplied
Manager I	IP Address of first SNMP Manager
Manager II	IP Address of second SNMP Manager
Location [SysLocation]	Location of the module
Contact [SysContact]	Name of the responsible systems engineer
Set	Designation of level of user who can set the configuration
Get	Designation of level of user who can view the configuration
Trap	Designation of level of user who can capture information
Authentication Failure Trap Enabled	Turns on Community Name Checking

Configure SNMP Page Links

The following table describes the links on the Configure SNMP Page.

Link	Results
Home	Returns you to the Quantum Welcome Page.
Configure NOE	Provides the ability to configure and change the NOE through the Ethernet Configuration Page.
NOE Properties	Provides information about NOE properties
NOE Diagnostics	Displays links to Ethernet Statistics and the Crash Log File Diagnostics
Support	Provides you with contact information for technical assistance, sales, and feedback

Completion Message

Clicking on the **Update SNMP** button displays a new page containing the message "Successfully updated SNMP database". Note that this page contains the same links as those on the Configure SNMP Page.

Note: The NOE module has to be reset for the changes to take effect.

Configure Address Server Page

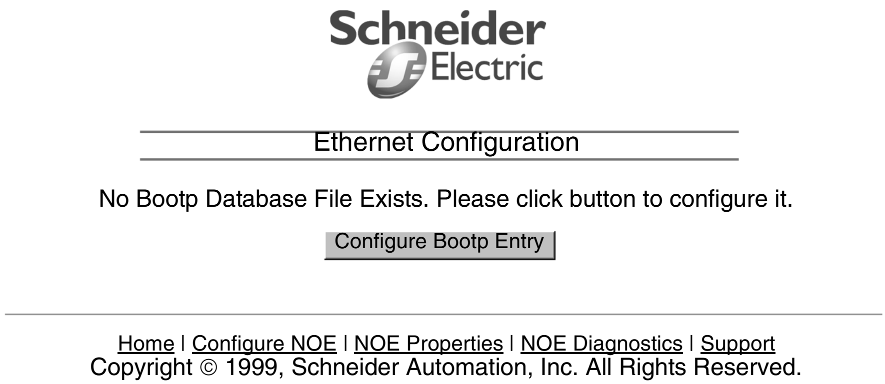
Overview

The following information describes how to configure the Address Server for the 140 NOE 771 x0, Transparent Factory modules.

Note: For information describing how to configure the Address Server for the 140 NOE 771 x1, FactoryCast Web server modules, see the chapter entitled *Address Server Configuration / Faulty Device Replacement*. That chapter describes the DHCP process.

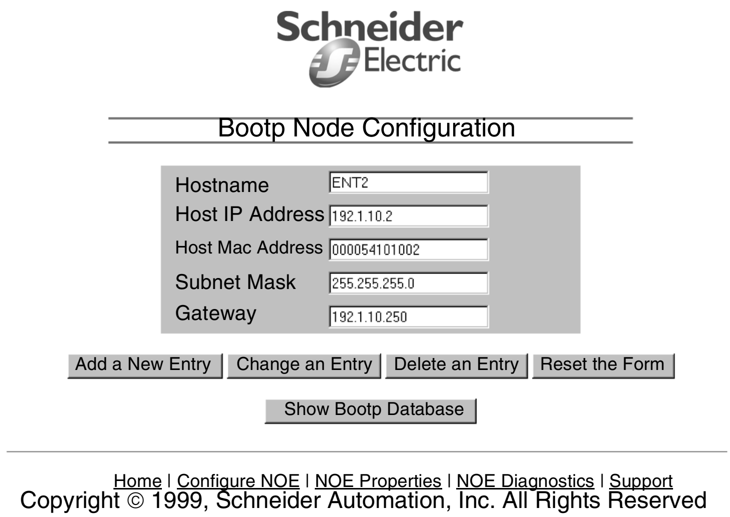
Introduction to the Configure BOOTP Initial Page

The BOOTP Database File does not exist. Therefore, you need to create the database. The following page is used to create a BOOTP database file. The page below appears for and is used for the 140 NOE 771 x0 modules.



Displaying the Bootp Node Configuration Form Page

The following information describes how to complete the **Bootp Node Configuration** page.

Step	Action
1	Click the **Configure Bootp Entry** button on the **Ethernet Configuration** page to configure a BOOTP Database File. **Result**: As the following figure shows, the **Bootp Node Configuration** page appears. 
2	If this is an initial BOOTP configuration, fill in the fields on the **Bootp Node Configuration** page and click the **Add a New Entry** button.

Initial Configuration

The following table describes the fields to complete on the BOOTP configuration form.

Field	Information to be Supplied
Hostname	Text to identify device
Host IP address	IP Address from System Administrator - read from sticker on device
Host Mac Address	IEEE Global Address
Subnet Mask	Supplied by system administrator and configured on the server by the system administrator
Gateway	Define the address of route to use to access nodes off the devices subnet

Adding to the BOOTP Database File

If you want to add an entry to the BOOTP Database File, complete the fields on the form, and press the **Add a New Entry** button.

Changing the BOOTP Database File

The following steps describe how to change an entry in the BOOTP Database File.

Step	Action
1	Enter the new information on the **Bootp Node Configuration** page
2	Click on the **Change an Entry** button. **Result**: The a new entry will be made at the bottom of the Database Table, and you will get a successful entry message.
3	Click on **Configure NOE** to return to the Configure NOE page.
4	Click on **Configure BOOTP.**
5	Enter the information to be old information.
6	Click on **Delete an Entry.**

Deleting the BOOTP Database File

The following steps describe how to delete an entry in the BOOTP Database File.

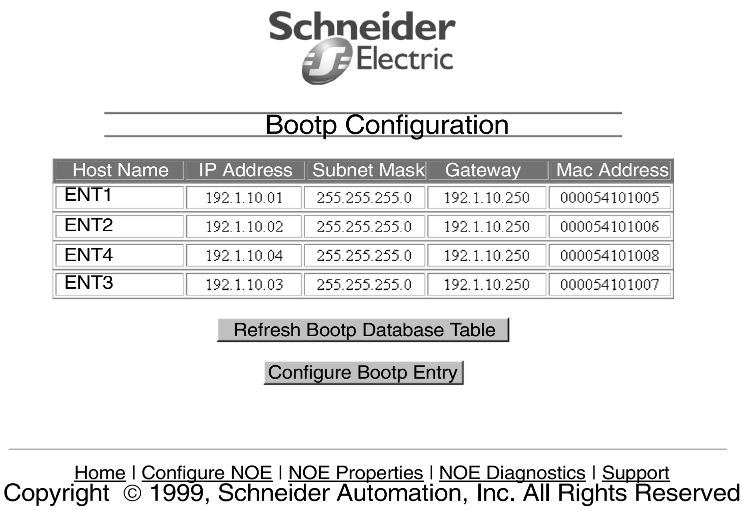
Step	Action
1	Enter the new information for the item to be deleted.
2	Click on the **Delete an Entry** button. **Result:** A delete successful message is displayed.
3	Click on **Configure NOE** to return to Configure NOE page.
4	Click on **Configure BOOTP.**
5	Click on **Refresh Bootp Database Table** to view revised Database file.

Resetting the Form

To clear the fields in the **Bootp Node Configuration** page, press the **Reset the Form** button. You will then be able to fill in the Database File entries information to be added, changed, or deleted.

Displaying the BOOTP Database File

The following figure shows a sample current BOOTP Database File. To display the current BOOTP Database File, press the **Show Bootp Database** button.



Schneider Electric

Bootp Configuration

Host Name	IP Address	Subnet Mask	Gateway	Mac Address
ENT1	192.1.10.01	255.255.255.0	192.1.10.250	000054101005
ENT2	192.1.10.02	255.255.255.0	192.1.10.250	000054101006
ENT4	192.1.10.04	255.255.255.0	192.1.10.250	000054101008
ENT3	192.1.10.03	255.255.255.0	192.1.10.250	000054101007

Refresh Bootp Database Table

Configure Bootp Entry

Home | Configure NOE | NOE Properties | NOE Diagnostics | Support

Configure BOOTP Page Links

The following table describes the links on the **Bootp Node Configuration** page Page.

Link	Results
Home	Returns you to the Quantum Welcome Page
Configure NOE	Provides the ability to configure and change the NOE through the Ethernet Configuration Page
NOE Properties	Provides information about NOE properties
NOE Diagnostics	Displays links to Ethernet Statistics and the Crash Log File Diagnostics
Support	Provides you with contact information for technical assistance, sales, and feedback.

Extended Web Diagnostics Pages

Overview

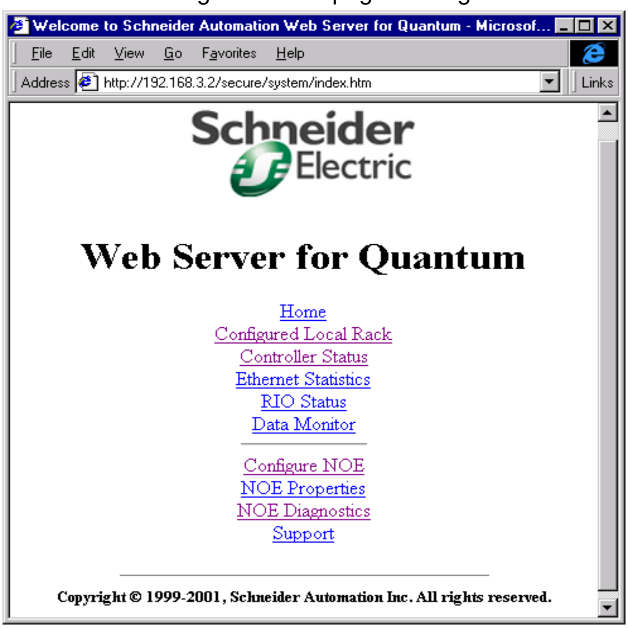
The 140 NOE 771 x1 embedded Web server provides Web pages that you may use to diagnose Transparent Factory / Real Time services. Those services are listed below:

- Global Data Diagnostics
  - Status of all Global Data services
  - Status of all subscribed and published variables
  - Publication / Subscription rate
- I/O Scanning Diagnostics
  - Status of all I/O Scanning services
  - Status of individual scanned devices
  - Actual I/O scanning rate
- Messaging Diagnostic
  - Diagnostic information for Modbus (port 502) messaging
- Bandwidth Monitoring
  - Throughput measurement of NOE by service

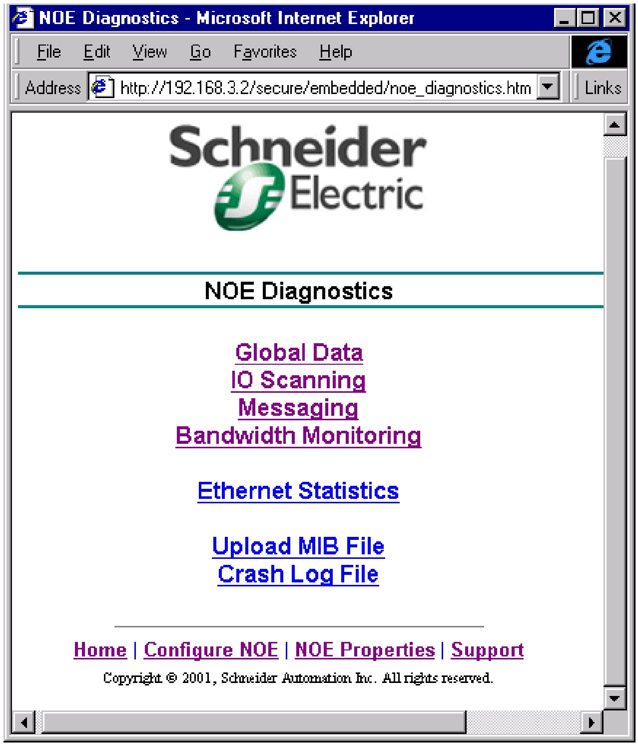
Note: All these pages are protected by the general HTTP password.

Accessing Web Diagnostics

You access the diagnostic Web pages through the **Web Server for Quantum** page.



On this page select the **NOE Diagnostics** link, which opens the **NOE Diagnostics** page.
On the **NOE Diagnostics** page select the desired service by clicking the appropriate link.

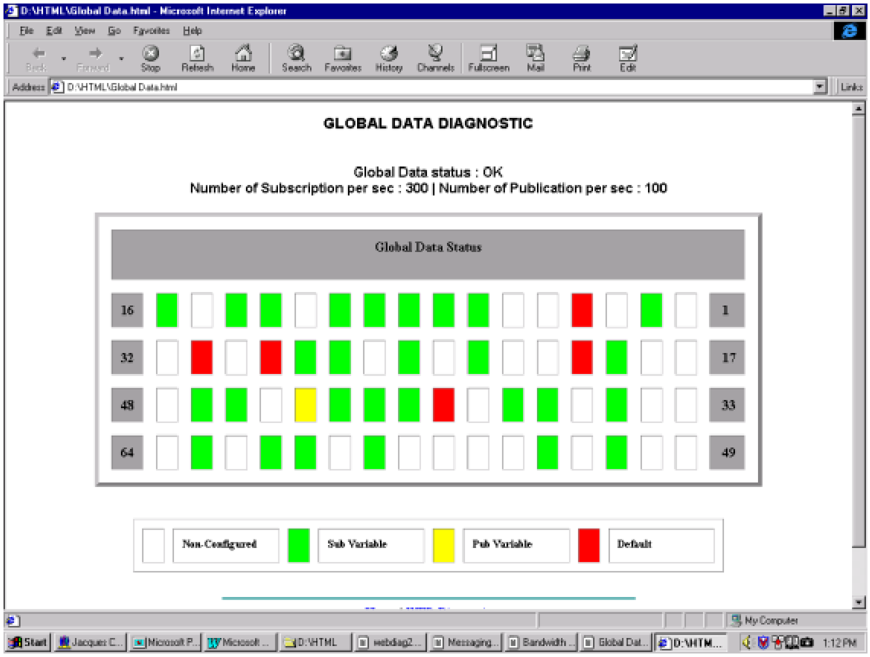


The first four links are the diagnostics.
- Global Data
- IO Scanning
- Messaging
- Bandwidth Monitoring

The last three links access other functions.
- Ethernet Statistics
- Upload MIB File
- Crash Log File

Global Data Diagnostics Page

This is the **Global Data Diagnostics** page.



This page displays information generated by the Global Data service. At the top of the page the following three items appear:

- Global Data status
- Number of subscriptions per second
- Number of publications per second

The above information is displayed both as text and as a graphic.

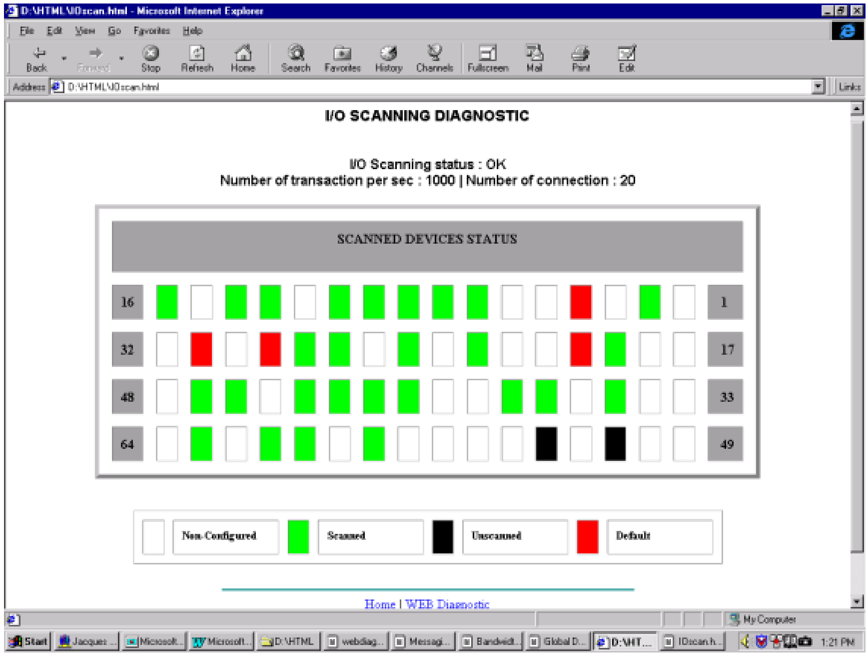
This page also displays the complete status for all variables published and subscribed within the same distribution group. Every variable is identified by its data ID. At the bottom of the page four boxes display indicating the status of the variables. Those boxes are color coded.

- White for all non configured variables
- Green for the healthy subscribed variables
- Black for the healthy published variables
- Red for a communication fault

If Red displays in the **Fault** variable box, you should check the system for problems. The **Global Data status** indicator displays **OK**, even if there are unhealthy variables.

I/O Scanning Diagnostic Page

This page displays information generated by the I/O Scanning service.



At the top of the page the following three items appear:
- I/O Scanning status
- Number of transactions per second
- Number of connections

The above information is displayed both as text and as a graphic.
At the bottom of the page four boxes display indicating the status of the devices.
Those boxes are color coded.
- White for all non configured devices
- Green for all scanned devices
- Black for all temporary unscanned devices
- Red for all devices in a default state

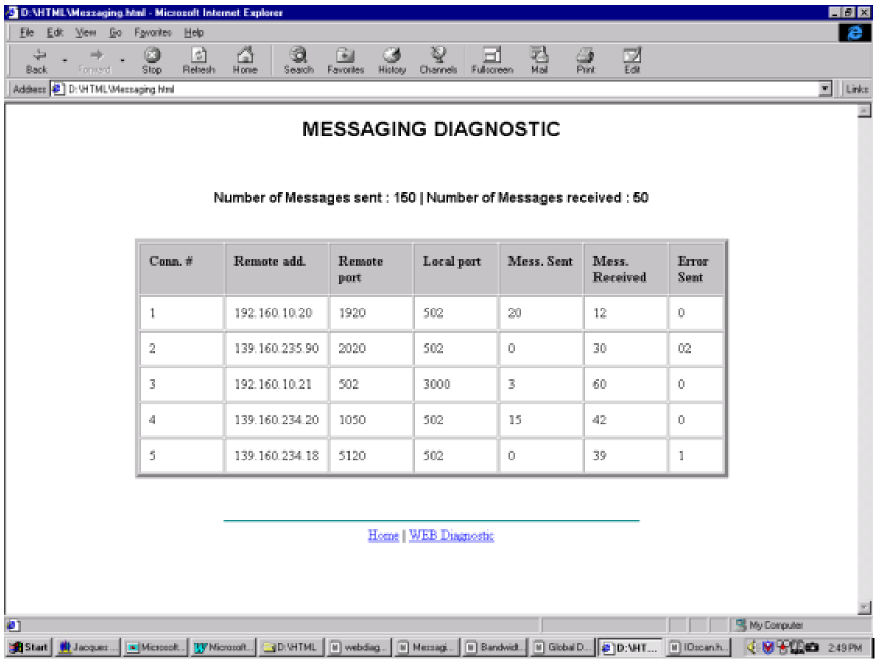
If Red displays in the **Fault** variable box, you should check the system for problems

Messaging Diagnostic Page

This page provides information concerning current open TCP connections on Port 502. The total number of messages sent and received on Port 502 displays on the top of this page.

> **Note:** Please note the following.
> - After a port 502 connection is closed, the connection will remain on the list for several minutes.
> - The total message counter is not reset after a port 502 connection is closed.
> - The I/O status indicator displays OK, even if there are unhealthy faulted scanned devices.

The **Messaging Diagnostic** page:



MESSAGING DIAGNOSTIC

Number of Messages sent : 150 | Number of Messages received : 50

Conn. #	Remote add.	Remote port	Local port	Mess. Sent	Mess. Received	Error Sent
1	192.160.10.20	1920	502	20	12	0
2	139.160.235.90	2020	502	0	30	02
3	192.160.10.21	502	3000	3	60	0
4	139.160.234.20	1050	502	15	42	0
5	139.160.234.18	5120	502	0	39	1

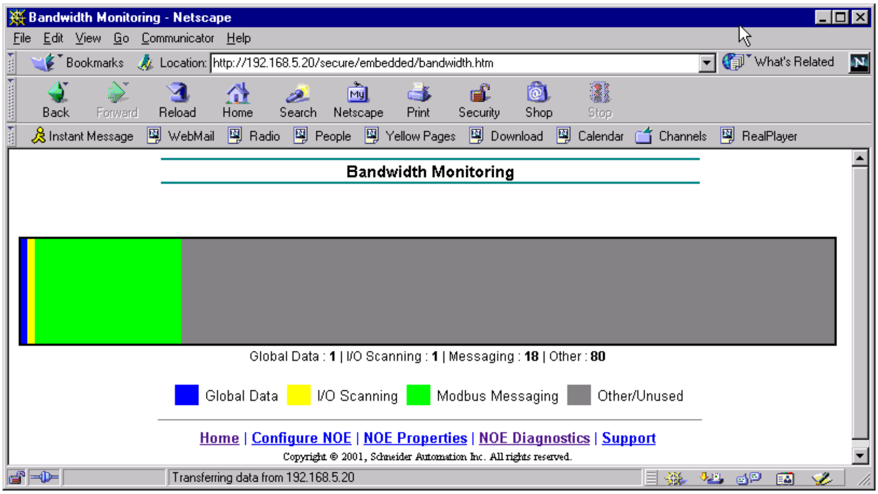
Home | WEB Diagnostic

For each connection (numbered from 1 to 64) the following information is given:
- Remote address (IP address)
- Remote port (TCP)
- Local port (TCP)
- Number of messages sent (**Mess. Sent**) on this connection
- Number of messages received (**Mess. Received**) on this connection
- Number of errors (**Error Sent**) on this connection

Bandwidth Monitoring Page

The Bandwidth Monitoring page indicates how the NOE's CPU is shared between the Global Data service, the I/O Scanner service, and the Messaging service. The information generated by this service displays both as a graphic and as text. The information generated gives you a view of the allocation of service usage. Knowing the allocation of services can help you to determine the number of and the distribution of NOEs on your system.

The **Bandwidth Monitoring** page:



The page reports four statistics one each for the three services and one for "other."
- Blue: Global Data (usage expressed as a % of maximum number of messages/second)
- Yellow: I/O Scanner (usage expressed as a % of maximum number of messages/second)
- Green: Modbus Messaging (usage expressed as a % of maximum number of messages/second)
- Gray: Other/Unused (usage expressed as a % of maximum number of messages/second.)

Percentages add to 100%.

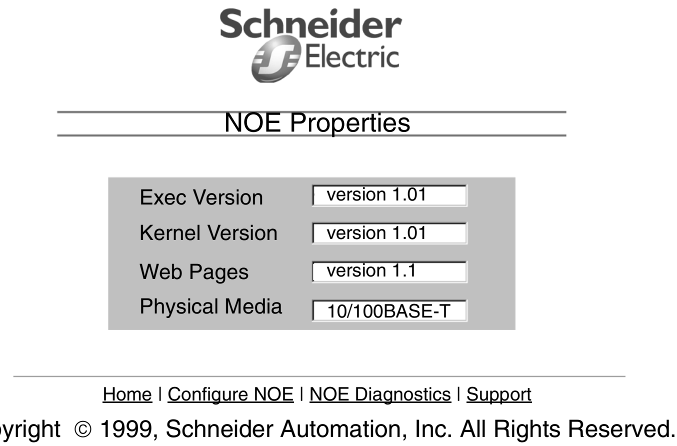
NOE Properties Page

Overview

The following information describes the NOE Properties Page.

Introduction to the NOE Properties Page

The following figure shows the NOE Properties Page, which displays the Exec, Kernel, Web Pages versions, and the Physical Media being used.



Note: The NOE Properties Page is "information only." The fields cannot be changed.

NOE Properties Page Links

The following table describes the links on the NOE Properties Page.

Link	Results
Home	Returns you to the Quantum Welcome Page
Configure NOE	Provides the ability to configure and change the NOE through the Ethernet Configuration Page
NOE Diagnostics	Displays links to Ethernet Statistics and the Crash Log File Diagnostics
Support	Provides you with contact information for technical assistance, sales, and feedback

NOE Diagnostics Page

Overview

The following information describes the NOE Diagnostics Page.

NOE Diagnostics Page Links

The following table describes the links on the NOE Diagnostics Page.

Link	Results
Home	Returns you to the Quantum Welcome Page
Ethernet Statistics	Displays the Ethernet Module Statistics Page where you can display the Ethernet statistics and reset the counters
Crash Log File Diagnostics	Displays the Crash Log entries for use in diagnosing the cause of crashes.
Configure NOE	Provides the ability to configure and change the NOE through the Ethernet Configuration Page
NOE Properties	Provides information about NOE properties.
Support	Provides you with contact information for technical assistance, sales, and feedback

Crash Log Diagnostics

Introduction to the Crash Log Diagnostics Page

The Crash Diagnostics Page displays a crash log file when a crash has occurred, and a status message when no crash has occurred.
Press **Clear Crash Log File** to clear the file.

Crash Log Diagnostics Links

The following table describes the links on the Crash Log Diagnostics Page.

Link	Results
Home	Returns you to the Quantum Welcome Page
Configure NOE	Provides the ability to configure and change the NOE through the Ethernet Configuration Page
NOE Properties	Provides information about NOE properties
NOE Diagnostics	Displays links to Ethernet Statistics and the Crash Log File Diagnostics
Support	Provides you with contact information for technical assistance, sales, and feedback

Contacting Schneider Automation Page

Overview

The following information describes the Contacting Schneider Automation Page.

Introduction to the Contacting Schneider Automation Page

The following figure shows the Contacting Schneider Automation Page, which contains information about how to obtain support for the NOE 771 xx modules.


Schneider Electric

Contacting Schneider Automation

Merlin Gerin
Modicon
Square D
Telemecanique

There are numerous ways to reach us for assistance:

Technical Information
Click here to go to the Schneider Automation web site

Technical Assistance
If you need technical assistance with a product or service, contact us by email at *customercentral@schneiderautomation.com*
or telephone us at 1-800-468-5342 or 1-978-975-9700

Note: Be sure to supply your name, telephone number, company name and address within your email to assure an immediate response.

Feedback
Thoughts, comments, ideas about our site? Please let us know by contacting us at *feedback@modicon.com.*

U.S. Sales Offices
Click here to look up a location of a Sales Office in the U.S.

Home | Configure NOE | NOE Properties | NOE Diagnostics

Address Server Configuration / Faulty Device Replacement

8

At a Glance

Introduction

This section covers the Address Server Configuration / Faulty Device Replacement service available only on the NOE 771 -01 and -11, Transparent Factory / Real Time modules. The Faulty Device Replacement service offers you a method of handling device replacement without disrupting the system nor interrupting service.

What's in this Chapter?

This chapter contains the following topics:

Address Server Configuration / Faulty Device Replacement

Overview

The Address Server provides 2 capabilities:

1. Standard DHCP Server Behavior
   Enter the MAC Address and IP Configuration. The NOE DHCP server will provide the IP configuration when the device sends a DHCP request.
2. Faulty Device Replacement Behavior
   Enter both a Role Name and the IP Configuration of the device. The device will send the Role Name with its DHCP request. With the DHCP response from the NOE, the device will receive its IP Configuration, plus the name and location of a configuration file.
   The next step for an FDR-compliant device is to download it's configuration from the NOE.
   Consult your Schneider Automation Sales Representative for the current list of FDR-Compliant devices.

The Address Server in the NOE supports both modes at the same time. You select a mode by either inputting the Mac Address or the Role Name in the **Address Server Node Configuration** page. You may enter only one or the other, but not both.

The Faulty Device Replacement capability allows automatic configuration of FDR-compliant devices.

Identifying a Role Name

Inherent in the discussion of Faulty Device Replacement is the idea of a Role Name. A Role Name is a logical name that you assign to a device, a logical name that has meaning in your application.

Some Examples of a Role Name might be:

- **ENT_6** (6th Momentum ENT in your application)
- **OUTPUT_VALVE_2** (2nd Output Valve in your application)

Role Names are case sensitive.

Faulty Device Replacement

The Faulty Device Replacement service offers you a method of handling device replacement without disrupting the system nor interrupting service. Should a device fail, replacing that device is easy. When the new device is physically connected to the network, the system (including the new device) is able to
- Provide the replacement device with the IP address of the previous device
- Ensure that new device is functioning in the same manner as the previous device
- Restore the I/O device application parameters in order to restart the device with the same configuration as before the failure

Faulty Device Replacement enables you to avoid configuring a new device when a faulty device is replaced: You enter the device name in the new device, and the task is completed. You have a new configuration scheme for I/O and smart devices, which allows:
- Creating an automatic network configuration
- Managing automatic application parameters

Faulty Device Replacement is implemented using the combination of DHCP and FTP/TFTP standard protocols. The device implements a DHCP client and an FTP or TFTP client. Choosing between FTP and TFTP has no direct impact on your system. The choice depends only on device memory footprint: TFTP is much smaller than FTP.

Faulty Device Replacement offers the following functionality
- FDR Manager
- FDR Agent

Faulty Device Replacement management is based on three entities:
- Agent device embedding the DHCP client and FTP/TFTP client
- DHCP server
- FTP/TFTP server

Role Name

The Logical Role Name should be written on devices. The technician can then get a new device from stores, enter the corresponding Role Name into the device, and place the device in the system. The device automatically gets its configuration and starts running with no further input from the technician. This process is designed to get your machine up and running quickly. All the technician has to do for any FDR compliant device is enter the Role Name into the replacement device.

Address Server Limits

This table displays the parameters and limits of the Address Server.

Parameter	Limit
Maximum Number of Address Server Entries	128
Maximum Size of Configuration File per Device	4K bytes
Total Size of Faulty Device Replacement Storage	512K bytes
Maximum Role Name Size	16 Characters

Note: For the DHCP Server to function correctly the following must be observed:
- Address class and subnet class configured for the device must match
- Address class of the NOE and of the devices must be the same

Operating on a Corporate Network

Note: Operating on a Corporate Network

- Before placing the NOE on a corporate network, Schneider Automation recommends that you discuss the installation with your MIS department. It is likely that your company's corporate network has at least one DHCP Server running already. If the NOE's DHCP server is running on the same network, it may disturb the network.
- To avoid any possible problem related to the NOE's DHCP server on the corporate network, you must ensure that the DHCP server is not running in the NOE by not having Address Entries in the Configuration. If there are no configured devices in the Address Server Configuration page, then the NOE will not start the DHCP server.

Available FDR Agents

Three FDR agents are available
- Momentum ENT
- Micro ETZ
- ATV58*

*Available 2002

The `role-name.prm` configuration files are stored in the NOE in non-volatile memory. Therefore, after a power failure all configurations will be available.

BOOTP and DHCP Compatible Devices

Use either the MAC address or the Role Name to assign IP addresses. Therefore, you may use the DHCP server with devices that support BOOTP only, such as Momentum ENT v1.

Understanding Faulty Device Replacement

Understanding Confirmation and Error Messages

In addition to highlighting errors the system provides confirming information and error messages.

Confirmation Message If you successfully added, modified, or removed and entry, the following alert message appears.

Error Messages Error messages, appearing as an icon in the seventh column, display on the Address Server Configuration page, or they appear as dialog box.

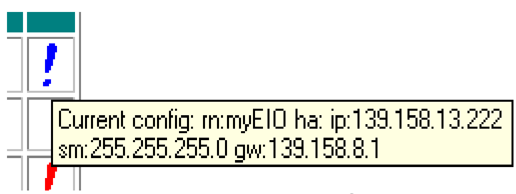
Error Icon If an entry is not loaded in the DHCP server or loaded with a different configuration, an icon of an exclamation point displays in the seventh column. The icon informs you of the difference between the current and stored information.

- Not Loaded in Server: A red icon displays. 
- Duplicate Configuration: A blue icon displays. 

Place the pointer over the icon and a pop-up window appears and displays a message

- Not Loaded in Server:

- Present Configuration:



Error Dialog Box If you entered an existing role name or MAC address, you will receive an alert message asking you to correct the entry.

Modifying the Database

If you need to add or modify an entry in the database, use the **Address Server Node Configuration** page. Three fields must be filled in **Device IP address**, **Subnet Mask**, and **Gateway**.

Choose either the **Role Name** or **Device MAC address** field. When one field is selected, the other is made unavailable.

Adding Entries If you are adding a device, the page appears with values. You need to enter either a Role Name or a MAC Address.

If you are adding an entry, submit your selection using the **Add the Entry** button.

Modifying Entries If you are modifying an entry, the **Device IP address**, **Subnet Mask**, and **Gateway** fields display with the current configuration.

If you are modifying an entry, submit your selection using the **Reset the Form** button.

Each field of the **Address Server Node Configuration** page has restrictions, which are detailed below.

- **Role Name**
  Each role name must be unique. Only letters, numerals, and underscores are allowed. No more than 16 characters are allowed, and no spaces are allowed.
- **Device MAC Address**
  This address must be in hexadecimaL format and six (6) bytes long (6x2 characters). The MAC address can be entered with or without a delimiting character between each pair of lower or upper case hexadecimal characters. The delimiting characters improve readability. There are three choices for a delimiting character:
  - Space " "
    Use the spacebar to create the space.
  - Colon ":"
  - Hyphen "-"
- **Device IP address**
  The IP address must use the standard numeric Internet Protocol address that uniquely identifies each computer on the network. The IP address is a 32-bit identifier made up of four groups of numbers (from 0 through 255), each number separated by a period, for example `123.456.78.9`.
- **Subnet Mask**
  The subnet mask must be entered in IP address format.
- **Gateway**
  The gateway must be entered in IP address format. The gateway must be on the same subnet as the device.

Configuring Faulty Device Replacement

Configuring the Address Server

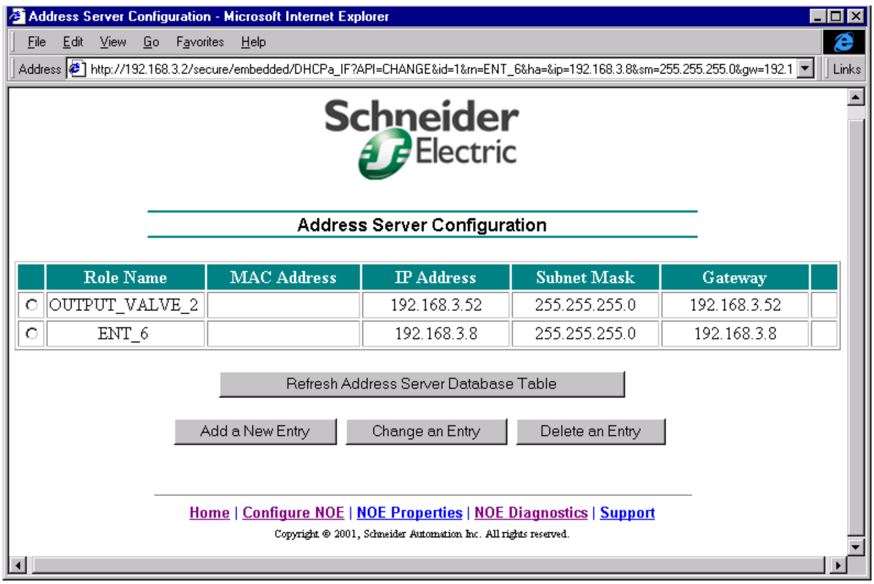
To configure the Address Server you use Web pages generated by the embedded Web server. The first page that appears is the **Address Server Configuration** page. The first column contains buttons used for selecting devices. The **Address Server Configuration** page displays configuration information for each device in the system and has seven columns in the table.

Displayed on this page is information about:

- Role Name
- Mac Address
- IP Address
- Subnet Mask
- Gateway

A additional, unnamed column indicates if there is a difference between the current and the stored configuration. If a difference exists, an exclamation point is displayed.

This is the **Address Server Configuration** page. All devices are compatible.



On the Address Server Configuration page you can:

- Add a New Entry
- Change an Entry
- Delete an Entry

Choosing Options

The Address Server Configuration page allows you to choose different options for adding or altering the configurations of your NOE. The options available to you are:

- Selecting an entry
- Adding an entry
- Changing an entry
- Deleting an entry

Below we describe the method and options chosen to perform any of the four options listed above. Screen shots are presented to accompany the *Adding an entry* section.

Selecting an Entry When the page displays, by default no entries are selected. Use the radio buttons in the first column to select an entry. Only one entry may be selected at a time.

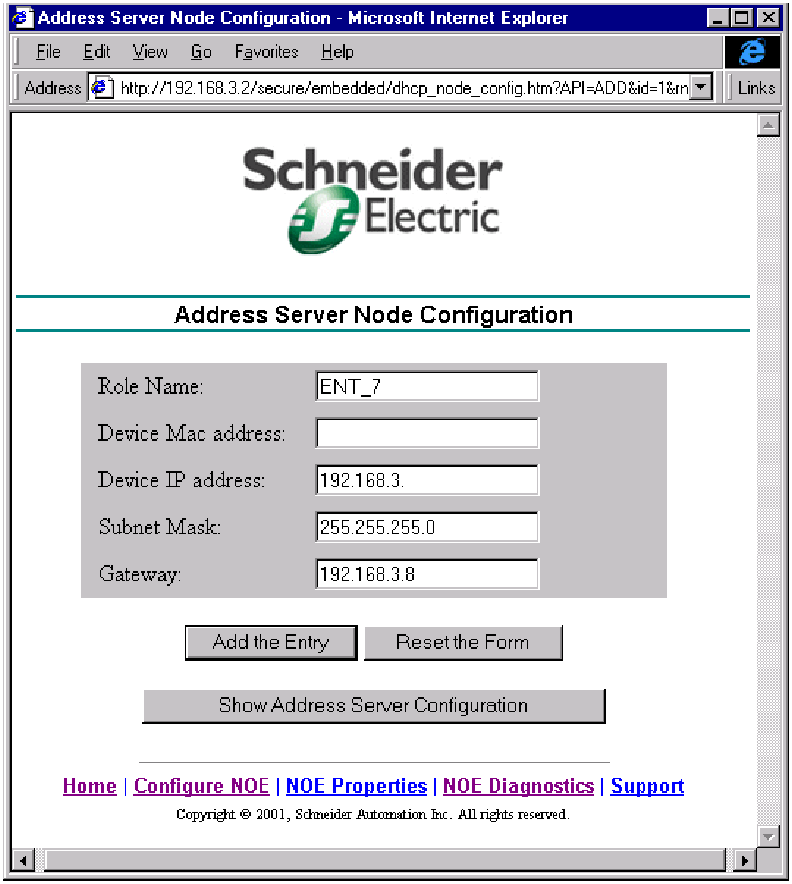
Adding an Entry When the **Add a New Entry** button is selected, the **Address Server Node Configuration** page appears. This page displays information about a device.

If you selected a device, this page displays the device's configuration. Configuration information displays in four of the five fields of the dialog box. Only the Role information field is blank. You should enter a Role name, for example *ENT_7*.

If no device is selected, default values appear in all the fields.

Changing an Entry Before using this button, you must select an entry in the database by choosing one of the radio buttons in the first column. If you fail to choose an entry, an error message appears.

When the **Change an Entry** button is selected, the **Address Server Node Configuration** page appears. The information displayed is for the device selected.



Deleting an Entry Before using this button, you must select an entry in the database by choosing one of the radio buttons in the first column. If you fail to choose an entry, an error message appears.

The entry selected will be removed from the database. Before completely deleting an entry, a warning message appears. Click **Yes** if you want to delete the entry, no if you don't.

If you click **Yes**, a dialog box appears.

Click OK. Another dialog box appears notifying you that the deletion was successful.

Highlighting Errors

If there are problems with the configuration parameter information entered, the system indicates problems using a highlighting mechanism. All the configurations appear in purple and italic, and the device with configuration problems appears in red and bold.

The system detects the following errors:

- Bad **Role Name**
  The valid **Role Name** characters are:
  - a through z (lower case)
  - A through Z (upper case)
  - "_" (underscore)
- Bad **MAC Address**
  The valid **MAC Address** characters are:
  - 0 through 9
  - A through F
- Wrong **IP Address**
- Wrong **Subnet Mask**
- Wrong **Gateway**
- Double Entry
  Each entry must have a unique **Role Name** or **MAC Address**. If a duplicate **Role Name** or **MAC Address** is entered, the system highlights the error.

Erroneous errors are not loaded into the DHCP server. Therefore, errors must be corrected before loading. There are two ways of correcting the error.

Correcting through Web page Make the changes on the Web page and submit the change.

Correcting through the Address Server configuration file Make the changes in the file and reboot the server.

Hot Standby

9

At a Glance

Introduction

The NOE 771 x0 modules offer a Hot Standby configuration available for Quantum controllers.

What's in this Chapter?

This chapter contains the following topics:

Overview of Hot Standby Solution for NOEs

Please Note

The Quantum Hot Standby system supports up to four NOE 771 Ethernet connections. For a more detailed description of the physical set up of a Hot Standby system, refer to the *Quantum NOE 771 xx Ethernet Modules User Guide*, 840USE11600, Chapter 9, "Hot Standby".

Description of the Hot Standby Solution

The Hot Standby solution provides bumpless transfer of I/O using remote I/O. The NOE Hot Standby support now allows automation IP Address change. Both controllers are configured identically. One controller is the Primary NOE; the other controller, the Secondary NOE. In case of a failure, the controllers switchover and the system recovers quickly.

The NOEs coordinate the swapping of IP addresses. After closing both the client and the server connections, each NOE sends a swap UDP message to its peer NOE. The sending NOE then waits a specified timeout (500 ms) for the peer swap of UDP messages. Either after receiving the messages or after a timeout, the NOE changes its IP address.

Note: NOEs must communicate with each other in order to swap IP Addresses. Schneider Electric recommends that you connect the primary and Secondary NOEs to the same switch because

- Communication failures between the NOEs increases the time to swap
- Connecting two NOEs to the same switch, minimizes the probability of a communication failure

Note: Schneider Electric recommends that a switch is used to connect the NOEs to each other or to the network. Schneider Electric offers switches; please contact a local sales office for more information.

The NOE waits for either a change in the controller's Hot Standby state or the swap of UDP messages. Then the NOE performs one of two Hot Standby actions.

If the NOE:

1. Detects that the new Hot Standby state is either primary or standby:
   The NOE changes the IP address
2. Receives a swap UDP message:
   The NOE transmits a Swap UDP message and swaps the IP address

All client/server services (I/O Scanner, Global Data, Messaging, FTP, SNMP, and HTTP) continue to run after the switchover from the old to the new Primary NOE.

Note: Failure of an NOE module is not a condition for the primary system to leave the primary state.

Hot Standby and NOE Module Functionality

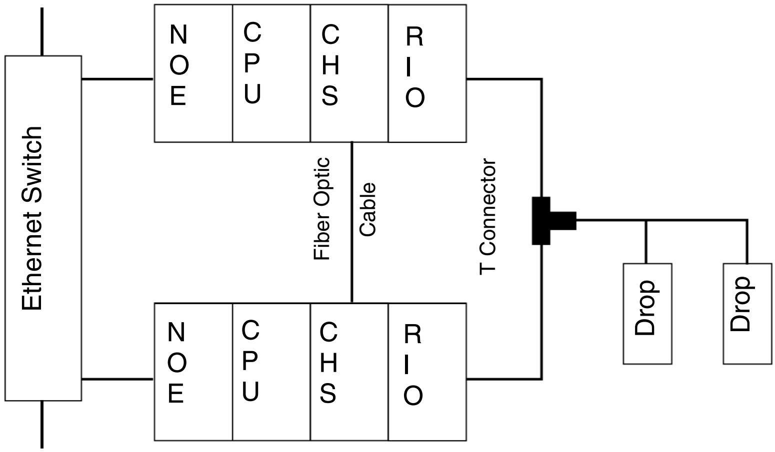
The NOE 771 family provides different Ethernet services. Some services are enabled or disabled in a Hot Standby system. The following table shows which services are enabled and disabled.

Service	NOE 771 x0	NOE 771 x1
I/O Scanning	Disabled	Enabled
Global Data	N/A	Enabled
Modbus Messaging	Enabled	Enabled
FTP/TFTP	FTP Enabled	Enabled
SNMP	Enabled	Enabled
HTTP Server	Enabled	Enabled
DHCP	N/A	Disabled

Hot Standby Topology

Hot Standby Interconnection

The following diagram shows a Hot Standby system the relationship between the two redundant systems.Two CHS 110 modules are connected via a fiber optic link. The RIOs are connected both to each other and to the RIO Drops.



Note: The following three items are important.
1. The two systems must be identical.
2. The order of the modules in each rack must be the same.
3. The software revisions must be the same.

In the preceding diagram the NOEs are connected to the same switch. Connecting to the same switch is recommended but not required. Connecting to the same switch is recommended because the NOEs communicate with each other in order to swap the IP address.
There are two reasons for connecting to the same switch:
- If a failure to communicate between the NOEs occurs, the time to swap increases.
- Therefore to minimize the probability of a failure, connect the two NOEs to the same switch.

The other requirement for the switches is that they are on the same sub network.

NOE Configuration and Hot Standby

TCP/IP Configuration

When an NOE goes into service the first time, the NOE attempts to get its IP Address from a BOOTP server. If no BOOTP server is available, the NOE derives its IP Address from its MAC address. Connecting to a BOOTP server or deriving the IP Address from a MAC address allows you a connection to the NOE, that enables you to download a project to the PLC.

All standard rules apply to IP addressing with the additional restriction that the IP address cannot be greater than 253 or broadcast address minus 2. Also, no other device can be assigned the configured IP + 1 address.

IP Address Assignment

Configuring the NOE

The NOE can be configured to work in conjunction with the Hot Standby controller. Since the Primary and Secondary controllers must have an identical configuration, the configured IP Addresses will be the same. The NOE's IP Address is either the configured IP Address or the configured IP Address +1. The IP Address is determined by the current local Hot Standby state.
In the Offline state, the IP Address is determined by whether or not the other controller is in transition to the Primary state.

Note: For a Hot Standby system, the two IP Addresses will be consecutive.

The following table shows the IP Address assignments.

Hot Standby State	IP Address
Primary	Configured IP Address
Standby	Configured IP Address + 1
Transition from Primary to Offline	Configured IP Address, if peer controller **does not** go to Primary
Transition from Standby to Offline	Configured IP Address + 1

Note: Offline - Results depend on whether or not the other controller is detected as in transition into the primary state. If Current IP is the configured IP Address, then change the IP Address to the configured IP Address + 1.

IP Address Transparency

For continued Ethernet communication, the new Primary NOE must have the same IP Address as the former Primary NOE. The IP Address in the Secondary NOE (an NOE in the secondary state) is IP Address + 1.
The NOEs integrated in the Hot Standby configuration coordinate this swapping IP Address with the management of Ethernet services used.

Note: Do not use the address IP + 1. For a Hot Standby system, do not use consecutive addresses of the configured IP Address. If you configure the last IP Address (255), NOE returns diagnostic code "Bad IP configuration".

NOE Operating Modes and Hot Standby

The NOE Modes

The NOE modes are

- **Primary Mode**
  The Hot Standby state is primary, and all services are active.
- **Secondary Mode**
  The Hot Standby state is standby, and all server services are except DHCP.
- **Standalone Mode**
  Occurs when NOE is in a non redundant system, or if the CHS module is not present or is not healthy.
- **Offline Mode**
  CPU is stopped.
  CHS module is in Offline mode.

The Hot Standby and the NOE operating mode are synchronized by the conditions described in the following table.

CHS Module Status	HSBY State	NOE Operating Mode
Present and Healthy	Primary	Primary
Present and Healthy	Standby	Secondary
Present and Healthy	Offline	Offline
Present and Healthy	Unassigned	Standalone
Not present or unhealthy	N/A	Standalone

Any one of four events will affect the NOE operating mode. These four events occur when the NOE is powered-up, when an NOE executes a Hot Standby switchover, when an NOE goes to offline mode, or when a new application is downloaded to the NOE.

Power-Up and IP Address Assignment

The process of powering up affects the NOE's IP Address assignment. To clarify what happens during a power-up, the following two sections describe the power-up effects on IP Address assignment and Ethernet services.
An NOE obtains its IP Address assignment at power-up as follows:

If the HSBY state is ...	Then the IP Address assigned is ...
Unassigned	Configured IP Address
Primary	Configured IP Address
Secondary	Configured IP Address + 1
Unassigned to Offline	See the *Offline Mode at Power-up Sequence* table following

If two NOEs power-up simultaneously, a "resolution algorithm" determines the Primary NOE, and after determining the Primary NOE, the "resolution algorithm" assigns the configured IP Address to the Primary NOE and then assigns the configured IP Address + 1 to the Secondary NOE.
Offline Mode at Power-up Sequence table:

Offline Mode at Power-up Sequence	Result
Controller A powers-up before controller B	• IP Address of controller A is configured IP Address • IP Address of controller B is the configured IP Address + 1
Both controller A and controller B power-up a the same time	The resolution algorithm will assign controller A the configured IP address and will assign controller B the configured IP address + 1.

The NOE performs a "duplicate IP" test by issuing an ARP request to the configured IP Address. If a response is received within 3 seconds, the IP Address remains at the Default IP and blinks a diagnostic code.
If no IP configuration exists, the NOE remains in standalone mode, and the IP Address must be obtained from either a BOOTP server or from a MAC address.

Power-Up and Ethernet Services

The process of powering up affects the status of client/server services. To clarify what happens during a power-up, the following section describes the power-up effects on the Ethernet services.
The following table shows how the status of an NOE service is affected by the Hot Standby state.

HSBY State	Status of NOE Services					
	Client Services		Client/Server Services	Server Services		
	I/O Scanner	Global Data	Modbus Messaging	FTP	SNMP	HTTP
Unassigned	Run	Run	Run	Run	Run	Run
Primary	Run	Run	Run	Run	Run	Run
Secondary	Stop	Stop	Run	Run	Run	Run
Offline	Stop	Stop	Run	Run	Run	Run

Hot Standby Switchover

The following steps describe how NOEs coordinate the Hot Standby switchover.

Step	Action
1	NOE A (installed in a HSBY rack) detects that is local controller changed from Primary to Offline.
2	NOE A changes its HSBY state from Primary to Offline with the same Ethernet services running, starts its watch-dog timer (with 500 ms timeout setting), and expects from its peer NOE an UDP request to swap the IP Address.
3	NOE B (installed in peer HSBY rack) detects that its local controller changed state from Secondary to Primary.
4	NOE B stops all Ethernet services, sends an UDP request to its peer NOE (NOE A) for the synchronization of the IP Address swap, starts its watch-dog timer (with 500 ms timeout setting), and then waits for an UDP response from its peer NOE.
5	Once NOE A receives the UDP request from NOE B (or after its watch-dog timer times out), it stops all Ethernet services, sends an UDP response to NOE B (no UDP response is sent to NOE B for watch-dog timeout case), swaps IP Address as Secondary, and starts Secondary services.
6	As soon as NOE B receives the UDP response from NOE A (or after its watch-dog timer times out), it swaps IP Addresses and starts Ethernet services as Primary.
7	After NOE A senses that its local controller changes state from Offline to Standby, it changes to Secondary accordingly.
8	The Secondary NOE now becomes the Primary NOE.
9	Primary NOE opens all client connections and listens for all server connections and re-establishes those connections.
10	Simultaneously, Secondary NOE listens for all server connections and re-establishes those connections.

Additional Switchover Information

The following list provides additional information about the NOE's IP addressing process resulting from a Hot Standby switchover.

- Some MSTR/IEC Function blocks will not complete their transaction as a result of the IP Address swap.
  In this case, the MSTR/IEC Function block will return the error code 0x8000.
- While the NOE is in the process of performing the above actions, a new MSTR/IEC Function block may become active.
  No resources are available to service the new MSTR/IEC Function block. Therefore, the NOE will **not** service this new MSTR/IEC Function block, and all three output lines will be low.

Going to Offline

When either the CPU stops or the Hot Standby state goes to offline mode, two events occur:

1. NOE mode goes to Offline
2. NOE uses the IP Address of the present configuration

IP Address Assignment and Going Offline

HSBY State	IP Address Assigned Is ...
Primary to Offline	Configured IP Address, if other controller **does not** go to Primary
Standby to Offline	Configured IP Address + 1

Address Swap Times

Description

The following table details what the "time for an Address swap" comprises, such as the time to close connections, time to swap IP addresses, or time to establish connections.
The following table shows the swap time for each of the Ethernet services.

Service	Typical Swap Time	Maximum Swap Time
Swap IP Addresses	6 ms	500 ms
I/O Scanning	1 initial cycle of I/O Scanning	500 ms + 1 initial cycle of I/O scanning
Global Data	For times, please see the 840USE11600, *Quantum NOE 771 xx Ethernet Modules User Guide*	500 ms + 1 CPU scan
Client Messaging	1 CPU scan	500 ms + 1 CPU scan
Server Messaging	1 CPU scan + the time of the client reestablishment connection	500 ms + the time of the client reestablishment connection
FTP/TFTP Server	The time of the client reestablishment connection	500 ms + the time of the client reestablishment connection
SNMP	1 CPU scan	500 ms + 1 CPU scan
HTTP Server	The time of the client reestablishment connection	500 ms + the time of the client reestablishment connection

Network Effects of Hot Standby Solution

Overview

The Hot Standby solution is a powerful feature of NOEs, a feature that increases the reliability of your installation. Hot Standby uses a network, and using the Hot Standby feature over a network can affect the behavior of

- Browsers
- Remote and Local clients
- I/O Scanning service
- Global Data service
- FTP/TFTP server

The following are factors you may encounter while using the Hot Standby solution.

Browsers

Note: In Hot Standby configuration the NOE's I/O scanner is enabled.

If a browser requests a page and during the process of downloading that page an IP Address swap occurs, the browser will either hang or time out. Click the **Refresh** or **Reload** button.

Remote Clients

Hot Standby swaps affect remote clients.
An NOE will reset under the following conditions:

- **Remote Connection Request during Hot Standby Swap**
  If a remote client establishes a TCP/IP connection during a Hot Standby swap, the server closes the connection using a TCP/IP reset.
- **Hot Standby Swap during Remote Connection Request**
  If a remote client makes a connection request and a Hot Standby swap occurs during the connection request, the sever rejects the TCP/IP connection by sending a reset.
- **Outstanding Requests**
  If there is an outstanding request, the NOE will not respond to the request, but the NOE will reset the connection.

The NOE will do a Modbus logout if any connection has logged in.

Local Clients

During a swap, the NOE will reset all client connections using a TCP/IP reset.

I/O Scanning Service

The I/O Scanning provides the repetitive exchange of data with remote TCP/IP nodes I/O devices. While the PLC is running the Primary NOE sends Modbus Read/ Write, read or write request to remote I/O devices, and transfer data to and from the PLC memory. In the secondary controller, the I/O scanning service is stopped.
When the Hot Standby swap occurs, the Primary NOE closes all connections with I/ O devices by sending a TCP/IP reset. The I/O scanning service in this NOE is standby.
After the swap, the new Primary NOE re-establishes the connection with each I/O devices. It restarts the repetitive exchange of data with these re-connections.

Global Data (Publish/ Subscribe) Service

The Hot Standby NOE is one station within a distribution group. Distribution groups exchange application variables. Exchanging application variables allows the system to coordinate all the stations in the distribution group. Every station publishes local application variable in a distribution group for all other stations and can subscribe to remote application variables independent of the location of the producer.
The communication port has only one multicast address.
In this network service, the Hot Standby controllers are viewed like only one station. The Primary NOE publishes the Hot Standby application variables and receives the subscription variables. The Secondary NOE global data service is in a stopped state.
When the Hot Standby swap occurs, the Primary NOE stops the Global Data service. The NOE does not publish the local variable during a swap. And after the swap, the new Primary NOE starts to publish application variables and to receive the subscription variables.

FTP/TFTP Server

The File Transfer Protocol/Trivial File Transfer Protocol (FTP/TFTP) server is available as soon as the module receives an IP address. Any FTP/TFTP client can logon to the module. Access requires the correct user name and password. Hot Standby allows only one active FTP/TFTP client session per NOE module.
When the Hot Standby swap occurs, the Primary and Secondary NOEs close the FTP/TFTP connection. If a user sends an FTP/TFTP request during the swap, the communication is closed.
Whenever you re-open communication, you must re-enter a user name and a password.

Using the Network Options Ethernet Tester

10

At a Glance

Introduction

This chapter describes how to use the *Network Options Ethernet Tester* with a Windows based PC. This program can monitor the network by supplying you with operational statistics and provides the capability of reading and writing PLC registers.

What's in this Chapter?

This chapter contains the following topics:

Installing the Network Options Ethernet Tester

Overview

The following information describes how to install the Network Options Ethernet Tester.

Introduction

An Ethernet module may act as a client or as a server.
If it will be acting as a client -- that is, initiating transactions on the network for its Quantum controller -- then you must program an MSTR block in ladder logic. For details about the MSTR block, please refer to *Transferring Data using Communication Blocks, p. 69*.
The Ethernet module may also act as a server, responding to requests and commands from devices on the network for its Quantum controller.
The Network Options Ethernet Tester utility allows you to get and clear statistics and to read and write registers over the network, using a Windows-based PC.
You may also create your own program using the Ethernet module as a server. For guidance in creating your own program, refer to *Ethernet Developers Guide, p. 269*.

Note: In its capacity as server, the Ethernet module can accept only 32 connections at any one time.
- 32 connections for the 140 NOE 771 -00 and -10
- 64 (client and server) connections for the 140 NOE 771 -01 and -11

If a new connection is attempted and the server has already reached its limit, it will terminate the last used connection to make room for the new one.

Installation Procedure

The Network Options Ethernet Tester is supplied to you on a utility diskette. The following steps describe how to install the tester on your PC.

Step	Action
1	Insert the Network Options Ethernet Tester utility disk into drive A:
2	Select **Run** from the Program Manager file menu.
3	Type A:\SETUP and click on the **OK** button–the Welcome dialog will appear.
4	Click on the **Next** button and follow the instructions that appear in each of the dialogs to complete the installation.* *Each installation dialog has **Back** and **Next** buttons that allow you to move back to the previous dialog or to move forward to the next dialog.

Establishing a Connection with an Ethernet Module

Overview

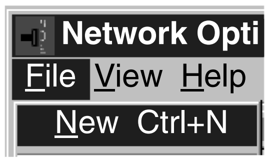
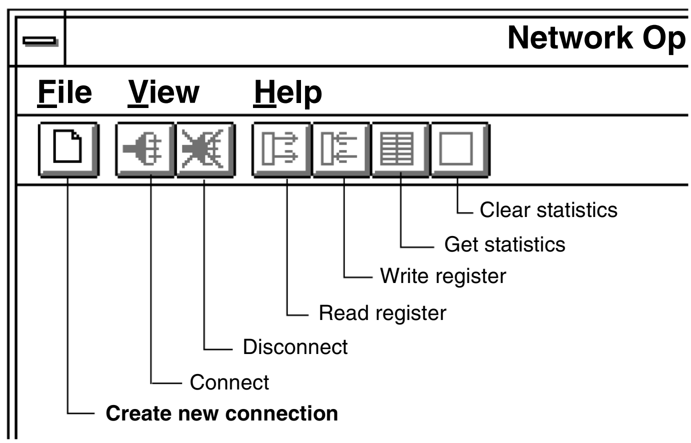
The following information describes how to use the Network Options Ethernet Tester when connecting to an Ethernet adapter.

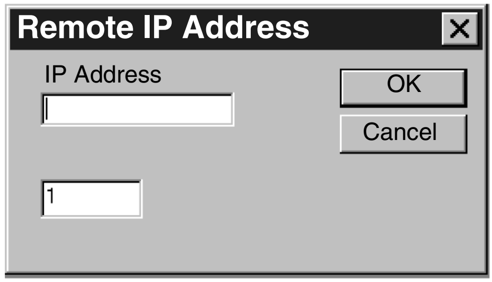
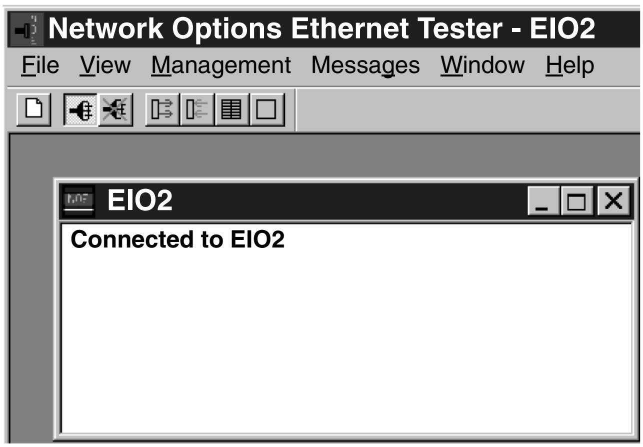
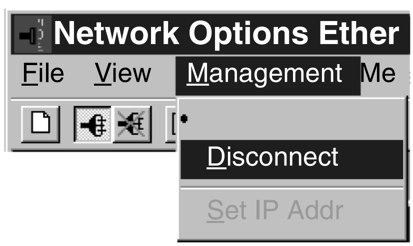
What You Must Know

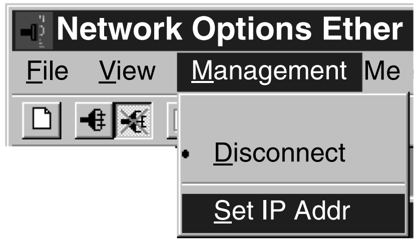
To establish a connection with an Ethernet module using the Network Options Ethernet Tester you must know the module's IP network address or role name.

Procedure for Connecting with an Ethernet Module

Perform the following steps to establish a connection with an Ethernet module using the Network Options Ethernet Tester.

Step	Action
1	As shown in the following figure, from the initial menu select **File | New**,  or click the **Create new connection** icon.  The result of **File | New** or click **Create new connection** is that the Remote IP Address dialog box displays.

Step	Action
2	Type the adapter's IP network address or role name in the IP Address field.  Click **OK**. As shown in the following figure, this dedicates a connection from your PC to the designated Ethernet module and brings you to the main menu. 
3	You may establish several connections with the same module or with other modules by repeating step 2 for each new connection.
4	When you are ready to disconnect, select **Management | Disconnect** from the pulldown menu, or click the **Disconnect** button in the toolbar. 

Step	Action
5	After disconnecting from one module, you may reassign its dedicated connection by selecting **Management | Set IP Addr** from the pull-down menu. Type the new IP network address or role name in the box provided. 

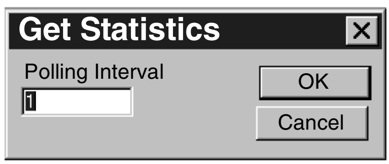
Getting and Clearing Statistics

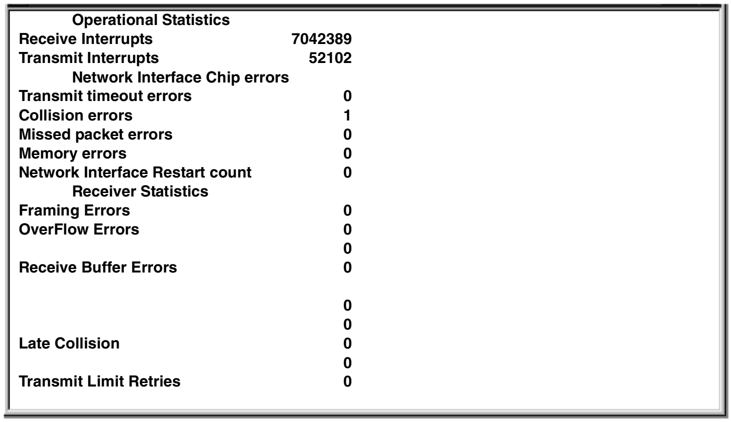
Overview

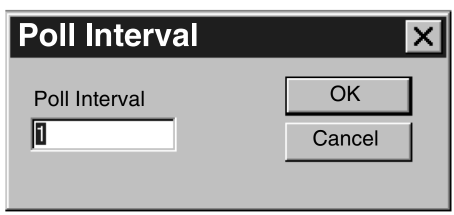
The following information describes how to use the Network Options Ethernet Tester to obtain and to clear statistics.

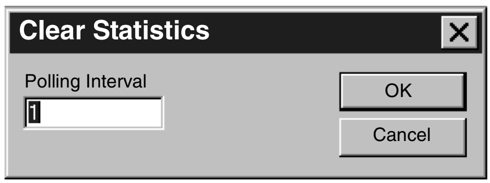
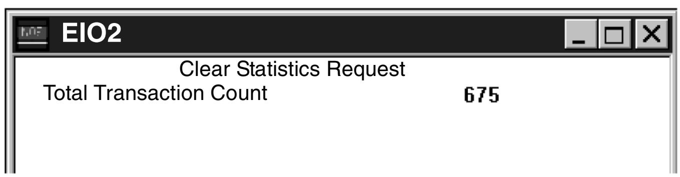
Procedures for Getting and Clearing Statistics

Perform the following steps to get statistics from the Ethernet module using your Network Options Ethernet Tester.

Step	Action
1	Establish a connection with the Ethernet module.
2	As shown in the following figure, select **Messages** from the main menu and choose **Get Stats** from the pull-down menu. Alternatively, you can click on the **Get Statistics** button in the toolbar. The Get Statistics dialog box, which is shown in step 3, displays. 
3	As shown in the following figure, you can type a polling interval (the number of seconds between transactions) in the box provided and click on the **OK** button. 

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>As shown in the following figure, complete statistics for the module will be printed in the window for this connection.

</td></tr>
<tr><td>5</td><td>As shown in the following figure, to change the polling interval without interrupting communication with the Ethernet module, select Messages and choose Poll Interval.

</td></tr>
<tr><td>6</td><td>As shown in the following figure, type the new polling interval in the box provided, and click on the OK button.

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

Step	Action
7	As shown in the following figure, to clear statistics select **Messages** and choose **Clear Stats** from the pull-down menu. Alternatively, click on the **Clear Statistics** button in the toolbar. 
8	The following figure shows the Clear Statistics dialog box. Click on the **OK** button.  **Result**: As shown in the following figure, the Clear Statistics Request for the connection displays. 

Statistics

Overview

The following information describes the type of statistics provided by the Network Options Ethernet Tester.

Statistics Description

The Network Options Ethernet Tester provides status information in the format shown in the following table. Other statistics provided by the Network Options Ethernet Tester appear in a list following the table.

Parameter	Information	
Model:	Model number	
Media:	10 BASE-T	HALF DUPLEX
	100 BASE-T	DUPLEX
	100 BASE-FX	
Controller:	Running	
	Stopped	
Crash Log Empty	Yes	
	No - there is a crash log entry present	

The Network Options Ethernet Tester can provide the following statistics:
- Total Transaction Count. How many transactions have been completed.
- IP Address.
- Receive Interrupts and Transmit Interrupts. The number of times the PCNET controller chip has generated interrupts.
- Transmit timeout errors. The number of times the transmitter has been on the channel longer than the interval required to send the maximum length frame of 1519 bytes. This is also known as a babble error.
- Collision errors. The number of collisions detected by the Ethernet chip.
- Missed packet errors. The number of times a received frame was dropped because a receive descriptor was not available.
- Memory errors. The number of times an Ethernet controller chip experienced an error accessing shared RAM. A memory error will cause a restart.
- Restart count. The number of times the Ethernet controller chip was restarted due to fatal runtime errors, including memory errors, transmit buffer errors, and transmit underflow.
- Framing error. The number of times an incoming frame contained a non-integer multiple of eight bits.
- Overflow errors. The number of times the receiver has lost part or all of an incoming frame, due to an inability to store the frame in memory before the internal FIFO overflowed.
- CRC errors. The number of times a CRC (FCS) error was detected on an incoming frame.

- Receive buffer errors. The number of times a receive buffer was not available while data chaining a received frame.
- Transmit buffer errors. The number of times the end packet flag on the current buffer was not set and the Ethernet controller did not own the next buffer. A transmit buffer error causes a restart.
- Silo Underflow. The number of times a packet was truncated due to data late from memory. A Silo Underflow will cause a restart.
- Late Collision. The number of times a collision was detected after the slot time of the channel had elapsed.
- Lost Carrier. The number of times a carrier was lost during a transmission.
- Transmit retries. The number of times the transmitter has failed after 16 attempts to transmit a message, due to repeated collisions.

These statistics also may be obtained from the MSTR block. Refer to the *Ladder Logic Block Library User Guide, 890 USE 100 00* for details.

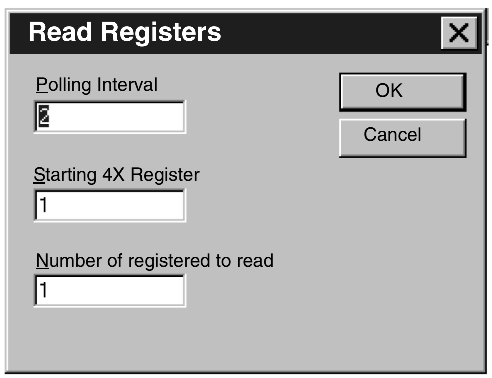
Reading Registers

Overview

The following information describes how to read Ethernet adapter registers with the Network Options Ethernet Tester.

Reading Registers Procedure

The following procedure describes how to read registers from the Ethernet adapter using your Network Options Ethernet Tester.

Step	Action
1	Establish a connection with the Ethernet adapter.
2	Select **Messages** from the main menu.
3	As shown in the following figure, choose **Read Register** from the **Messages** menu, or click the **Read Register** button in the toolbar. The Read Register dialog box displays. 
4	Type the number of seconds between transactions in the **Polling interval** field. 
5	In the **Starting 4x Register** field, type the register number of the first 4x register. When typing the 4x register number, omit the leading 40 or 400, as shown in the figure in Step 4.
6	Type in the number of registers to read in the **Number of registers to read** field.
7	Click **OK**. The register values display in the window for this connection. Five values will be listed in each row, with the number of the first register at the beginning of the row.

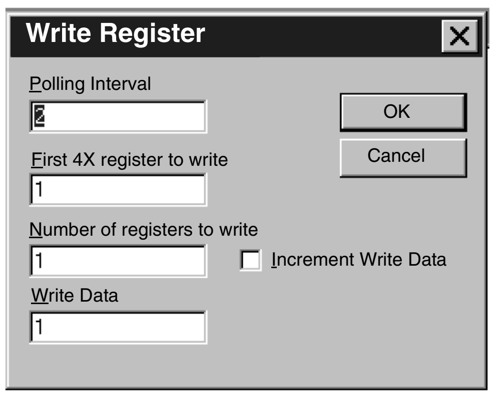
Writing Registers

Overview

The following information describes how to write registers from the Ethernet module to the Network Options Ethernet Tester.

Writing Registers Procedure

The following procedure describes how to write registers from the Ethernet module using your Network Options Ethernet Tester.

Step	Action
1	Establish a connection with the Ethernet module.
2	Select **Messages** from the main menu.
3	As shown in the following figure, select **Write Register** from the **Messages** menu, or click the **Write Register** button in the toolbar. The Write Register dialog box displays. 
4	Type the number of seconds between transactions in the **Polling Interval** field. 
5	Type in the number of the first 4x register you want to write in the **First 4x Register to Write** box. When typing the 4x register number, omit the leading 40 or 400, as shown in the figure in step 4.
6	Type in the number of registers to write in the **Number of registers to write** box.
7	Type in the data to be written to those registers in the **Write Data** box.
8	Select the **Increment Write Data** check box if you want the data you have entered to be increased by one with each transaction.
9	Click **OK**. The register values will be displayed in the window for this connection.

Read or Write Request Error

If you try to read or write registers and an error occurs, the Network Options Ethernet Tester will display a Read Request Error or Write Request Error. The error codes correspond to MSTR block error codes. For more information, refer to the *Ladder Logic Block Library User Guide*, 840USE10100.

Using the Test Button

Overview

This section describes the Test Button option available in the Network Options Ethernet Tester.

General Description - Test Button

The Network Options Ethernet Tester is an option that allows you to test data. There are three methods of testing:

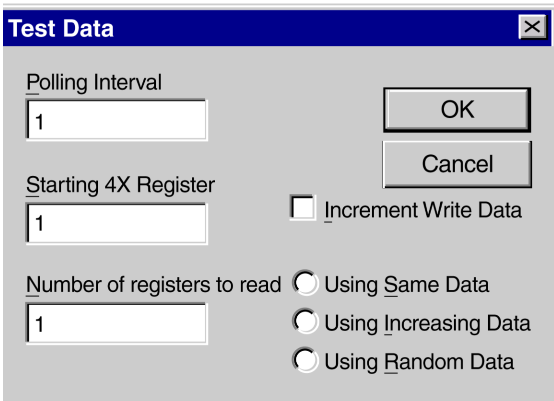
- Use the same data written to all registers
- Use increasing data written uniquely to each register
- Use random data written uniquely to each register

The test writes the data then reads the data. A pass/fail counter is used to display the number of times the data written is read correctly.

Test Function and Options

You may access the test option through the **Test Data** menu option. On the **Messages** menu click **Test Data...**.
The second option is to use the test button, which appears on the menu bar. Click the test button.

Both options open the **Test Data** dialog box.



The Test Data dialog box requests that you enter three variables:
- Polling Interval
- Starting 4X Register
- Number of registers to read

The **Test Data** dialog box offers you the option to increment Write Data with every transaction. Check the **Increment Write Data** box.
Once selected, you need to choose one of three options for incrementing Write Data. Choose by selecting the appropriate radio button.
- **Using Same Data**
  Each register receives the same data.
  For example: Register 1 receives the value 1. Register 2 receives the value 1.
- **Using Increasing Data**
  Each register receives unique data
  For example: Register 1 receives the value 1. Register 2 receives the value 2.
- **Using Random Data**
  Each register receives a data value assigned randomly.
  For example: Register 1 receives the value 625. Register 2 receives the value 264.

SNMP

11

At a Glance

Introduction

The following material describes the Simple Network Management Protocol (SNMP) and the Schneider private MIB. Under the Schneider private MIB is the Transparent Factory Ethernet private MIB. All are configured on your NOE.

What's in this Chapter?

This chapter contains the following topics:

SNMP

Overview

This following information describes the Simple Network Management Protocol (SNMP), which is configured on your NOE. Additional information is available in the section named *Configure SNMP Page*.

Introduction

Network management software allows a network manager to

- Monitor and control network components
- Isolate problems and find their causes
- Interrogate devices such as a host computer, routers, switches, and bridges to determine their status
- Obtain statistics about the networks to which they attach

Manager/Agent Paradigm

Network management software follows the conventional client-server model.
To avoid confusion with other network communication protocols that use the client/server terminology, network management software uses the following terms:

- *Manager*
  For the client application that runs on the manager's computer
- *Agent*
  For the application that runs on a network device

The manager uses conventional transport protocols (e.g., TCP or UDP) to establish communication with the agent. Managers and agents then exchange requests and responses according to the network management protocol.

Simple Network Management Protocol

Your NOE module is configured with the Simple Network Management Protocol (SNMP), which is the standard protocol used to manage a local area network (LAN). SNMP defines exactly how a *manager* communicates with an *agent*.
The SNMP defines the format of the requests that a manager sends to an agent and the format of the replies that the agent returns to the manager.

The MIB

Each object to which SNMP has access must be defined and given a unique name. Both the manager and agent programs must agree on the names and the meanings of the fetch and store operations. The set of all objects SNMP can access is known as a *Management Information Base (MIB*).

The Private MIB

Schneider obtained a private MIB, Groupe_Schneider (3833). Under the Groupe Schneider private MIB is a Transparent Factory Ethernet (TFE) private MIB. The Transparent Factory SNMP embedded component controls the Schneider private MIB function.

Choosing a SNMP Manager

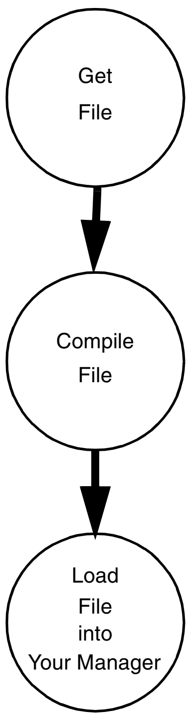
If you have an SNMP Manager operating now, you may continue to use that SNMP Manager. If you are selecting an SNMP Manager, there are many SNMP Managers on the market, and you may use any of these managers. You must use an SNMP Version 1 compliant manager.

Using a SNMP Manager

The process for obtaining a SNMP Manager

Step	Action
1	Get Schneider `.mib` file from the NOE Web page.
2	Compile `.mib` file in the compiler that comes with the NOE.
3	Load compiled `.mib` file to the manager.
4	When you are done, you will see the Schneider private MIB Manager in your manager.

The process is simple.



More SNMP Information

SNMP and related subjects are well documented on Web sites and in many books

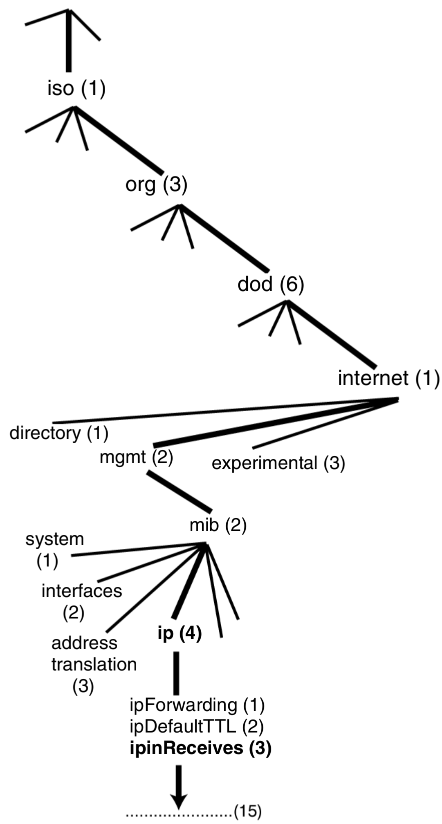
- As of this writing, a useful description appears on Microsoft's *Technet* pages. Browse to *http://www.microsoft.com/technet*. Use the **Search** function to find "Network Management for Microsoft Networks Using SNMP."
- Use an Internet search engine to search for a SNMP introduction, a SNMP tutorial, and other topics on SNMP.
- The SNMP FAQ from the news group `comp.protocols.snmp` appear on many `.com` and `.org` Web pages. Search for the combination of "`comp.protocols.snmp`" and "`FAQ`."
- A list of print books about SNMP appears in the SNMP FAQ. In addition, a search of most online retail book sites will yield a substantial list of titles.

ASN.1 Naming Scheme

An Example

Objects in a MIB are defined with the ASN.1 naming scheme that assigns each object a long prefix that guarantees that the name will be unique. For example, an integer that counts the number of IP datagrams that a device has received is named: *iso.org.dod.internet.mgmt.mib.ipInReceives*.

The following figure depicts the ASN.1 Naming Scheme example.



This object name is represented in an SNMP message by assigning each part an integer. So, the above message would appear as 1.3.6.1.2.2.4.3.
Each integer has the following meaning.
- 1 = ISO (International Organization for Standardization)
- 3 = identified organization — one of branches under the ISO root
- 6 = U. S. Department of Defense (DOD) — one of the children under branch1.3
- 1 = the Internet subtree under 1.3.6
- 2 = the mgm branch — (one of seven) of the Internet subtree. It is managed by the Internet Assigned Numbers Authority, and includes the standard MIBs
- 2 = mib-2(1) group of managed objects
- 4 = ip — the mib-2(1) IP group (one of 11)
- 3 = ipinReceivers — the MIB object

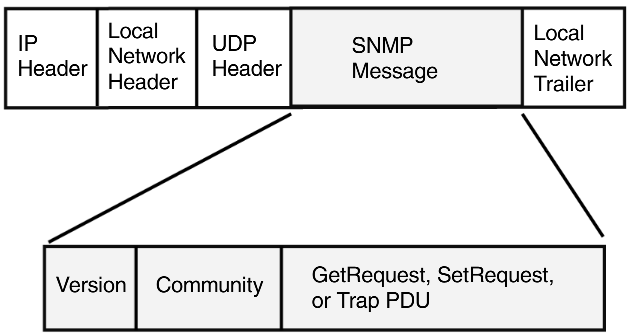
Configuring a NOE with SNMP

The Object Identifier (OID)

In the ASN.1 Naming Scheme example, the MIB object identified by the notation 1.3.6.1.2.2.4.3 is referred to as the Object Identifier or OID. All OIDs can be envisioned as part of a tree structure which begins at the root (ISO) and branches out with each subtree identified by an integer.

SNMP Protocol Data Units

SNMP uses Protocol Data Units (PDUs) to carry the requests and responses, between the manager and the agents, for the information contained in an OID.
As the following figure shows, the SNMP message is the innermost part of a typical network transmission frame.



The PDUs within the SNMP initiate the communication between the manager and the agents.
The SNMP installed on your NOE module uses the following three PDUs.
- GetRequest
- SetRequest
- Trap

GetRequest PDU

The GetRequest (shortened to Get) PDU is used by the SNMP manager to retrieve the value of one or more objects (OIDs) from an agent.

SetRequest PDU

The SetRequest (shortened to Set) PDU is used by the SNMP manager to assign a value to one or more objects (OIDs) residing in an agent.

Trap PDU

The Trap PDU is used by the agent to alert the manager that a predefined event has occurred.

Version & Community Identifiers

The version identifies the version number of the SNMP software being used by the manager and the agent. Your NOE supports Version 1 of the SNMP. The community is an identifier that you assign to your SNMP network. If community names for the manager and the agent do not agree, the agent will send an authentication failure trap message to the manager. If the community names and version number agree, the SNMP PDU will be processed.

What Can Be Configured

Your NOE module can be configured to send an authentication trap to two SNMP managers if it receives a community name in a Get/Set request that does not match the configured name. Also, you can configure the SysContact and SysLocation via the configuration page in the module's Embedded Web pages. After making changes in the SNMP Configuration Web page and to set those changes, reboot the module using hot swap.

Configuring an NOE with TFE Private MIB

Introduction

A MIB, a Management Information Base, is an element used in network management. Network management services are based on the need to monitor and to manage:

- Performance
- Fault Occurrences
- Security

Each MIB contains a finite number of objects. Manage your MIB with a management station running an SNMP management application. The management application uses **GETs** and **SETs** to retrieve system information and to set system environment variables.

> **Note:** The TFE private MIB is available only in the 140 NOE 771 -01 and -11, the Transparent Factory / Real Time modules.
> The 140 NOE 771 -00 and -10, the Transparent Factory modules, use the previous MIB.

Schneider Private MIB

Schneider Automation obtained a Private Enterprise Number (PEN) from the Internet Assigned Numbers Authority (IANA). That number represents a subtree in the SNMP MIB, a number that is a unique identifier used for Groupe Schneider. The object identifier for the root of the Groupe Schneider subtree is `1.3.6.1.4.1.3833` and represents a path to the subtree as follows:

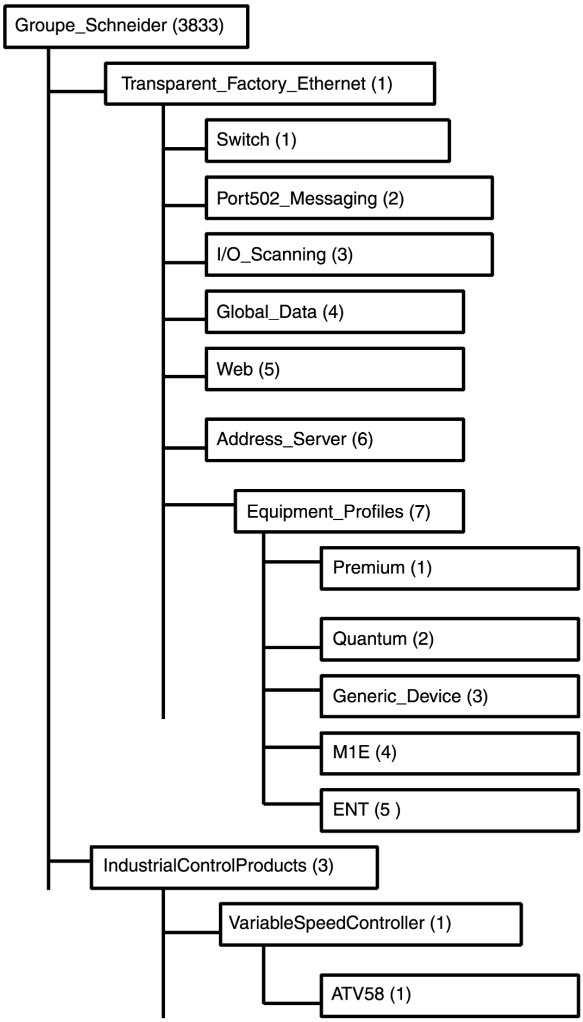
```
iso(1)
    org(3)
        dod(6)
            internet(1)
                private(4)
                    enterprise(1)
                        GroupeSchneider(3833)
                            Transparent_Factory_Ethernet(1)
```

Under the GroupeSchneider private MIB is a Transparent Factory Ethernet (TFE) private MIB, **Transparent_Factory_Ethernet(1)**.

TFE Private MIB

The Transparent Factory SNMP-embedded component controls the Schneider private MIB function. The Schneider private MIB, and associated services, perform Network Management on all components of the system. The Transparent Factory private MIB provides the data to manage the main Transparent Factory communication services for all the communication components of the Transparent Factory architecture (ETYs, NOEs, third party toolkit, ENTs, M1Es). The Transparent Factory private MIB does not define the specific management applications and policies.

The diagram below illustrates the Schneider Electric (Groupe_Schneider (3833) private enterprose MIB subtree.



The **Groupe_Schneider (3833)** subtree is the root of Groupe Schneider's private MIB in the Structure of Management Information (SMI) used by SNMP and defined in RFC-1155, which is a specification that defines the structure and identification of management information for TCP/IP-based Internets.

Transparent Factory Ethernet Subtree

The **Transparent_Factory_Ethernet (1)** subtree defines groups that support the Transparent Factory Ethernet services and devices.

Service	Description
Switch (1)	Subtree defines a brand of switches labeled: ConneXium switches private MIB
Port502_Messaging (2)	Subtree defines objects for managing explicit client / server communications supporting applications, such as HMI, SCADA, or programming tools
I/O_Scanning (3)	Subtree defines objects for managing I/O device communications that use the I/O Scanner mechanism with the MB/TCP protocol
Global_Data (4)	Subtree defines objects for managing the application coordination service using a publish / subscribe protocol
Web (5)	Subtree defines objects for managing the activity of the embedded Web servers
Address_Server (6)	Subtree defines objects for managing the activity of the BOOTP and (or) DHCP servers
Equipment_Profiles (7)	Subtree identifies objects for each type of device in Transparent Factory Ethernet's product portfolio

Device subtrees, or groups, will be defined for the following devices:
- **Premium(1)**
- **Quantum(2)**
- **Generic_Device(3)**
- **M1E(4)**
- **ENT(5)**

As devices are added to Schneider's catalog, Schneider's private MIB will be extended in the following manner:
- If needed, a Transparent Factory, communication-service object will be added for the new device in the corresponding **Equipment_Profiles(7)** subtree. As many objects as needed can be added to this subtree.
- If needed, a new branch will be added at the same level as **Transparent_Factory_Ethernet(1)**. This subtree will be created for product-specific objects (such as the ATV58 object under the **IndustrialControl-Products (3) subtree)**

When a new device is created, a corresponding object description is created in the ASN.1 format. The ASN.1 file(s) are then given to producers of SNMP manager software for inclusion in their products.

Port502 Messaging Subtree

The Port502_Messaging (2) subtree, or group, provides connection management and data flow services. The following list describes the function of each object.

Service	Description
port502Status(1)	Indicates the status of the service (Idle, Operational)
port502SupportedProtocol(2)	Indicates the supported protocols (MODBUS, Xway)
port502IpSecurity(3):	Indicates the status of the Port502 IP Security service (enabled/disabled)
port502MaxConn(4)	Indicates the maximum TCP connection number supported by the Port502 entity
port502LocalConn(5)	Indicates the TCP connection number currently opened by the local Port502 entity
port502RemConn(6)	Indicates the TCP connection number currently opened by the remote entity to the local Port502 entity
port502IpSecurityTable(7)	Indicates a table containing the number of unsuccessful TCP connection open tries from a remote TCP entity
port502ConnTable(8)	Indicates a table containing Port502 TCP specific information (MsgIn, MsgOut)
port502MsgIn(9)	Indicates the total number of Port502 messages received from the network
port502MsgOut(10)	Indicates the total number of Port502 messages sent from the network
port502MsgOutErr(11)	Indicates the total number of error messages built by the Port502 messaging entity and sent to the network
port502AddStackStat(12)	Indicates the support of additional port502 stack statistics 1 - Disabled 2 - Enabled
port502AddStackStatTable(13)	Indicates additional stack statistics for Port502 (optional)

I/O Scanning Subtree

The I/O_Scanning (3) subtree, or group, contains the objects related to I/O Scanning device management and associated MODBUS communications on Port502.

Service	Description
ioScanStatus(1)	Indicates the global status of the I/O Scanning service 1 - Idle 2 - Operational 3 - Stopped
ioScanMaxDevice(2)	Indicates the maximum number of devices supported by the I/O Scanning entity
ioScanPolledDevice(3)	Indicates the number of devices currently polled by the I/O Scanning entity
ioScanTransSend(4)	Indicates the total number of transactions sent by the I/O Scanning entity
ioScanGlbHealth(5)	Indicates the global status of health for the I/O Scanning service 2 - OK: Every remote I/O device is responding 4- Warning: At least one remote I/O device is not responding
ioScanDeviceTable(6)	Displays a table containing information on each remote devices polled by the I/O Scanning entity

Global Data Subtree

The Global_Data (4) subtree, or group, contains the objects related to the Global Data service.

Service	Description
glbDataStatus(1)	Indicates the global status of the Global Data service 1 - Idle 2 - Operational 3 - Stopped
glbDataMaxPub(2)	Indicates the maximum number of published variables configured by the Global Data entity
glbDataMaxSub(3)	Indicates the maximum number of subscribed variables configured by the Global Data entity
glbDataPub(4)	Indicates the total number of publications sent to the network
glbDataSub(5)	Indicates the total number of subscriptions received from the network
glbDataPubErr(6)	Indicates the total number of publication errors detected by the local entity
glbDataSubErr(7)	Indicates the total number of subscription errors detected by the local entity
glbDataGlbSubHealth(8)	Indicates the global status of health for the Global Data subscribed variables 2 - OK: The health status of all subscribed variables are OK 4 - Warning: At least one subscribed variable has a health fault
glbDataPubTable(9)	Displays a table containing information on each published variable (the number of publications, the source IP address, the number of errors)
glbDataSubTable(10)	Displays a table containing information on each subscribed variable (the number of subscriptions, the source IP address, the number of errors, Health)

Web Subtree

The Web (5) subtree, or group, contains the objects related to the Web server service.

Service	Description
webStatus(1)	Indicates the global status of the Web service 1 - Idle 2 - Operational
webPassword (2)	Indicates a switch to enable or disable the use of Web passwords 1 - Disabled 2 - Enabled
webSuccessfullAccess (3)	Indicates the total number of successful accesses to the Web site
webFailedAttempts (4)	Indicates the total number of unsuccessful accesses to the Web site

Address Server Subtree

The Address_Server (6) subtree, or group, contains the objects related to the Address Server service. The Address Server can be either a BOOTP server or a DHCP server.

Service	Description
addressServerStatus(1)	Indicates the global status of the addressServer service 1 - Idle 2 - Operational

Equipment Profile Subtree

The Equipment_Profiles (7) subtree contains a set of common objects.

Service	Description
profileProductName(1)	Displays the commercial name of the communication product in a string form (for example: 140 NOE 771 11)
profileVersion(2)	Displays the software version of the communication product in a string form (for example: Vx.y or V1.1)
profileCommunicationServices(3)	Displays a list of the communication services supported by the profile (Port502Messaging, I/O scanning Messaging, Global Data, Web, and Address Server)
profileGlobalStatus(4)	Indicates the global status of the communication module 1 - nok 2 - ok

Service	Description
profileConfigMode(5)	Indicates the IP configuration mode of the communication module 1 - Local: The IP configuration is created locally 2 - dhcpServed: The IP configuration is created by a remote DHCP server
profileRoleName(6)	Indicates the role name for the IP address management if it exists (Empty string if there is none)
profileBandwidthMgt(7)	Indicates the status of Bandwidth Management 1 - Disabled 2 - Enabled
profileBandwidthDistTable(8)	Indicates the CPU time distribution between Global Data, Port502 Messaging, I/O Scanning
profileLedDisplayTable(9)	Displays a table giving the name and the state of each module's LEDs
profileSlot(10)	Indicates the position of the communication module inside the rack if there is one. If there is no rack, the profileSlot value will be zero
profileCPUType(11)	Indicates that if the CPU type exists, this variable identifies the host for which that communication module is a part. If there is no host, the string is empty
profileTrapTableEntriesMax(12)	Indicates the maximum numbers of entries in the Trap Table. This entry equals the number of possible remote managers
profileTrapTable(13)	Displays a table allowing you to enable or disable the private traps for each of the communication services
profileSpecificId(14)	Indicates a unique Profile Specific Identification inside the equipmentProfile object of the Schneider Transparent Factory MIB. (For example the PLC Premium family is 100)
profileIpAddress(15)	Indicates the IP address of the SNMP agent
profileIpNetMask(16)	Indicates the subnet mask associated with the IP address of the SNMP agent. The value of the mask is an IP address with all the network bits set to 1 and all the host bits set to 0
profileIpGateway(17)	Indicates the default Gateway IP address of the SNMP agent
profileMacAddress(18)	Indicates the Ethernet media-dependent address of the SNMP agent

Private Traps and MIB Files

Traps are used to signal Status Changes to the manager. Using traps helps to avoid adding traffic.

The four status changes signaled by the trap are for the:

- LEDs
- Communication Ports
- I/O Scanning Health Values
- Global Data Health

The following list describes the characteristics of private traps, which means that they can:

- Send messages to the two managers whose IP addresses are configured in the SNMP configuration (either the PL7 or the Web page)
- Use the community name given to this configuration
- Enable or disable each of the Transparent Factory MIB groups: Switch (1), Port502_Messaging (2), I/O_Scanning (3), Global_Data (4), Web (5), Address_Server (6), and Equipment_Profiles (7)

Private traps are described in the MIB ASN.1 description, which is contained in a **`.mib`** text file.

Maintenance

12

At a Glance

Introduction

This chapter details information about system maintenance including accessing and clearing the Crash Log and downloading the new NOE Exec.

What's in this Chapter?

This chapter contains the following topics:

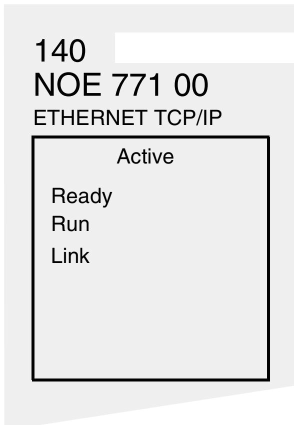
Responding to Errors

Overview

The following information describes how to respond to errors on the NOE 771 x0.

Detecting Errors

When faults occur, the NOE 771 x0 LED display can help you determine what went wrong. The following figure shows the pattern that the LEDs should display during normal operation.



The **Run** indicator will be solid. The **Coll** LED may flash, indicating that collisions are occurring on the Ethernet network. Such collisions are normal.
If a fault occurs, the normal LEDs may be extinguished or other indicators may light. This section will discuss errors reported by the **Active**, **Ready**, **Coll**, **Link**, **Kernel**, **Appl** and **Fault** indicators.
For each type of error, try the suggested remedies in the order given. If no remedy suggested here overcomes the error, call your local service representative or call Schneider Electric customer service at 1-800-468-5342 for further directions.
Certain error codes are recorded in the MSTR block. For instructions about how to read and interpret those codes through ProWORX NxT, Modsoft, or Concept, please refer to *MSTR Function Error Codes, p. 75*.

Procedure for Responding to an Active LED Error Indicator

If the Active LED fails to light, the NOE 771 00 module is not communicating with the backplane. The following procedure describes the steps to perform to respond to an Active LED error.

Step	Action
1	Make sure the NOE 771 module and the controller are installed properly.
2	Verify that the controller is functioning; if it is not, replace it.
3	If neither the new controller nor the NOE 771 module functions, replace the backplane.
4	Make sure that no more than two network option modules -- including NOE, NOM, NOP, and CRP 811 modules -- have been installed in the backplane with a 140 CPU 113 or 213; no more than six network option modules with a 140 CPU 424 or 534.
5	Check the version of the controller executive. You must have version 2.0 or greater to support the Ethernet module. Earlier versions do not recognize the module.
6	If steps 4 and 5 above check out ok, replace the NOE 771 module.

Procedure for Responding to a Ready LED Error Indicator

If the **Ready** LED fails to light, the NOE 771 module has failed internal diagnostic tests. The following procedure describes the steps to perform.

Step	Action
1	Make sure that power has been applied to the backplane.
2	If step 1 checks out ok, replace the NOE 771 module.

Procedure for Responding to a Link LED Error Indicator

If the **Link** LED fails to light, the NOE 771 module is not communicating with the Ethernet hub/switch. The following procedure describes the steps to perform to respond to a **Link** LED error.

Step	Action
1	Make sure that the cable has been installed correctly and the module is functioning properly.
2	Verify that the hub/switch is working properly.
3	If steps 1 and 2 check ok, replace the NOE 771 module.

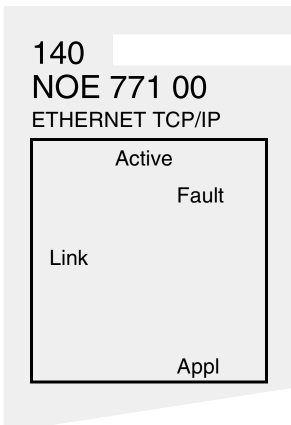
Kernel LED Error

The following table describes the **Kernel** LED errors that may occur and how to respond to them.

If	Then
The **Ready** LED is on and the **Kernel** LED is flashing	the module has detected an invalid software image.
The **Ready** LED is on and the **Kernel** LED is shining steadily,	an attempt to download a software image has failed and the module is in kernel mode.
Either of the above conditions exists.	download a new NOE Exec See *Establishing a Connection with an Ethernet Module, p. 217*.

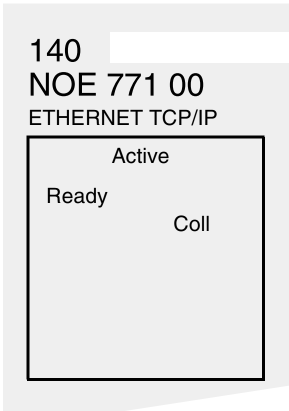
Fault LED

The **Fault** LED will flash briefly following an error as the module attempts to recover. The following figure shows the **Fault** LED.



Collision LED Error

If the twisted pair cable has not been connected properly, the **Coll** LED will shine steadily and the **Link** LED will be extinguished. (This condition does not occur with fiber optic modules.)
The following figure shows the Collision LED.



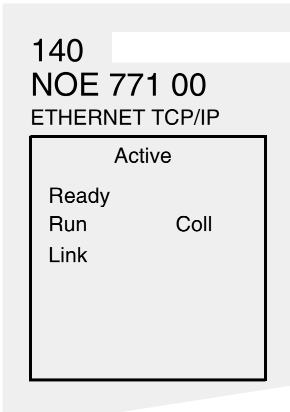
Procedure for Responding to a Collision LED Error

If the Collision LED fails to light, use the following procedure.

Step	Action
1	Make sure that the cable has been installed properly and is working properly.
2	Verify that the Ethernet hub/switch is functioning properly.

Collision LED Normal Condition

If the **Coll** LED is flashing, the module is reporting collisions on the Ethernet network. While such collisions are normal, the frequency of the flashes is an indication of the volume of traffic on the network. The flashes may be so frequent that the LED appears to be shining steadily. Heavy traffic will slow communications. If response time is important to your application, you should consider segmenting your network to reduce the frequency of collisions.
The following figure shows the Collision LED under normal conditions.



Run LED

The following table describes the action to be taken if the **Run** LED is flashing. The action depends on the number of flashes in sequence.

# of Flashes in Sequence	Action
Three	Check Ethernet connection.
Four	Change IP address
Five	Provide IP address
Six	Connect using default IP address and configure
Seven	Download NOE Executive

Application LED

If the module crashes, it will note the reason in a log. If the module is able to recover, the **Appl** LED will light, indicating that an entry has been made in the crash log. To learn how to read and clear the crash log, refer to *Reading and Clearing the Crash Log, p. 255*.

Reading and Clearing the Crash Log

Overview

The following information describes the crash log.

Introduction

The crash log provides you with the ability to capture conditions that lead to an anomalous condition. By providing the crash log to Schneider Electric technical support, you can facilitate their assistance in resolving your problems.

> **Note:** The crash log is provided with the understanding that, with a complex product in thousands of customer applications, there may be conditions that require advance diagnostics. The crash log is one of the tools used to solve complex problems.

The Crash Log

If the **Appl** indicator is lit, entries have been made in the crash log. The log may hold up to 64K of entries.

Reading the Crash Log

The crash log can be read from the Embedded Web Pages (see *Embedded Web Pages, p. 157*) or via FTP.

Procedure for Reading the Crash Log via FTP

The following procedure describes the steps to perform to access the crash log via FTP.

Step	Action
1	Log into the module's FTP Server
2	Change the directory to *wwwroot/conf/diag*
3	Perform an FTP to get the crash log: get crash log

Clearing the Crash Log

The crash log can be cleared from the Embedded Web Pages (see *Embedded Web Pages, p. 157*) or via FTP.

Procedure for Clearing the Crash Log via FTP

The following procedure describes the steps to perform to access the crash log via FTP.

Step	Action
1	Log into the module's FTP Server.
2	Change the directory to *wwwroot/conf/diag.*
3	Perform an `FTP rm crash.log` to delete the crash log file.

Downloading a New NOE Exec

Introduction

The following tools can be used to download a new NOE Exec:
- Concept EXECLoader (see *The Concept EXECLoader, p. 257*)
- FTP (see *Downloading a New NOE Exec via FTP, p. 261*)

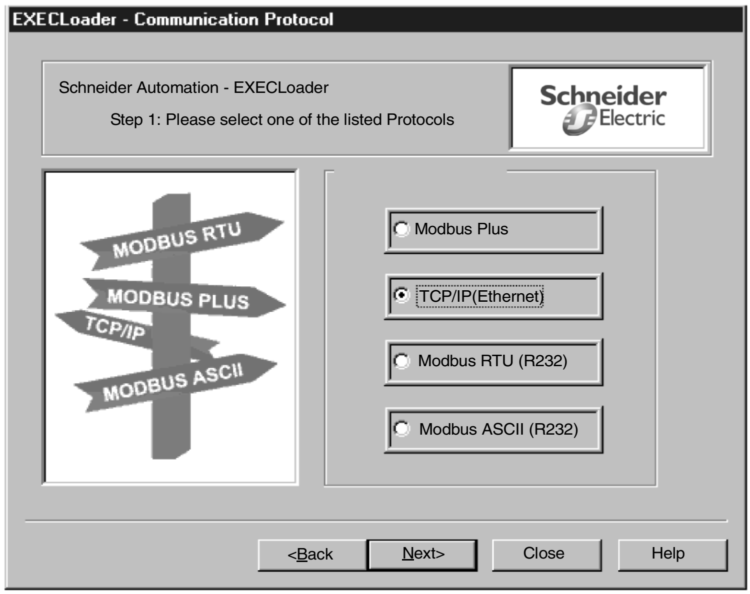
The Concept EXECLoader

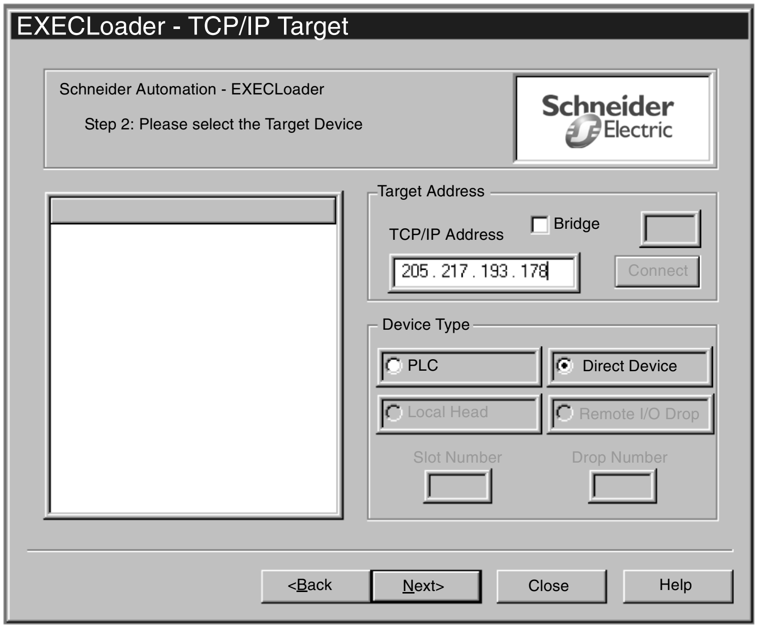
Overview

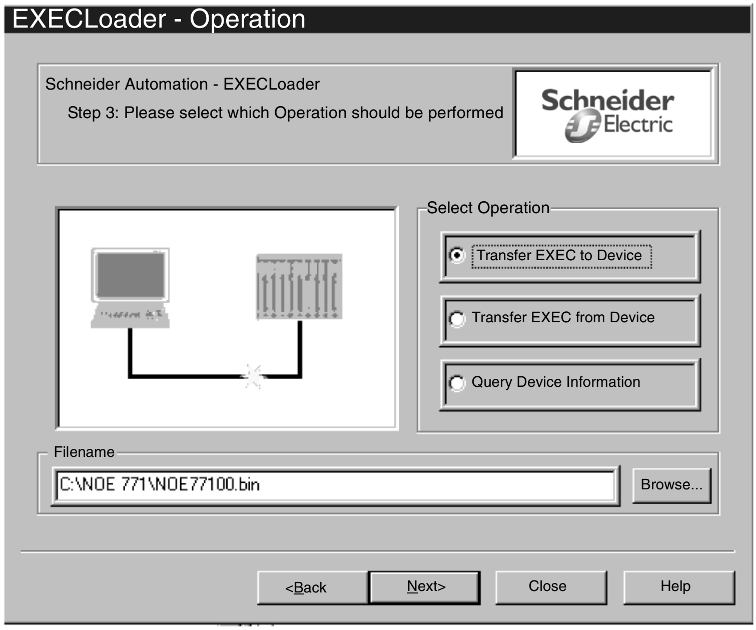
This section describes the use of the Concept EXECLoader, for downloading a new NOE Exec.

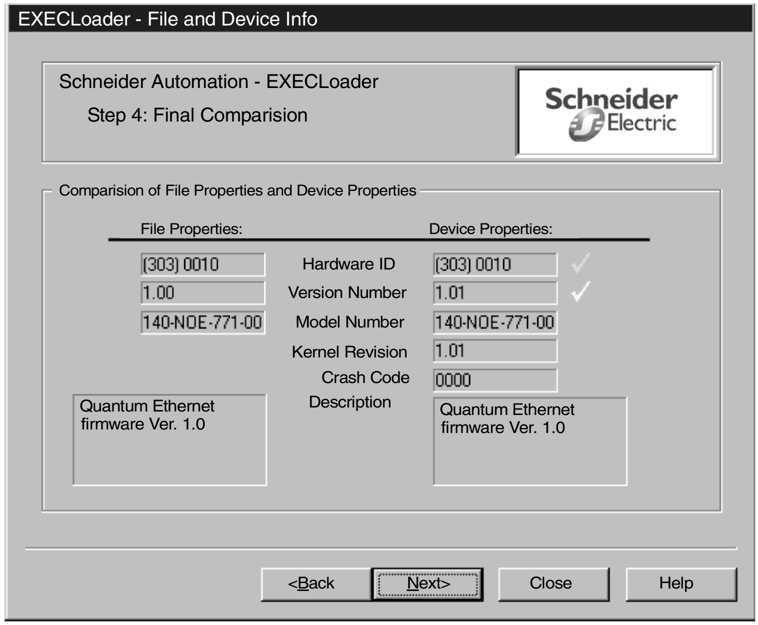
Procedure for Downloading NOE Exec

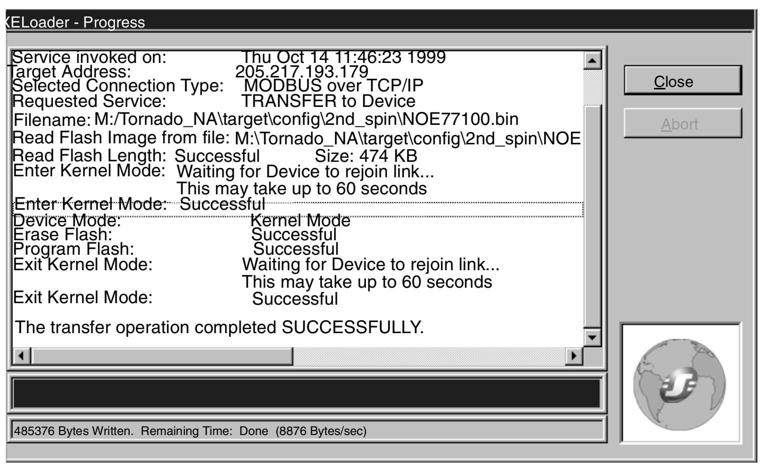
The following procedure provides the steps involved in downloading a new NOE Exec, using the Concept EXECLoader.

Step	Action
1	Activate the Exec Loader program.
2	Click on the **Next** button to start the Exec Loader process. **Result:** As shown in the following figure, the EXECLoader - Communication Protocol screen is displayed.  Click **TCP/IP [Ethernet**.}

Step	Action
3	Click on the **Next** button **Result**: As shown in the following figure, the EXECLoader - TCP/IP Target screen is displayed.  Click on **Direct Device**.

Step	Action
4	Click on the **Next** button. **Result**: As shown in the following figure, the EXECLoader - Operation screen is displayed.  Click on **Transfer EXEC to Device**.
5	Use Browser to select the file name.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>6</td><td>Click on the Next button.
Result: The EXECLoader - File and Device Info screen is displayed.


</td></tr>
<tr><td>7</td><td>Click on the Next button.
Result: As shown in the following figure, The EXECLoader - Progress screen is displayed.


When the process is completed, you can click the Close button.</td></tr>
</table>

Downloading a New NOE Exec via FTP

Procedure

The following procedure describes the steps to use to download a new NOE Exec via FTP. An example follows the procedure.

Step	Action
1	At the DOS prompt, type `FTP`, followed by the IP Address and press **Enter**.
2	At the User prompt, type: `USER` and press **Enter**.
3	At the password prompt, enter your FTP Password and press **Enter**.
4	At the FTP prompt, type `cd wwwroot/conf/exec` and press **Enter**.
5	At the FTP prompt, type `put` and press **Enter**.
6	At the local file prompt, type `NOE77100.bin` and press **Enter**.
7	At the remote file prompt, type `NOE77100.bin` and press **Enter**.
8	After the transfer is complete you must reboot the NOE to allow the new EXEC to become operational. **Note:** The file name is case sensitive and must be entered with the name in uppercase and the extenion in lowercase as shown in the figure below. For example: `NOE77100.bin`

Sample FTP Session

Note: The NOE Kernel can not be downloaded via FTP.

The following FTP session was used to download an NOE Exec.

Command Prompt - ftp 205.217.193.173

```
C:\noe77100>ftp  205.217.193.173
331 Password required
Password:
230 User logged in
ftp> cd wwwroot/conf/exec
250 Changed directory to "/FLASH0/wwwroot/conf/exec"
ftp> put
(local-file) NOE77100.bin
(remote-file) NOE77100.bin
200 Port set okay
150 Opening BINARY mode data connection
226 Transfer complete
485376 bytes sent in 3.06 seconds (158.41 Kbytes/sec)
ftp> dir
200 Port set okay
150 Opening BINARY mode data connection
-rwx---A-- 1 user                     2 kerVer
-rwx---A-- 1 user                485376 NOE77100.bin
226 Transfer complete
86 bytes received in 0.01 seconds (8.60 Kbytes/sec)
ftp> _
Connected to 205.217.193.173.
220 VxWorks FTP server (VxWorks 5.3.1) ready.
User (205.217.193.173:(none)): USER
```

Reboot Information after FTP

CAUTION

An EXEC update via File Transfer Protocol (FTP) may cause a crash to the system.

Be sure to Reboot the module after downloading by FTP.

Failure to follow this precaution can result in injury or equipment damage.

Downloading a New NOE Kernel

Overview

Version 2.00 of the NOE Executive adds a new feature to allow the updating of the low level Kernel within the NOE 771x0's firmware.   The following procedure should be followed to assure the proper installation of new Kernel firmware.

Note: The NOE Kernel can not be downloaded via FTP.

CAUTION

NOE Operation

Failure to perform this operation will render the NOE inoperable.

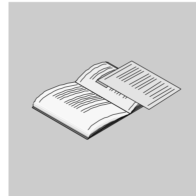
Failure to follow this precaution can result in injury or equipment damage.

Procedure

Follow these steps to download a NOE Kernel.

Step	Action
1	Check the current version of the NOE's Executive firmware following the instructions as defined in the section on 'Downloading a New NOE Exec'.
2	If the Executive is not at 2.00 or above the Executive must be updated first. After loading the new Executive and before loading the Kernel, make sure to cycle power to the NOE. See sample EXECLoader screen below:
3	Loading the Kernel firmware is performed using the EXECLoader in the same manner as the Executive firmware.
4	After the transfer comes up as Successful, the NOE needs approximately. 1 minute to burn the new Kernel into the NOE's flash and will then go through a reboot sequence.

Appendices



At a Glance

Introduction

The Appendices provide supplementary reference information for the Quantum NOE 771 xx series of modules.

What's in this Appendix?

The appendix contains the following chapters:

Specifications

A

Specifications

Overview

The following information describes the main specifications for the Quantum 140 NOE 771 xx Ethernet Module.

Specification Table

The main specifications for the Quantum 140 NOE 771 xx Ethernet Module are described in the following table

Communication Ports	**One auto-sensing 10/100Base-T shielded twisted pair (RJ-45 connector) port and one 100Base-FX (MT-RJ connector) port. Both ports transmit and receive Modbus commands encapsulated in TCP/IP protocol**
Bus Current Required	750 mA
Power Dissipation	3.8 W
Fuse	None
Programming Software	
Type and version	Concept, Ver. 2.2, or higher
	Modlink, Ver. 2.0, or higher
	Modsoft, Ver. 2.6, or higher
	ProWORX NxT, Ver. 2.1, or higher
Firmware	
CPU Type and version	Quantum Executive, Ver. 2.0, or higher
NOE Upgradeable	Field Upgradeable via FTP or Programming Panel.
Operating Conditions	
Temperature	0 to +60° C
Humidity	0 to 95% Rh non condensing @ 60° C
Altitude	15,000 ft (4500 m)

Communication Ports	**One auto-sensing 10/100Base-T shielded twisted pair (RJ-45 connector) port and one 100Base-FX (MT-RJ connector) port. Both ports transmit and receive Modbus commands encapsulated in TCP/IP protocol**
Vibration	10-57 Hz @ 0.0075 mm d.a
	57-150 Hz @ 1 g
Storage Conditions	
Temperature	-40 to +85°C
Humidity	0 to 95% Rh non condensing @ 60°C
Free Fall	1 m unpackaged
Shock	3 shocks / axis, 15 g, 11 ms

Ethernet Developers Guide

B

At a Glance

Introduction

This chapter contains information to assist ethernet developers.

What's in this Chapter?

This chapter contains the following topics:

Overview

Introduction

This appendix describes the design of the sample TCP/IP application named Network Options Ethernet Tester (NOET). The NOET application is a multiple document interface windows application that verifies the installation of the Quantum Ethernet TCP/IP module and also serves as a sample application for developers. TCP/UDP system port number 502 is used with ASA protocol_id of 0.

References

Inside Visual C++, Second Edition, David J. Kruglinski
Window Sockets, An Open Interface for Network Programming under Microsoft® Windows Version 1.1

What the Sample Application Does

The sample application performs the following calls:
- Calls the window socket function **socket** to create a socket.
- Calls the window socket function **setsockopt** to set the socket attributes.
- Calls the window socket function **connect** to establish a connection.
- Calls the window socket function **send** to transmit the request to the remote node.
- Calls the window socket function **recv** to receive the response from the remote node.
- Calls the window socket function **closesocket** to close the connection and release the socket.

In addition, the NOET encodes the request. The request consists of a header followed by a Modbus message.
As shown in the following table, the header contains an invoke identifier, a protocol type, the command length, and a destination identifier

Invoke Identifier	Protocol Type	Command Length	Destination ID	Modbus Message

The winsock.lib import library provided by the installation is used to link the window socket calls.

Development Environment

The sample application was developed with Microsoft Visual C++, version 1.52. The sample application uses Microsoft Foundation Class. The initial application was generated by the Visual C++ application wizard.

Class Descriptions

Overview

The following information describes each class.

List of Classes

- **CSample_app.** The Csample_app is the application class. This application was generated by the application wizard, and the source is in the file sam_app.cpp. The class declaration is in sam_app.h.
- **CMainFrame.** The CMainFrame is derived from the MFC class CMDIFrameWnd and is the application's main window frame. The source for CMainFrame is in mainfrm.cpp, and the declaration is in mainfrm.h. The code for CMainFrame was initially generated by the application wizard, and was modified to process window timer messages.
- **CSample_doc.** The CSample_doc is the document class. The declaration is in sam_doc.h and the implementation is in sam_doc.cpp.
- **CSample_View.** The CSample_View is the view of the document. It is derived from the CScrollView class. The declaration is in the sam_vw.h class, and it is implemented in the sam_vw.cpp, disp.cpp, tcp_hlp.cpp, and the tx_rx.cpp files.
- **CIP_dig**. The CIP_dlg class is the dialog class for getting the IP address. It is derived from the CDialog class. The declaration is in the cip_dlg.h file and the implementation is in the cip_dlg.cpp file. Both of these files were generated by The Visual C++ class wizard.
- **ClrStatsDlg.** The ClrStatsDlg class is the dialog class for clearing statistics. It is derived from the CDialog class. The declaration is in the cstatdlg.h file and the implementation is in the cstatdlg.cpp. Both of these files were generated by The Visual C++ class wizard.
- **GetStatsDlg.** The GetStatsDlg class is the dialog class for get statistics. It is derived from the CDialog class. The declaration is in the gstatdlg.h file and the implementation is in the gstatdlg.cpp file. Both of these files were generated by The Visual C++ class wizard.
- **CPollDlg.** The CPollDlg class is the dialog class for determining the poll period. It is derived from the CDialog class. The declaration is in the polldlg.h file, and the implementation is in the polldlg.cpp file. Both of these files were generated by The Visual C++ class wizard.
- **CReadDlg.** The CReadDlg class is the dialog class for determining the registers to read. It is derived from the CDialog class. The declaration is in the readdlg.h file, and the implementation is in the readdlg.cpp file. Both of these files were generated by The Visual C++ class wizard.
- **CWriteDlg.** The CWriteDlg class is the dialog class for determining the registers to write and the write data. It is derived from the Cdialog class. The declaration is in the writedlg.h and the implementation is in the writedlg.cpp file. Both of these files were generated by The Visual C++ class wizard.
- **CAboutDlg**. The CAboutDlg class is the dialog class for about. Both the declaration and its implementation are in the sam_app.cpp file.

The CSample_doc Class

Overview

The following information describes the document class.

Description

The CSample_doc (the document class) contains the user data used by the CSample_View class. The user data consists of the remote node's IP address, the transaction type, and its associated values. The different transaction types are read register, write register, clear statistics, and get statistics. In addition to the transaction type and the associated values, the document class also contains the poll interval.

A user modifies the user data via a menu or tool bar. The CSample_doc processes the menu or tool bar window command message by invoking the corresponding dialog. The state of the various menu items and tool bar buttons depends on the connection state between the application and the remote node. The CSample_View class maintains the connection state, and hence sets the state of the menu items and tool bar buttons.

The CSample_View Class

Overview

The following information describes the CSample_View class.

What It Does

The CSample_View class manages the TCP/IP connection, sends requests to remote nodes, and displays either connection state, or the results of a transaction. In addition, it sets the states of the tool bar buttons and menu items.

Accessing TCP/IP

The CSample_View interfaces with window sockets via its application programming interface, and via messages sent by the window sockets DLL to the CSample_View window. The reference for the window socket API is given above. The first call made to the window sockets DLL must be WSAStartup. This call is made by InitInstance member function of the CSample_app class. The last call to the window socket DLL must be WSACleanup. This call is made by the ExitInstance member function of the Csample_app class.

The CSample_View allocates and sets the socket attributes. The attributes it sets are:

- Set Linger to cause a hard close
- Receive out of band data in the normal data stream
- Disable Nagel algorithm for send coalescing

When the Nagel algorithm is disabled, if the stack receives an application message, it will immediately pass the message to the application and will send a TCP/IP acknowledgment message. Although this can generate more traffic, the application receives the message sooner than if the Nagel algorithm is enabled. The member function tcpip_setsocket_options sets the socket attributes.

The window socket interface provides the WSAAsyncSelect function which notifies the window of network events. The member function tcpip_setsocket_options calls WSAAsyncSelect function.

The following tables describes the different events.

Event	Description
FD_READ	A socket can read data
FD_WRITE	A socket can write data
FD_OOB	A socket can read out-of-band data
FD_CONNECT	A connect response has been received
FD_CLOSE	The connection has been closed

One of the parameters to the WSAAsyncSelect is a user defined message the window socket DLL sends to the window. The sample application user message is WM_TCPIP_EVENT and is defined in the file wn_msh.h. MFC architectural framework calls the CSample_View tcpip_event member function to process this message. As with all functions that process messages, tcpip_event parameters are a word and a long word. The word parameter is the socket, and the long word parameter contains the network event, and an error code.
Tcpip_event examines the network event and calls the member function indicated in the following table.

Network Eventt	**Member Function**
FD_READ	OnTcpIpRead(
FD_WRITE	OnTcpIpWrite()
FD_OOB	OnTcpIpOob(
FD_CONNECT	/OnTcpIpConnect
FD_CLOSE	OnTcpIpClose()

Application Message Format

TCP/IP transmits a message as a stream. There is no indication of the start of a message nor the end of the message. The NOE option module adds a header to determine the message boundaries. The message is a Modbus message. The header contains the following fields.
The header contains the following fields.

- Invoke Identifier. This two byte field associates a request with the response. The client application picks the invoke identifier, and server returns the same invoke identifier in the response.
- Protocol Type. This two byte field identifies the protocol type. Currently, the only protocol supported is Modbus.
- Command Length. This two byte field is the size of the rest of the message.
- Destination Identifier. This one byte field is reserved for future use.

The Modbus message follows the header. The message does not contain the address field; instead, the first byte is the Modbus function code.
The data structure for the header is declared in modbus.h and the CSample_View encode_header function encodes the header. The member functions are encode_clear_stats, encode_read_stats, encode

Timers and Transaction Processing

Overview

The following information describes how CSample_View responds to a timer message.

Timers

CSample_View requires to periodically receive a timer message. This message triggers the CSample_View to transmit a message. Since window timers are a limited resource, the window associated with CMainFrame class receives the timer messages. CMainFrame member AddTimerList function will place a window on its timer list. When CMainFrame processes the WM_TIMER message, it sends each window on its time list the user defined WM_POLL_INTERVAL message.
MFC calls CSample_View member OnInitalUpdate function when it is first being created. OnInitialUpdate calls CMainFrameís AddTimerList in order to receive the WM_POLL_INTERVAL message. MFC architectural framework calls CSample_View OnPollInterval member function to process this message.

Transaction Processing

CSample_View transaction processing consists of establishing a connection, transmitting the request, receiving the response, and displaying the response. CSample_View uses both a transmit and a receive state machine to advance a transaction.

Transmit State Machine

Overview

The following information describes the states for the transmit state machine.

Description

The following list contains the different states for the transmit state machine:

- IDLE. In the IDLE state, there is no connection.
- RESOLVING_NAME. In the RESOLVING_NAME state, CSample_View is waiting for the window socket DLL to convert a node's name into an IP address.
- CONNECTING. In the CONNECTING state, CSample_View is waiting for the window socket DLL to generate the FD_CONNECT event. This event indicates if the attempt to establish a connection succeeded or failed.
- CONNECTED. The CONNECTED state indicates that a connection has been successfully established.
- WAIT_TO_TX. In the WAIT_TO_TX state, CSample_View is waiting to transmit the message. It transmits the message, when the time from the last transmit exceeds the specified poll interval.
- BLOCKED. When CSample_View attempts to send a message, the window socket DLL may not be able to transmit the complete message. This is a flow control condition, and CSample_View enters the BLOCKED state. The window socket DLL generates the FD_WRITE event when it can send more data.
- TX_DONE. CSample_View enters the TX_DONE when it has completed transmitting the request.

If the CSample_View is in the IDLE state, and user selects either the connect menu item or the connect tool bar button, CSample_View OnManagConnect function attempts to connect with its tcpip_initate_connection function. This function examines the remote destination and determines if it's a name or an IP address. If it's a name, OnManagConnect changes the transmit state to RESOLVING_NAME, and it invokes the window sockets DLL WSAAsyncGetHostByName function to resolve the name. Window sockets DLL will generate the user defined WM_TCPIP_NAME_RESOLVED message, which indicates if the name has been resolved. The OnTcpIpNameResolved member function processes the WM_TCPIP_NAME_RESOLVED message. If the name is not resolved, OnTcpIpNameResolved changes the transmit state back to IDLE.

If the remote node is an IP address or if it's a name that has been resolved, then CSample_View tcpip_connect_rq function is called to initiate a connect request to the remote node. The listen port for the connect request is five hundred and two, and is defined by the constant MBAP_LISTEN_PORT in modbus.h. If tcpip_connect_rq succeeded in initiating a connect request, then tcpip_connect_rq changes the transmit state to CONNECTING; otherwise it changes the transmit state to IDLE.

The window sockets DLL generates an FD_CONNECT event which indicates if the connect request succeeded or failed. CSample_View OnTcpIpConnect function processes the FD_CONNECT event. If the connect request succeeded, OnTcpIpConnect changes the transmit state to CONNECTED, otherwise it changes the state to IDLE.

Recall that MFC architectural framework calls CSample_View OnPollInterval member function to process WM_POLL_INTERVAL message sent as result of CMainFrame class processing a WM_TIMER message. OnPollInterval examines the transmit state. If the transmit state is CONNECTED and the user has selected a transaction type, then OnPollInterval calls CSample_View TransmitUserRequest function.

TransmitUserRequest encodes a request based on the transaction type, saves the current time, and calls CSample_View TransmitMessage function. OnPollInterval uses the saved time to determine when to transmit the next request. Transmit-Message attempts to send a message to the remote side. To send the message, TransmitMessage enters a loop. In the body of the loop, transmit message calls the window socket DLL send function.

The following list describes the outcomes of the send function and the actions taken.

- The message was sent successfully. TransmitMessage changes the transmit state to TX_DONE and exits the loop.
- Only part of the message was sent. TransmitMessage reenters the loop.
- Send function returns an error indicating there is no buffer space within the transport system. TransmitMessage changes the transmit state to BLOCKED and exists the loop.
- Send function returns some other error. TransmitMessage closes the connection, changes the transmit state to IDLE, and exits the loop.

When buffer space within the transport system becomes available to transmit messages, the window socket DLL generates a FD_WRITE event. CSample_View OnTcpWrite function processes the FD_WRITE function by calling TransmitMessage.

The receive state machine (see *Receive State Machine, p. 278*) processes the response to a request. When the receive state machine has completed receiving the response, it changes the transmit state machine from the TX_DONE state to the WAIT_TO_TX state.

Recall that the TransmitUserRequest saves the time. CSample_View OnPollInterval uses this saved time to determine if a new request needs to be transmitted. OnPollInterval is called by MFC architectural framework to process the WM_POLL_INTERVAL sent when CMainFram class processes the window timer message, WM_TIMER. OnPollInterval examines the transmit state. If the transmit state is WAIT_TO_TX and the elapsed time from the previous transmit request exceeds the poll interval, then OnPollInterval calls TransmitUserRequest to start another transaction.

Receive State Machine

Overview

The following information describes the receive state machine.

Description

The following list contains the different states for the receive state machine:

- RX_HEADER. In the RX_HEADER state, the receive machine is receiving the message header.
- RX_BODY. In the RX_BODY state, the receive machine is receiving the response message associated with the requested transaction.
- DUMP_BODY. In the DUMP_BODY state, the receive message is receiving a message, but there is no associated transaction with respect to this message.

The window socket DLL generates the FD_READ event whenever there is data to be read. If only part of the data is read, it generates another event. CSample_View OnTcpIpRead function processes the FD_READ event and drives the receive state machine.

When an FD_READ event is generated, it is possible that the complete message is not present. The remote node may have attempted to send a 100 byte response, but the transport system may have only had buffer space to transmit three bytes. The receiver will get a FD_READ for the three bytes. OnTcpIpRead calls CSample_View rx_msg to read the receive data into the buffer. There are three parameters to rx_msg. The first parameter is a pointer to a receive buffer. The second input parameter is the receive size. The third parameter is both an input and output parameter. On both input and output, the third parameter is the number of bytes read. These parameters allow the processing of a partially received message.

The receive state machine maintains a variable that is the number of bytes received. Initially, the receive state machine is in the RX_HEADER state, and the number of bytes received is zero.

When OnTcpIpRead is called and the receive state is RX_HEADER OnTcpIpRead calls rx_msg with receive size equal to the header size. On return, OnTcpIpRead examines the number of bytes received. If the number of bytes received is not equal to the header size, then the receive machine remains in the RX_HEADER state, and OnTcpIpRead returns.

If, upon return, the number of bytes received is the same size as the header size, then the header has been received. OnTcpIpRead sets the number of bytes received to zero, and the receive size is obtained from the header. These two values will be used the next time rx_msg is called. OnTcpIpRead also obtains the transaction identifier and the protocol type from the header. If the transaction identifier matches the transmit request identifier and the protocol type is MODBUS, then OnTcpIpRead changes the receive state to RX_BODY. However if either transaction identifier does not match or the protocol is not MODBUS, then OnTcpIpRead changes the receive state to DUMP_BODY.

When OnTcpIpRead is called and the receive state is RX_BODY, OnTcpIpRead calls rx_msg with receive size equal to the value obtained from the header. On return, OnTcpIpRead examines the number of bytes received. If the number of bytes received is not equal to the receive size, then the receive machine remains in the RX_HEADER state, and OnTcpIpRead returns.

If upon return the number of bytes received is the same as the receive size, then OnTcpIpRead has read the response to a transaction. OnTcpIpRead saves the results and invalidates the client area which causes the results to be displayed. OnTcpIpRead also changes the transmit state to WAIT_TO_TX, and resets the state receive state machine by setting the state to RX_HEADER and the number of bytes received to zero. It then returns.

When OnTcpIpRead is called and the receive state is DUMP_BODY, OnTcpIpRead calls rx_msg with receive size equal to the value obtained from the header. On return OnTcpIpRead examines the number of bytes received. If the number of bytes received is not equal to the receive size, then the receive machine remains in the RX_HEADER state, and OnTcpIpRead returns.

If upon return the number of bytes received is the same as the receive size, the OnTcpIpRead has completed reading the message. Since this message does not correspond to an transaction, the only processing OnTcIpRead performs is resetting the receive state machine.

The member function rx_msg calls the window socket recv function to read data. The recv function either returns a non negative number that is the number of bytes read or it returns an error. If the number bytes read is zero, then the connection no longer exits, and rx_msg closes the socket, and sets the transmit state to IDLE. If the recv function returns the error indicating that no receive data is available, then rx_msg just returns. For any other recv function error, rx_msg closes the socket, and sets the transmit state to IDLE.

Displaying on the Screen

Overview

The following information describes the different types of displays and the CSample_View member functions for displays.

Description

CSample_View m_display member indicates the display type. The following list describes the different types of displays and the CSample_View member functions for showing the display:

- Displaying the connection state. The different connection states displayed are IDLE, RESOLVING NAME, and CONNECTING. ConnPaint member function displays the connection state.
- GetStatsPaint member function displays the results of a get statistics request.
- ClearStatsPaint member function displays the results of a clear statistics request.
- ReadRegPaint member function displays the results of a read register request.
- WriteRegPaint member function displays the results of a write register request.

MFC architectural framework calls CSample_View OnDraw member function to process the window WM_PAINT message. OnDraw examines m_display member variable and calls the corresponding member function described in the previous paragraph. Whenever CSample_View needs to display a result, it calls Cview Invalidate function which causes a WM_PAINT message.

CSample_View is derived from MFC CScrollView class. This class handles the scroll logic. To perform the scroll logic, CScrollView requires the size of the document. It is informed of the document size via its SetScrollSizes member function.

CSample_View UpdateScrollSizes member function based on the display type calculates the document size, and then calls SetScrollSizes. CSample_View calls UpdateScrollSizes if the display type changes or if the user changes the window size.

Quantum Ethernet TCP/IP Modbus Application Protocol

C

At a Glance

Introduction

This chapter describes the Quantum Ethernet TCP/IP Modbus Application Protocol.

What's in this Chapter?

This chapter contains the following topics:

Overview

Introduction

The following information describes the Modbus Application Protocol (MBAP).

The Modbus Application Protocol (MBAP) is a layer-7 protocol providing peer-to-peer communication between programmable logic controllers (PLCs) and other host-based nodes on a LAN. Collectively, these nodes implement all or part of a control application used for industrial automation applications in the automotive, tire and rubber, food and beverage, and utilities industries, to name a few.

Modbus protocol transactions are typical request-response message pairs. Modbus requests contain function codes representing several classes of service including data access, online programming, and program download and upload classes. Modbus responses can be ACKs with and without data, or NACKs with error information.

The Modbus Application Protocol can be transmitted over any communication system that supports messaging services. However, the current Quantum implementation transports Modbus Application Protocol PDUs over TCP/IP. Both Ethernet II and IEEE 802.3 framing are accommodated, although Ethernet II framing is the default.

For more information, consult the *Modbus Protocol Reference Guide* (PI-MBUS-300).

Modbus Application Protocol PDU

Overview

The following information describes the structure and content of the Modbus Application Protocol PDU.

Description

The Modbus Application Protocol PDU, mbap_pdu, is received at TCP port number 502. The current maximum size of the mbap_pdu for this class of services is 256 bytes. The structure and content of the mbap_pdu is defined to be:

```
mbap_pdu::={inv_id[2], proto_id[2], len[2],dst_idx[1],
data=mb_pdu}
```

The header is seven bytes long and includes the fields listed in the following table:

Field	Description
inv_id	[2 bytes] invocation id used for transaction pairing
proto_id	[2 bytes] used for intra-system multiplexing, default is 0 for Modbus services
len	[2 bytes] the len field is a byte count of the remaining fields, and it includes the dst_id and data fields

The remainder of the pdu includes two fields:

Field	Description
dst_idx	[1 byte] destination index is used for intra-system routing of packets (currently not implemented)
data	[n bytes] this is the service portion of the Modbus pdu, mb_pdu, and it is defined below

The service portion of the Modbus Application Protocol, called mb_pdu, contains two fields:

```
mb_pdu::={func_code[1], data[n]}
```

The following table describes the fields in mb_pdu.:

Field	Description
func_code{1 byte	Modbus function code
data	[n bytes] this field is function code dependent and usually contains information such as variable references, variable counts, and data offsets

The size and content of the data field are dependent on the value of the function code.

Example

Here are the values for a sample mbap_pdu for reading a register:

`00 01 00 00 00 06 01 03 00 00 00 01`

The following table shows the structure and content for this example:

inv_id	00 01	
	proto_id	00 00
	len	00 00
	dst_idx	01
	func_code	03
	data	00 00 00 01

Modbus Application Protocol Service Classes

Introduction

There are several classes of service that are part of the Modbus Application Protocol. Each of these classes is described below.

Data Access

Read/write both discrete and analog data values from PLC register files.

Online Programming

Services make relatively minor alterations to ladder logic programs with a highly controlled introduction of these changes into the executing program.

Image Download/ Upload

Image download services support the downloading of a ladder logic control program to the PLC. Image upload services support the uploading of a ladder logic control program from a PLC to PC host for archival/backup purposes.

Configuration

Configuration services allow the user to define parameter values which affect the PIC’s register files, I/O map, communication port configuration and scan attributes, to name a few.

Device Execution State Control

The class of service allows the user to start/stop the PLC scan execution. These services require the user to be in an application login context which is obtained through other Modbus services.

Modbus Application Protocol PDU Analysis

Oveview

The following information provides an analysis of the Modbus Application Protocol.

Analysis

The Modbus Application Protocol PDU is transmitted over a TCP/IP Ethernet stack. Both Ethernet II and IEEE 802.3 framing will be accommodated. Ethernet II framing is the default.

```
. . .   from the wire in for IEEE 802.3 framing   . . .
. . .  is IEEE 802.3 framing if length <=1500   . . .802.3_pdu
::= {dst_addr[6], src_addr[6], length[2], data=802.2_pdu}*an
IEEE 802.3 PDU has a maxFrameSize of 1518 octets
*an IEEE 802.3 PDU has a minFrameSize of 64 octets802.2_pdu :
{dsap[1], ssap[1], frm_cntrl[1], snap_hdr[5], data=ip_pdu}
*the snap_hdr is associated with a "well-known" 802.2 sap
snap_hdr
 ::={org_code[3], ethertype[2] }

*the snap hdr (sub network access protocol) allows the older
style
Ethernet protocols to run on the newer IEEE 802.2 interface.
The
ethertype parameter indicates the service, ex. ip or arp.  IP
has a value
0x800.. . .   from the wire in for Ethernet II framing   . . .
. . .  is Ethernet II framing if length >1500   . . .802.3_pdu
::= {dst_addr[6], src_addr[6], length[2], data=ip_pdu}. . .
the common part of the packet begins here   . . .ip_pdu ::=
{ip_hdr[20], data=tcp_pdu}tcp_pdu ::= {tcp_hdr[24],
data=appl_pdu=mbap_pdu}
```

The mbap_pdu is the Modbus Application Protocol whose messages are received at a well-known port. The current maximum size of the mbap_pdu for this class of services in 256 bytes.

Structure and Content

The structure and content of the mbap_pdu is defined to be:

```
mbap_pdu ::={ inv_id[2], proto_id[2], len[2], dst_idx[1],
data=mb_pdu }The header is 7 bytes long, and includes the
following fields:
inv_id[2 bytes] invocation id used for transaction pairing
proto_id[2 bytes] used for intra-system multiplexing,default
is 0 for Modbus
serviceslen[2 bytes] the len field is a byte count of the
remaining fields and
includes the dst_id and data fields.
```

The remainder of the pdu includes two fields:

```
dst_idx[1 byte] destination index is used for intra-system
routing of
packets. (currently not implemented)data[n bytes] this is the
service portion of the Modbus pdu, mb_pdu, and is
defined below
```

The service portion of the Modbus Application Protocol, called mb_pdu, contains 2 fields:

```
mb_pdu ::= { func_code[1], data[n] }
func_code[1 byte] MB function codedata[n bytes] this field is
function code dependent and usually contains
information such as variable references, variable counts, and
data offsets.
```

The size and content of the data field are dependent on the value of the function code.

TCP/IP Specific Issues

Overview

The following information describes some TCP/IP specific issues.

Broadcast/ Multicast

Although broadcast and/or multicast are supported by both IP network address and IEEE 802.3 MAC address, the Modbus Application Protocol does not support either broadcast or multicast at the application layer.
Schneider Electric's Quantum PLCs use broadcast addressing because they use ARP as the means of locating the destination node. The client interface to the Modbus Application Protocol service on the PLC, the MSTR block, requires the user to provide the destination IP address. Also the embedded stack does use a pre-configured default gateway IP address in the case where ARP does not succeed.

TCP Port Number

Schneider Electric has obtained a well-known system port from an Internet Authority. Schneider Electric's well-known system port number is 502. The Internet Authority assigned the system port number 502 to asa-appl-proto with Dennis Dubé as the company point of contact.
This port number allows Schneider Electric to transport various application protocols over with TCP or UDP. The particular protocol is indicated by the value of the proto_id parameter in the mbap_pdu. Currently the only assignment is 0 meaning Modbus Application Protocol.

Reference Documents

Overview

The following information provides a list of reference documents that you may find helpful.

Introduction

Following is a list of related documentation.

- ANSI/IEEE Std 802.3-1985, ISO DIS 8802/3, ISBN - 0-471-82749-5, May 1988
- ANSI/IEEE Std 802.2-1985, ISO DIS 8802/2, ISBN 0-471-82748-7, Feb 1988
- RFC793, TCP (Transmission Control Protocol) DARPA Internet Program Protocol Specification, Sep 1981
- RFC 791, IP (Internet Protocol) DARPA Internet Protocol Specification, Sep 1981
- RFC826, An Ethernet Address Resolution Protocol (ARP), David Plummer, NIC Sep 1982
- RFC1042, A Standard for the Transmission of IP Datagrams over IEEE 802.2 Networks, Postel & Reynolds, ISI, Feb 1988
- RFC 792, ICMP (Internet Control Message Protocol) DARPA Internet C Control Message Protocol Specification, Jon Postel, Sep 1981
- RFC951, BOOTSTRAP PROTOCOL (BOOTP), Bill Croft and John Gilmore, September 1985
- RFC783, The Trivial File Transfer Protocol (TFTP) rev 2, K.R. Solons MIT, June 1981

NOE 771 -00, -01, and -11 Modules I/O Scanner Performance Statistics

D

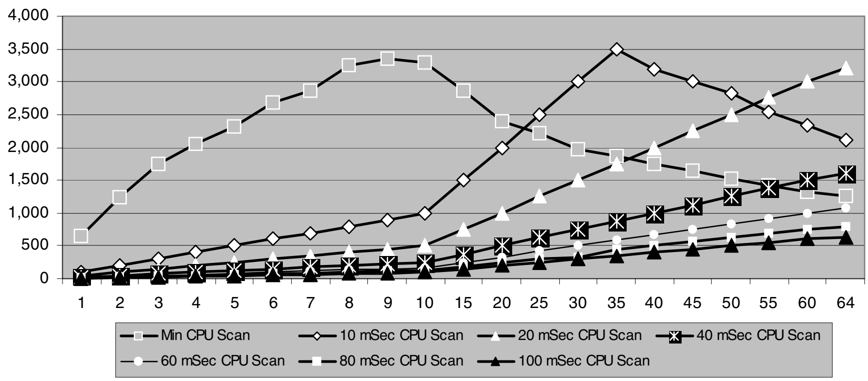
140 NOE 771 -00, -01, and -11 I/O Scanner Performance

Overview

The following information describes the performance of the 140 NOE 771 -00, -01, and -11 I/O Scanner with various Quantum CPUs.

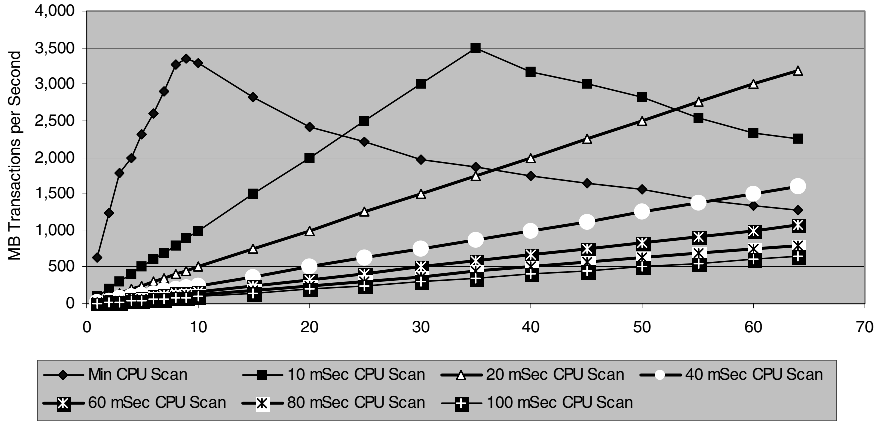
Quantum 113 CPU

The following figure shows the performance of the 140 NOE 771 -00, -01, and -11 I/O Scanner with Quantum 113 CPU.



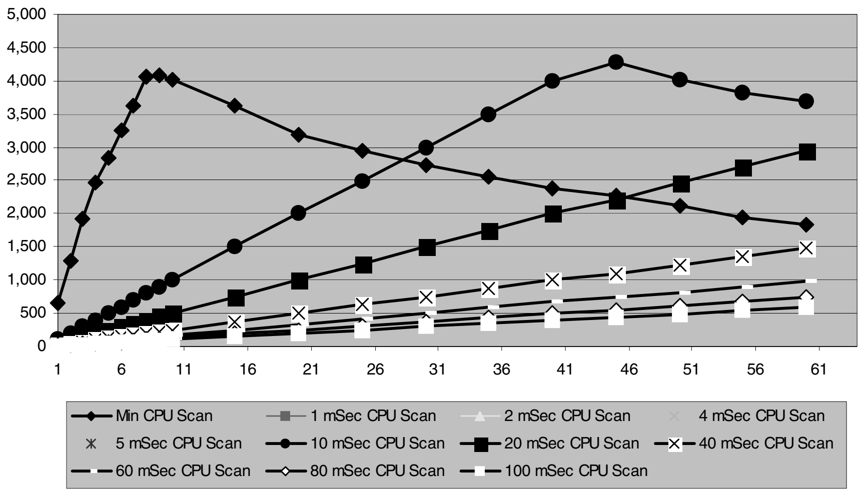
Quantum 213 CPU

The following figure shows the performance of the 140 NOE 771 -00, -01, and -11 I/O Scanner with Quantum 213 CPU.



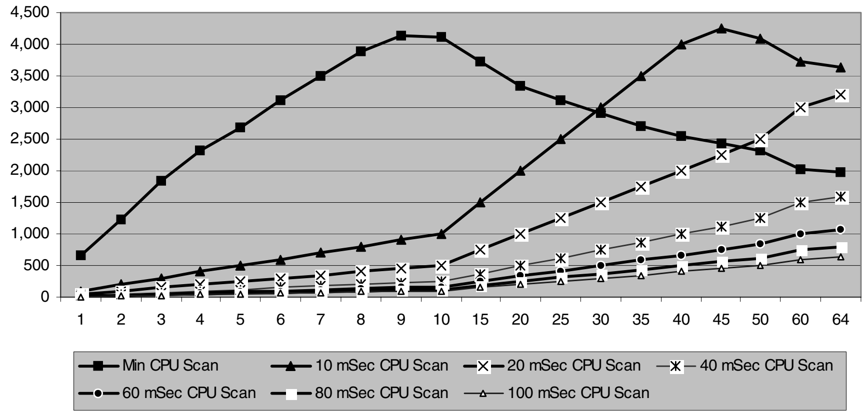
Quantum 424 CPU

The following figure shows the performance of the 140 NOE 771 -00, -01, and -11 I/O Scanner with Quantum 424 CPU.

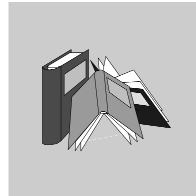


Quantum 534 CPU

The following figure shows the performance of the 140 NOE 771 -00, -01, and -11 I/O Scanner with Quantum 534 CPU.



Glossary



A

ACK Acknowledgement

address On a network, the identification of a station. In a frame, a grouping of bits that identifies the frame's source or destination.

API Application Program Interface. The specification of functions and data used by one program module to access another; the programming interface that corresponds to the boundary between protocol layers.

ARP Address Resolution Protocol. A network layer protocol used to determine the physical address which corresponds to the IP address for a host on the network. ARP is a sub-protocol which operates under TCP/IP.

ASN.1 Abstract Syntax Notation. Grammar used to define a protocol (OSI scope)

B

Backplane A metal plate with a bus-bar and couplers. Use the backplane to attach a module and make a PLC bus connection.

BOOTP BOOTstrap Protocol. A protocol used at power-up in order to get an IP address which is provided by a BOOTP server and is based on the module's MAC address.

bps Bits per second.

bridge A device that connects two or more physical networks which use the same protocol. Bridges read frames and decide whether to transmit or block them based on their destination address.

BSP Board Support Package. A software package that maps a specific real-time operating system (RTOS0 onto a specific hardware.

C

CHS Hot Standby module -- provides fault tolerance for remote I/O by connecting two redundant NOE modules.

client A computer process requesting service from other computer processes.

Concept A software package that facilitates controller configuration.

Cyclic Data Exchange Provides data transfer between two or more NOE 771 xx controllers on a TCP/IP network.

D

default gateway The IP address of the network or host to which all packets addressed to an unknown network or host are sent. The default gateway is typically a router or other device.

DHCP Dynamic Host Configuration Protocol. An improved version of BOOTP.

DHCP Client Host on the network obtaining its configuration from a DHCP Server.

DHCP Server Server providing configuration parameters to a DHCP Client.

DNS Domain Name System. A protocol within TCP/IP used to find IP addresses based on host names

F

FactoryCast An embedded Web server which the user customizes, permitting user access to controller diagnostics and Ethernet configuration.

field A logical grouping of contiguous bits that convey one kind of information, such as the start or end of a message, an address, data, or an error check.

firewall A gateway that controls access to a network or an application.

frame A group of bits which form a discrete block of information. Frames contain network control information or data. The size and composition of a frame is determined by the network technology being used.

framing types Two common framing types are Ethernet II and IEEE 802.3.

FTP File Transfer Protocol. The protocol (over TCP) used to read or write a file into a remote station (the FTP server side).

G

gateway A device which connects networks with dissimilar network architectures and which operates at the Application Layer. This term may refer to a router.

Global Data (Publish / Subscribe) Service of inter PLC synchronization (shared databases).

H

half duplex (HDX) A method of data transmission capable of communication in two directions, but only one direction at a time.

host A node on a network.

hostname A domain name given to a specific computer on a network and used to address that computer.

HTTP A domain name given to a specific computer on a network and used to address that computer.

hub A device which connects a series of flexible and centralized modules to create a network.

I

I/O Drop One or two (depending on the system type) Remote I/O Channels consisting of a fixed number of I/O points.

I/O Map An area in the controller configuration memory used to map input and output points. Previously called traffic cop.

I/O Scan A procedure the processor follows to monitor inputs and control outputs.

I/O Scan List A configuration table which identifies the targets with which repetitive communication is authorized.

I/O scanner Software component which is in charge of scanning Ethernet based Momentum IO in order to get inputs and set outputs.

ICMP Internet Control Message Protocol. A protocol within TCP/IP used to report errors in datagram transmission.

Internet The global interconnection of TCP/IP based computer communication networks.

IP Internet Protocol. A common network layer protocol. IP is most often used with TCP.

IP Address Internet Protocol Address. A 32-bit address assigned to hosts using TCP/IP.

L

layer In the OSI model, a portion of the structure of a device which provides defined services for the transfer of information.

Legacy In the sense of network communication: Existing Components (PLC products etc.) that do not provide special (hardware) support for Control Intranet.

M

MAC Address Media Access Control address. The hardware address of a device. A MAC address is assigned to an Ethernet TCP/IP module in the factory.

MBAP Modbus Application Protocol

MIB Management Information Base. Database that holds the configuration of a SNMP enabled device.

MODBUS A communication system that links Modicon controllers with intelligent terminals and computers over common carrier or dedicated lines

Modsoft A software package that facilitates programming the NOE module.

MSTR A special master instruction which uses ladder logic to read and write controller information.

N

N_PDU Protocol Data Unit exchanged at layer N level (OSI model)

NACK Negative acknowledgment indicating an error.

NDDS Network Data Delivery Services.

network Interconnected devices sharing a common data path and protocol for communication.

node An addressable device on a communications network.

NOET Network Options Ethernet Tester

O

OIT / OID Object Information True / Object ID (identify OIT) Contain databases managing SNMP (MIBs)

OSI model Open System Interconnection model. A reference standard describing the required performance of devices for data communication. Produced by the International Standards Organization.

P

packet The unit of data sent across a network.

Peer Cop Software that allows you to configure data blocks to be transferred between controllers on a Modbus Plus network.

PING Packet Internet Groper. A program used to test whether a destination on a network can be reached.

PLC Programmable Logic Controller

port An access point for data entry or exit within a host using TCP services.

protocol Describes message formats and a set of rules used by two or more devices to communicate using those formats.

ProWORX NxT A software package that facilitates the use of the I/O Scanner to configure data blocks to be transferred between controllers on a TCP/IP network.

R

repeater A device that connects two sections of a network and conveys signals between them without making routing decisions or filtering packets.

RFC Request For Comment. Paper identified by a number in Internet world. They define the state of art regarding Internet protocols (ruled by IETF = Internet Engineering Task Force) http://www.ietf.org

router A device that connects two or more sections of a network and allows information to flow between them. A router examines every packet it receives and decides whether to block the packet from the rest of the network or transmit it. The router will attempt to send the packet through the network by the most efficient path.

S

server Provides services to clients. This term may also refer to the computer on which the service is based.

SNMP Simple Network Management Protocol

socket The association of a port with an IP address, serving as an identification of sender or recipient.

stack The software code which implements the protocol being used. In the case of the NOE modules it is TCP/IP.

STP Shielded Twisted Pair. A type of cabling consisting of several strands of wire surrounded by foil shielding, twisted together.

subnet A physical or logical network within an IP network, which shares a network address with other portions of the network.

subnet mask A bit mask used to identify or determine which bits in an IP address correspond to the network address and which bits correspond to the subnet portions of the address. The subnet mask is the network address plus the bits reserved for identifying the subnetwork.

switch A network device which connects two or more separate network segments and allows traffic to be passed between them. A switch determines whether a frame should be blocked or transmitted based on its destination address.

T

TCP Transmission Control Protocol.

TCP/IP A protocol suite consisting of the Transmission Control Protocol and the Internet Protocol; the suite of communications protocols on which the Internet is based.

Traffic Cop A Quantum software routine that facilitates the placement of an NOE 771 module into a specified location.

U

UDP User Datagram Protocol. A protocol which transmits data over IP.

Uni-Te Télémecanique unified application protocol (used in S7, Premium, and Micro PLC ranges).

URL Uniform Resource Locator. The network address of a file.

UTP Unshielded Twisted Pair. A type of cabling consisting of insulated cable strands which are twisted together in pairs.

W

Web Worldwide interconnection of stations based on Internet protocols. The most famous one is HTTP (Web server).

Winsock The Microsoft implementation of the Windows Sockets networking API based on the Berkeley UNIX Sockets interface for supporting TCP/IP.

WWW World Wide Web. A hypertext-based, distributed information system in which clients and servers are freely available.

Index

Numerics

F

G

H

I

L

M

N

P