Quantum NOE 771 ••

Ethernet Modules
User Guide

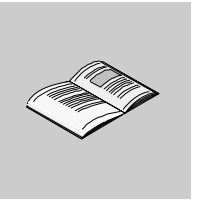
10/2009

31001913.06

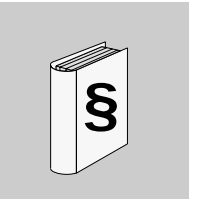
www.schneider-electric.com

Schneider Electric

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 or serious injury.

⚠ WARNING

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

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

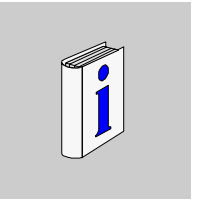
CAUTION
CAUTION, used without the safety alert symbol, indicates a potentially hazardous situation which, if not avoided, **can result in** equipment damage.

PLEASE NOTE

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

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

About the Book



At a Glance

Document Scope

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Design your application so that unmonitored modules support communication only to non-critical parts of the application.

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

This manual describes the functionality of the Quantum NOE 771 •• product line.

The following PLC modules are covered:
- **140 NOE 771 00**
- **140 NOE 771 01**
- **140 NOE 771 10**
- **140 NOE 771 11**

This manual provides 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 •• modules
- installation of the NOE 771 •• modules on a Quantum backplane

This manual describes the procedures for:
- configuring the modules from your programming panel using 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 FactoryCast 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 NOET with a Windows-based PC to monitor the network

Nomenclature

The following table describes the naming scheme.

140 NOE 771		Model Numbers
••	refers to	-00, -01, -10, -11
•0	refers to	-00, -10
•1	refers to	-01, -11
0•	refers to	-00, -01
1•	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 should have some knowledge about the use of PLC systems and a working knowledge of either the Concept, ProWORX NxT, or Modsoft programming tools. You also must 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 User Manual	840 USE 503
BooTP Lite Ethernet IP Address Download Utility Instructions	31002087
FactoryCast User Guide	31001229
Hot Standby Planning and Installation Guide	840 USE 106

Ladder Logic Block Library User Guide	840 USE 101
MODBUS Protocol Reference Guide	31002841
Open MODBUS Specification	www.modicon.com/openmbus
ProWORX NxT User Guide	372 SPU 680 01
ProWORX 32 User Guide	372 SPU 780 01
Remote I/O Cable System Planning and Installation Guide	890 USE 101

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

Product Related Information

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

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

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

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

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

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

User Comments

We welcome your comments about this document. You can reach us by e-mail at techcomm@schneider-electric.com.

Product Description

1

Introduction

This chapter provides product overviews of the Quantum modules for Ethernet communication. It covers the 140 NOE 771 •• modules and the built-in Ethernet port of the 140 CPU 6 •• modules.

What's in this Chapter?

This chapter contains the following topics:

NOE 771 •• Module Overview

Overview

The following information provides an overview of the Quantum 140 NOE 771 •• 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 PLC to communicate with devices over an Ethernet network. The electronics for the NOE 771 •• modules are contained in a standard Quantum single width case 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.

Key Features

The key features of the 140 NOE 771 (-00, -01, -10, -11, -21) models are listed below.

	-00	**-01**	**-10**	**-11**	**-21**
HTTP Server	X	X	X	X	X
FTP Server	X	X	X	X	X
Flash File System	X	X	X	X	X
BOOTP Client	X	X	X	X	X
BOOTP Server	X	X	X	X	X
SNMP V2 Agent	X	X	X	X	X
MODBUS Messaging	X	X	X	X	X
I/O Scanner	X	X		X	X
Hot Standby	X		X		
Global Data - Publish / Subscribe		X		X	X
Bandwidth Monitoring		X		X	X
Faulty Device Replacement (DHCP Server)		X		X	X
Enhanced Web Diagnostics		X		X	X
Schneider Private MIB		X		X	X
FactoryCast Application			X	X	X
User Programmable Web Pages			X	X	X

Front Panel Components

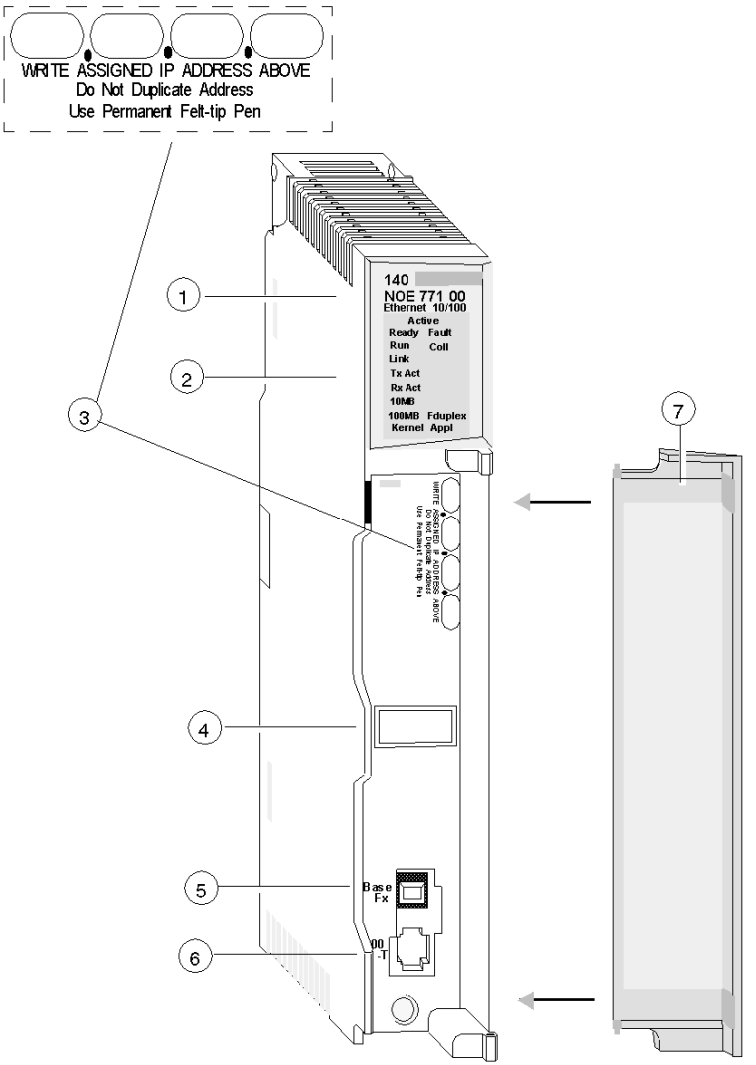
The front panel of the NOE 771 •• modules contain identification markings, color codes, and LED displays. A writable area for an IP address, a global address label, and 2 Ethernet cable connectors is located behind the removable front panel door.

The following table provides a description of the front panel components, which are shown in the front view figure.

Component	Description
LED indicator Panel	Indicates the operating status of the module, and the fiber optic or Ethernet communications network it is connected to. (See LED Indicators in this chapter.)
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 socket for connection to a 100 megabit fiber optic Ethernet cable.
10/100BASE-T Connector	Provides an RJ-45 socket for connection to a shielded, twisted pair Ethernet cable.

Front View

The following figure shows the front of the NOE 771 00 Ethernet module.



1 model number, module description, color code
2 LED display
3 IP address writable area
4 global address label
5 100 Base Fx MT-RJ cable connector
6 10/100 Base-T RJ-45 cable connector
7 removable door

LED Indicators

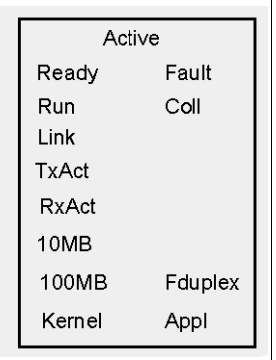
Overview

The LED indicator panel provides continuous operating information about the NOE 771 •• modules and their connection to the network.

LED Indicator Panel

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.
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.
Kernel	Amber	On when in Kernel Mode. Flashing while in download 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 offline.
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

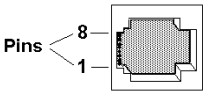
Overview

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

10/100 BASE-T Twisted-Pair Connector

The NOE 771 •• modules' 10/100 BASE-T connector is a standard RJ-45 twisted pair socket.

The following figure shows the 10/100 BASE-T connector.



Schneider Electric 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 •• modules' 100 BASE-FX connector is an MT-RJ socket or a mating fiber optic cable connector. (See the figure in Front View *(see page 16)*.)

For the NOE 771 ••, 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 •• is a 1-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)

Overview

Refer to the I/O Scanner *(see page 120)* section in this manual.

Modbus Messaging

Introduction - Client

All NOE 771 •• 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 use IEC communication blocks.

Introduction - Server

All NOE 771 •• 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 •• supports up to 128 simultaneous Modbus/TCP server connections. The NOE 771•• allows only one programming panel to be logged in at a time to guarantee consistency of changes to the controller configuration.

The NOE supports the following Modbus/TCP commands.

- 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 •• Modbus/TCP server.

Parameter	Value
Typical Response Time (ms)	0.6
Number of Modbus connections (client and server)	64 (-01, -11, -21) 16 (Client -00) 32 (Server -10)
Number of simultaneous login channels	1

NOTE: NOE 771 •• Modbus/TCP performance measurements are made with the Quantum 140 CPU 534 14 PLC.

FTP and HTTP Server

FTP Server

The NOE 771 ••'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:
- updates the NOE's firmware by downloading a new Exec
- provides error log visibility by uploading error log files
- uploads/downloads BootP server and SNMP configuration files.

The default user name is USER, and the default password is USERUSER. Both the user name and password are case sensitive. The instructions for changing the password and adding or deleting user names to the FTP server *(see page 43)* are discussed later in this guide.

There should be only one FTP client per module.

HTTP Server

The NOE 771 ••'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 later of either an Internet Explorer or Netscape browser.

The NOE 771 ••'s HTTP server allows you to view the following information:
- module's Ethernet statistics
- controller and I/O information
- BootP/DHCP/FDR (Faulty Device Replacement) 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 using the configuration page *(see page 47)* on the NOE 771 0•'s Web embedded pages.

For the NOE 771 1• modules, they can be changed using the FactoryCast configurator.

The NOE 771 •• supports a maximum of 32 HTTP simultaneous connections.

NOTE: Browsers may open multiple connections, 32 HTTP connections do not indicate 32 simultaneous users.

NOTE: The NOE 771 0• module does not support user downloaded Web pages. You will need to purchase the 140 NOE 771 1• or -21 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. A BootP server receives the request, and extracts the required IP address information from its database. The server then places 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 •0 module comes supplied with a BootP server. This feature allows you to provide IP addresses to all the I/O devices, which the NOE 771 •0 is servicing. Providing a BootP server that is built into your NOE 771 •0 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 •0's BootP server cannot be used to provide its own IP address.

You can configure your NOE 771 •0'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 -•1 models.

Dynamic Host Configuration Protocol (DHCP) is a superset of the BootP protocol. Your 140 NOE 771 •1 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 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 page 171)*.

If you are migrating a BootP configuration from a 140 NOE 771 •0 module to the new 140 NOE 771 •1 module, see the Address Server Configuration / Faulty Device Replacement topic *(see page 171)* for details on automatic upgrade of your configuration for the new DHCP server.

NOTE: Before placing the NOE on a corporate network, Schneider Electric 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, 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 arranged 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 the user 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 want to 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 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 3 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, 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 •1 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 •0 modules.

System	Minimum Version Number
Exec Version	92.15
Kernel Version	92.15
Web Pages Version	2.2.2

Customer Support

Customer support is available to registered Schneider Electric users.

Please find the nearest Schneider Electric sales office by visiting **http://www.schneider-electric.com**. In the **Select a country** list, click the country closest to you for customer support.

Installing the Module

2

Overview

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

What's in this Chapter?

This chapter contains the following topics:

Before You Begin

Initial Checks

⚠ CAUTION

DUPLICATE ADDRESS HAZARD

Do not connect the module to your network until you have ensured that its IP address will be unique on the network. Two devices with the same IP address can cause unpredictable operation of your network.

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

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Design your application so that unmonitored modules support communication only to non-critical parts of the application.

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

Before you install your module, you need to complete the following checks.

- Determine how the NOE 771 •• module will be assigned its Ethernet address parameters (the default method is BootP).
- Verify that your Ethernet network is properly constructed.

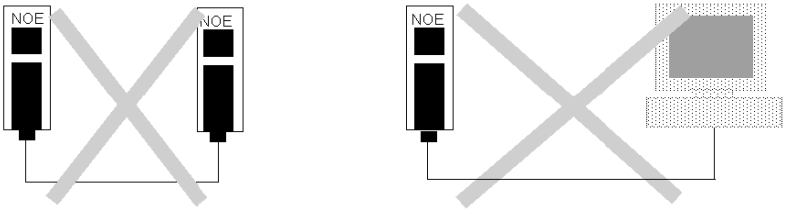
Determining the Appropriate Ethernet Address Parameters

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 must configure the module from your programming panel *(see page 51)*.

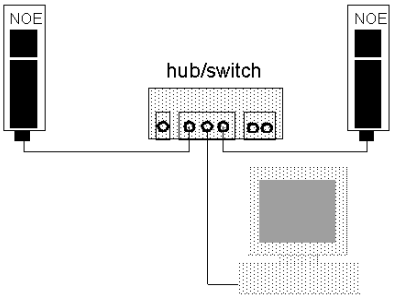
Verifying the Network Topology

Do not use a standard cable to connect an Ethernet Web embedded server module directly to another device. You must use a cross link cable. 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

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

Twisted Pair Length

The following table shows that the maximum length of cable between devices depends on the type of device.

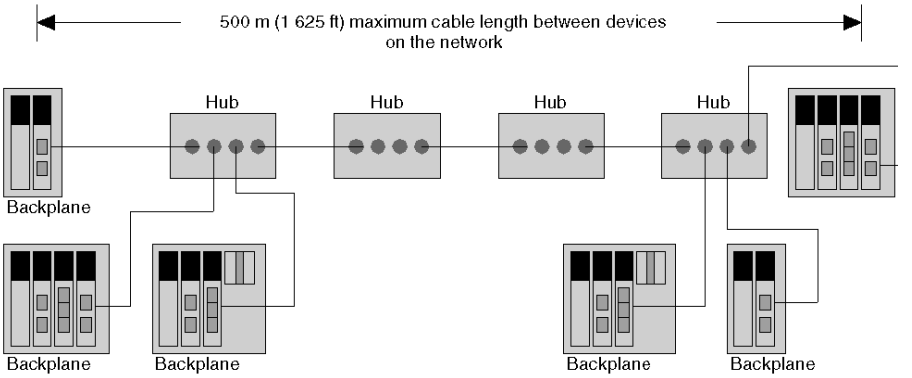
Type of Device	Maximum Cable Length from Device to Hub	Maximum Hubs Between Any 2 Nodes	Maximum Cable Length Between Most Distant Nodes on Network
Hub	100 m	4	500 m
Switch	100 m	Unlimited	Unlimited

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

Cabling with Conventional 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



100 BASE-T Cable Distances

The 100 BASE-T cabling allows for 2 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	Maximum Cable Length in Twisted Pair TX-T2-T4
DTE-DTE (no repeater)	100 m (325 ft)
1 Class I repeater	200 m (650 ft)
1 Class II repeater	200 m (650 ft)
2 Class II repeaters	205 m (665 ft)

100 BASE-FX Cable Distances

The 100 BASE-FX cabling allows for 2 hubs with a link maximum distance of 412 m (1 339 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	Maximum Cable Length Twisted Pair TX and Fiber FX	Maximum Cable Length Fiber FX
DTE-DTE (no repeater)	n.a.	412 m (1339 ft)
1 Class I repeater	260.8 m (*1*)	272 m (884 ft)
1 Class II repeater	308.8 m (*1*)	320 m (1040 ft)
2 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. To restrict access to your Ethernet controller and I/O network, you may want to consider a firewall.

Types of Firewalls

There are 2 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 •• module. Ensure that an open slot is available in which to mount the module.

NOTE:

- The NOE 771 •• module can be installed only in a local backplane.
- Ensure that installing the NOE does not exceed the Quantum backplane requirements.

Backplane Slot Placement

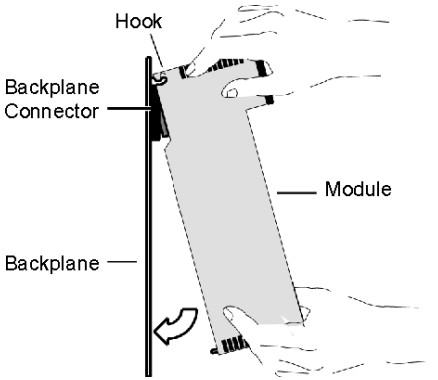
The modules may be placed in any slot on the backplane. They do not have to be placed next to each other.

Tools Required

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

Mounting the Module in the Backplane

Follow the steps below to mount the NOE 771 •• module on to a Quantum backplane.

Step	Action
1	Holding the module at an angle, mount it on the 2 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 the connector engages the backplane connector.
3	Use a Phillips-head screw driver to tighten the screw at the bottom of the module from 2 through 4 in-lbs or from .22 through .45 Newton meters of torque.

Connecting the Cable

Accessories

NOTE: The 140 NOE 771 •• 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*.

The following are switches, which Schneider Electric sells.

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

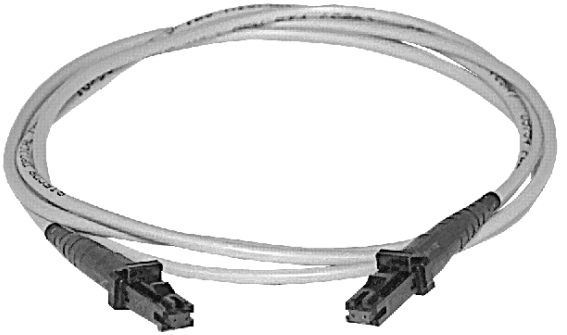
The following are Schneider Electric cables that support multicast filtering.

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

Fiber Optic

Remove the protective cap 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. Note the plug only fits to the socket in 1 way. It should snap into place.

The following figure shows MT-RJ fiber optic cable.



Assigning Ethernet Address Parameters

Overview

⚠ CAUTION

DUPLICATE ADDRESS HAZARD

Be sure that your NOE 771 •0 module receives a unique IP address. Two or more devices with the same IP address can cause unpredictable network operation.

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

The following information describes how to assign IP address parameters.

As shipped from the factory, the NOE 771 •• module does not contain an IP address. You must program the unit with an Ethernet configuration extension to give it 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 assign IP address parameters using the BootP Lite software utility.

NOTE: You can use the Web pages to configure the IP address for the NOE 771 01, -11 and -21 modules.

Using a BootP Server

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.

See *Using BootP Lite to Assign Address Parameters, page 50*.

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

On startup, an unconfigured NOE 771 •• 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 2 minutes, the module uses the default IP address derived from its MAC address.

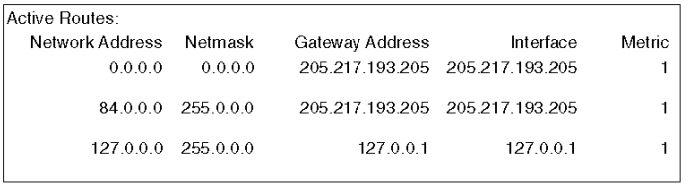
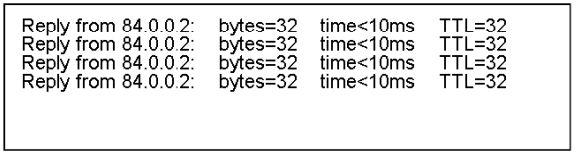
NOTE: The MAC address is assigned at the factory and is recorded on a label on the front panel, above the cable connector. This is a unique 48-bit global assigned address. It is set in PROM. The Ethernet address is recorded on the label in hexadecimal, in the form 00.00.54.xx.xx.xx.

Connecting to the Default IP Address

To connect to the default IP address with your PC, set up an active route from your PC. To do this with either Windows 95/98/ME/NT/2000 or Windows XP, use the following procedure. You can use the routes for connecting Ethernet components with other address ranges.

Step	Action
1	Be sure the NOE module is running.
2	Obtain the default IP address of the NOE derived from its MAC address (e.g. 84.0.0.2).
3	Open an MS-DOS Window.

4	Add an active route for the local NOE by typing:
	`C:\>ROUTE ADD <target> MASK <mask> <gateway>`
	e.g.
	`C:\>ROUTE ADD 84.0.0.0 MASK 255.0.0.0 205.217.193.205`
	Use the default IP address of the NOE module as target address. Use class A subnet mask for connecting to every 84.0.0.0 address. The gateway address is the IP of your PC.
	Result: MS Windows will now talk to any address that starts with an 84, which:
	● is directly connected to a hub or switch accessible to your machine
	- or -
	● the specified route/gateway can see

5	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. 
6	Verify that you have made a connection by typing `C:\>ping 84.0.0.2` The following figure shows that the connection is verified. 

Active Routes:

Network Address	Netmask	Gateway Address	Interface	Metric
0.0.0.0	0.0.0.0	205.217.193.205	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

```
Reply from 84.0.0.2:  bytes=32  time<10ms  TTL=32
Reply from 84.0.0.2:  bytes=32  time<10ms  TTL=32
Reply from 84.0.0.2:  bytes=32  time<10ms  TTL=32
Reply from 84.0.0.2:  bytes=32  time<10ms  TTL=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 Configuring the Ethernet Address Parameters *(see page 62)*.

If BootP Server Responds

If the server responds with address parameters, the NOE 771 •• 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 2 minutes.

If BootP Server Does Not Respond

If no BootP response is received, the NOE 771 •• module will use the default IP address.

During this time, the Run indicator will display a pattern of 5 flashes for a BootP attempt and 6 flashes for using the default IP.

NOE 771 •• Duplicate IP Address Test

In all cases, when the NOE 771 •• module receives an IP address, it will test for duplicate addresses by sending broadcast ARP requests 3 times at 5-second intervals.

If a duplicate IP address is found on the network, the NOE 771 •• will stay offline to avoid a network disruption. It will display a pattern of 4 flashes to indicate a duplicate IP address detection.

Automatic ARP

If there are no replies to its requests, the NOE 771 •• will send automatic ARP 3 times at 2-second intervals to announce its presence on the network.

Establishing the FTP Password

Establishing the FTP Password

The FTP password is established using the embedded Web server. This topic contains information about initially accessing the Web server. Initially, the system administrator must change the FTP password, which restricts access for the system administrator only.

See *Embedded Web Pages, page 143* for detailed information.

Accessing the Web Server

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

The Web pages can be viewed only across the World Wide Web using version 4.0 or later 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 1• modules, see the *FactoryCast User's Guide* (31001229).

Accessing the Module's Home Page

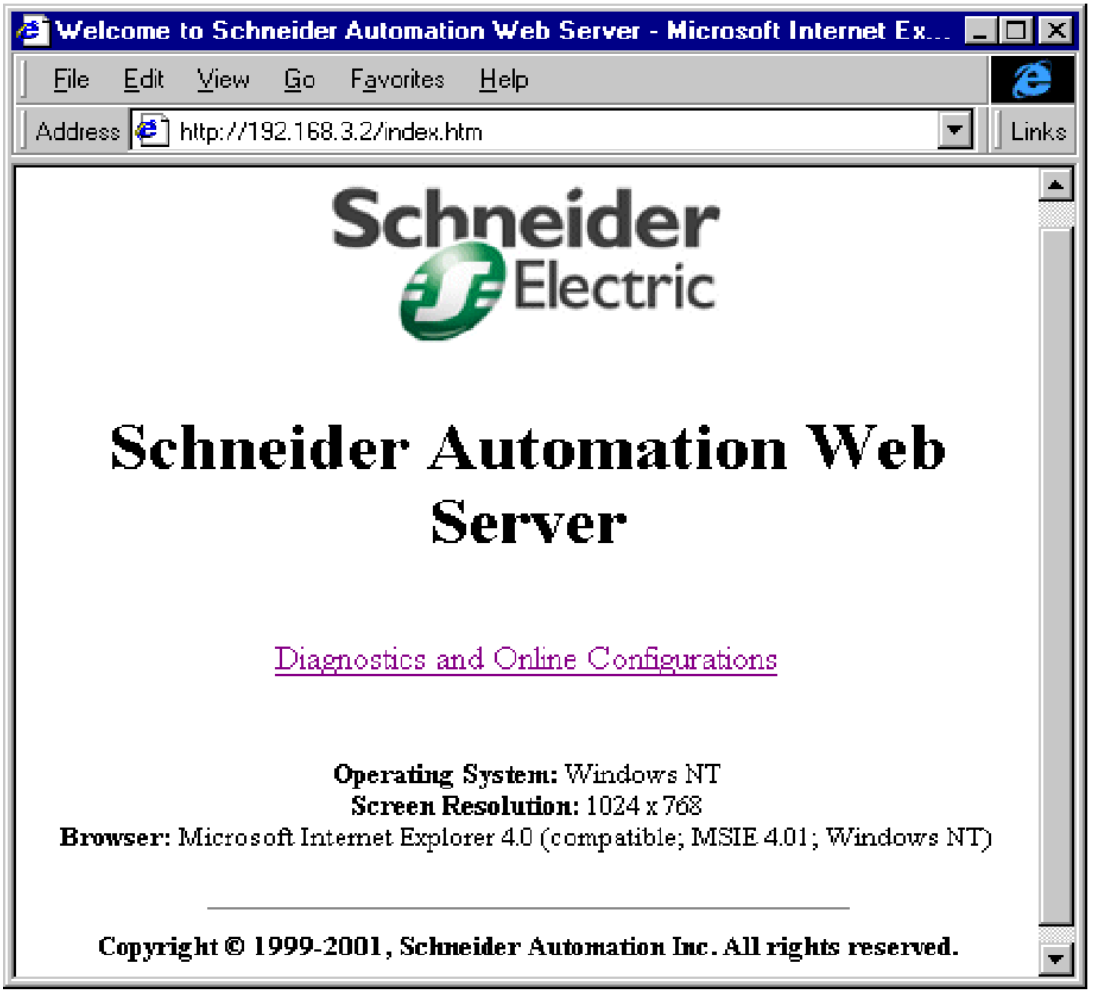
Before you can access the module's home page, you must enter the full IP address or URL in the **Address** or **Location** field in the browser window.

Example: http://hostname (where hostname is the full IP address or DNS host name)

Result: The Schneider Automation Web server home page displays.

Schneider Web Utility Home Page

The following figure shows the Schneider Automation Web server home page.

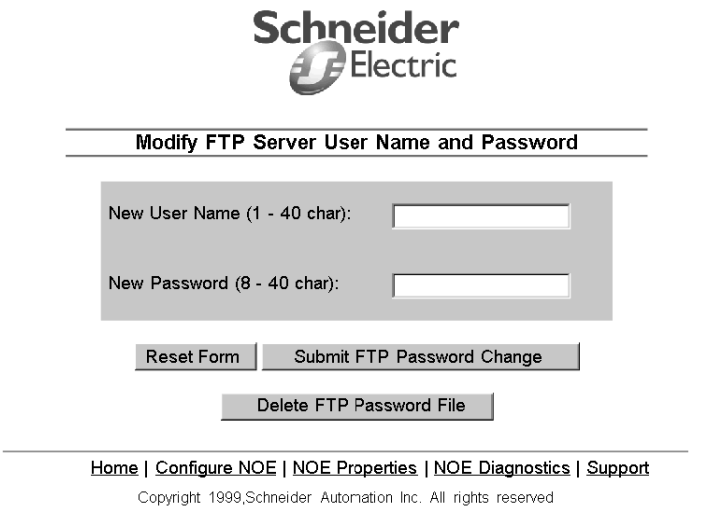


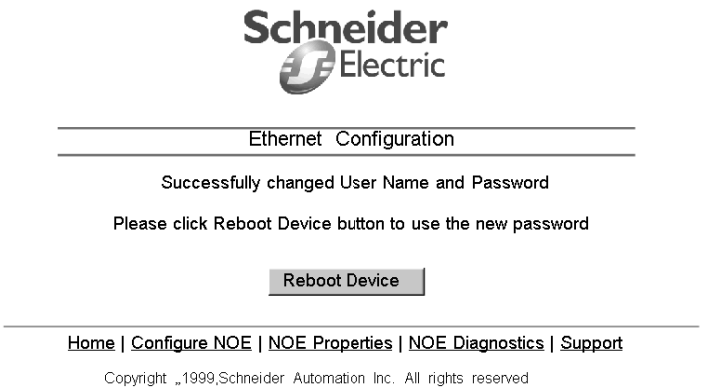
From this page, you can access pages to perform the following:

- change the FTP password
- change the HTTP password
- monitor diagnostic and configuration information
  (see *Embedded Web Pages, page 143*)

Modifying the FTP Server Password

Follow the steps below to link to the correct Web page to modify the FTP password.

Step	Action
1	Enter the URL. **Example**: http://hostname/secure/embedded/ftp_passwd_config.htm
2	Enter a user name and password in the **Enter Network Password** dialog box. Click **OK**. **Note**: The default user name is **USER**, and the default password is **USERUSER**. You should change both when you install the module. 
3	Enter a new user name and password in the respective fields in the **Modify FTP Server User Name and Password** dialog box. Click **Submit FTP Password Change**. 

Step	Action
4	The following dialog box appears.  If you click **Reboot Device**, the user name and password reset for the NOE 771 •• board. **Note**: The reboot requires approximately 40 seconds. (With large applications, the reboot may require up to 60 seconds). During the reboot, all services provided by the NOE 771 •• are not available.

Establishing the HTTP Password

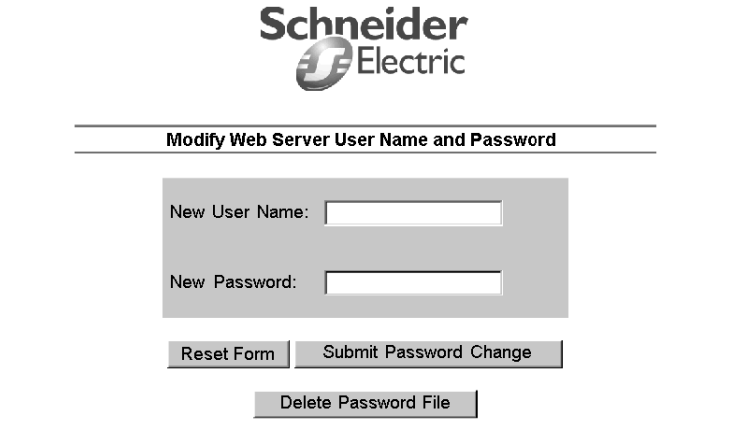
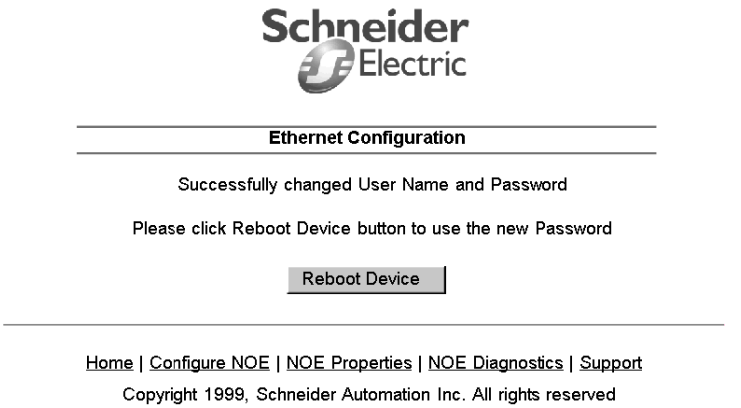
Overview

The following information describes how to set the HTTP password for the NOE 771 0• only.

Modifying the HTTP Password

Follow the steps below to link to the correct Web page to modify the HTTP password.

Step	Action
1	Enter the URL. **Example**: http://hostname/secure/embedded/http_passwd_config.htm
2	Enter a user name and password in the **Enter Network Password** dialog box. Click **OK**. **Note**: The default user name is **USER**, and the default password is **USER**. You should change both when you install the module. 

Step	Action
3	Enter a new user name and password in the respective fields in the **Modify Web Server User Name and Password** dialog box. Click **Submit Password Change**. 
4	The following dialog box appears.  If you click **Reboot Device**, the user name and password reset for the NOE 771 0• board. **Note**: The reboot requires approximately 40 seconds. (With large applications, the reboot may require up to 60 seconds). During the reboot, all services provided by the NOE 771 0• are not available.

Establishing the SNMP Community Strings

Overview

SNMP Community Strings are used to restrict access to the SNMP agent. Set the strings to non-trivial names during module installation.

Establishing the SNMP Community Strings

Follow the steps below to establish the SNMP Community Strings.

Step	Action
1	Enter the following 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 in the **Set**, **Get**, and **Trap** fields. 
3	Click **Update SNMP**.
4	Reboot the module using hot swap.

Using BootP Lite to Assign Address Parameters

BootP Lite Utility

⚠ CAUTION

UNINTENTIONAL OPERATION — INCORRECT MAC ADDRESS

- Verify the MAC address of the target device before invoking BootP Lite Server Software.
- You must enter the correct parameters of the target controller, or 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.

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

Instead of a BootP server, Schneider Electric's BootP Lite utility can be used to provide the IP address, subnet mask, and default gateway to the NOE 771 •• module.

NOTE: Refer to the BootP Lite server software and user documentation, which are available for download at www.schneider-electric.com.

Configuring the Module with Concept

3

Introduction

This chapter describes how to configure the NOE 771 module from your programming panel 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 the module's default IP address.

What's in this Chapter?

This chapter contains the following topics:

Selecting Your PLC

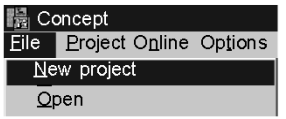
Initial Setup

After you have installed the NOE 771•• module *(see page 37)* in a Quantum backplane, you can configure it using Concept. To begin configuring the NOE 771 ••, first select your CPU (PLC).

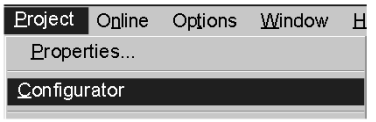
NOTE: For detailed information about how to use Concept, refer to the set of manuals shipped with that software.

Selecting a CPU

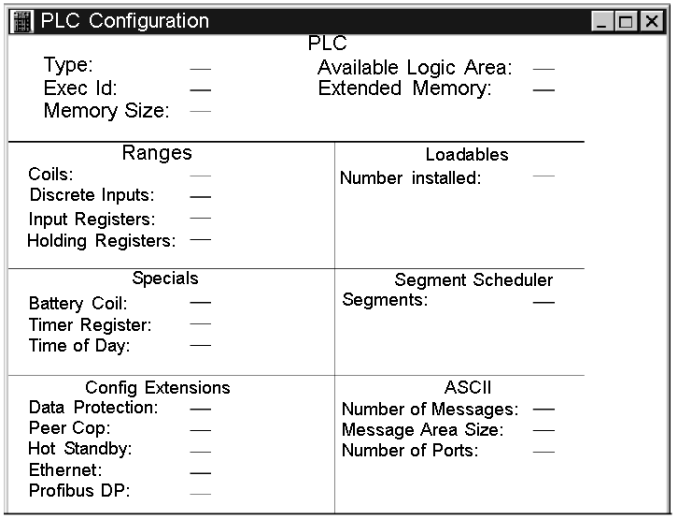
Perform the following steps to select a CPU.

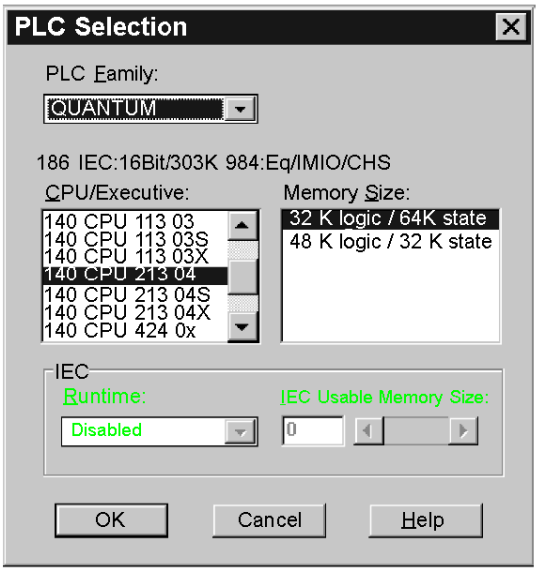
Step	Action
1	Open Concept on your programming panel (PC).
2	Click **File** → **New project**.  **Result**: A new project opens, and the file name (untitled) appears over the menu bar. 

Step	Action
3	Click **Project** → **Configurator**.

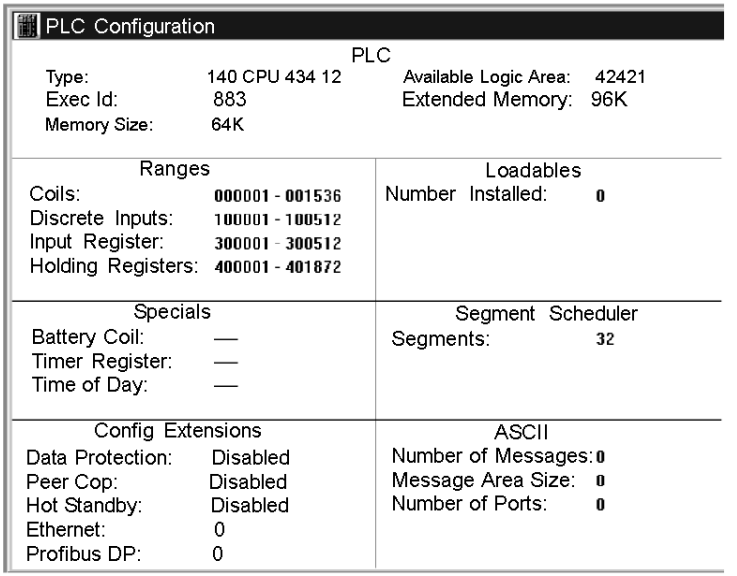


Result: The **PLC Configuration** screen opens.



Step	Action
4	Double click the **Type** field in the PLC section of the **PLC Configuration** screen. **Result**: The **PLC Selection** dialog box opens. The default selection is Quantum. 
5	In the **CPU/Executive** list, select the CPU that is installed in your Quantum backplane. **Note**: Depending on the CPU you select, you may need to select the correct memory size applicable to it in the **Memory Size** list.

Step	Action
6	Click **OK**. **Result**: Your PLC type and default configuration parameters display in the PLC Configuration screen.



Configuring the Number of Ethernet Modules

Next, you must configure the number of Ethernet modules that your system will contain, as shown in Cabling Schemes *(see page 34)*.

Setting the Number of NOEs

Overview

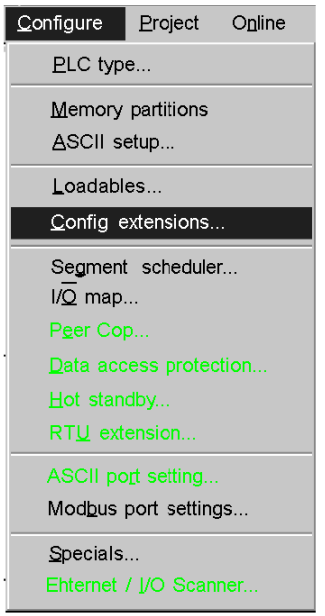
You may configure from 2 to 6 Ethernet modules in a single controller, depending on the model. A 140 CPU 113 or 213 will accept a total of 2 network option modules, including NOE, NOM, NOP, and CRP 811. A 140 CPU 424, 434, 534, 434A, or 534A will accept 6. See *I/O Scanner Concepts, page 120* regarding the mix of I/O scanners and NOE modules per CPU.

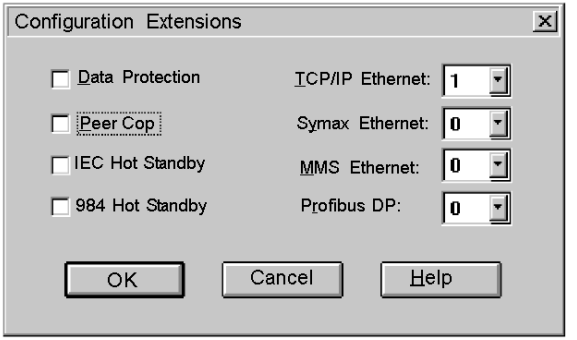
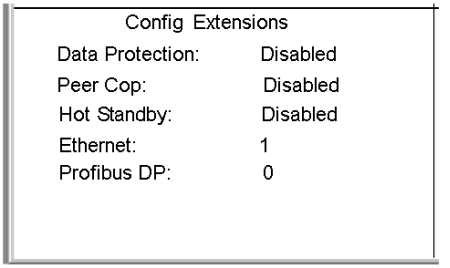
Memory Requirements

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

Setting the Number of NOEs

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

Step	Action
1	Click **Configure** → **Config extensions**, or double-click anywhere in the Config Extensions section of the screen.  **Result**: The Configuration Extensions dialog box opens.

Step	Action
2	In the **TCP/IP Ethernet** list, select the number of NOE modules you want to configure. 
3	Click **OK**. **Result**: The Ethernet status changes from 0 to the number selected in Step 2. 

Creating an I/O Map for the NOEs

Next, you must create an I/O map for the NOEs in your configuration, as shown in Accessing and Editing the I/O Map *(see page 58)*.

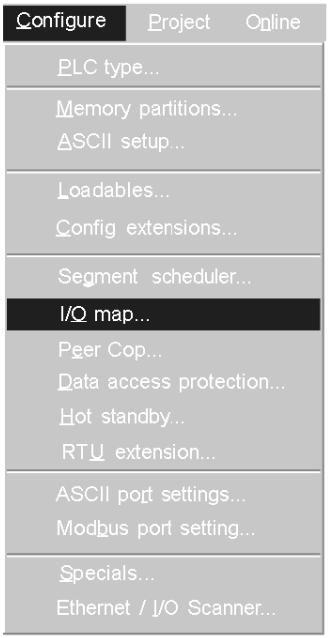
Accessing and Editing the I/O Map

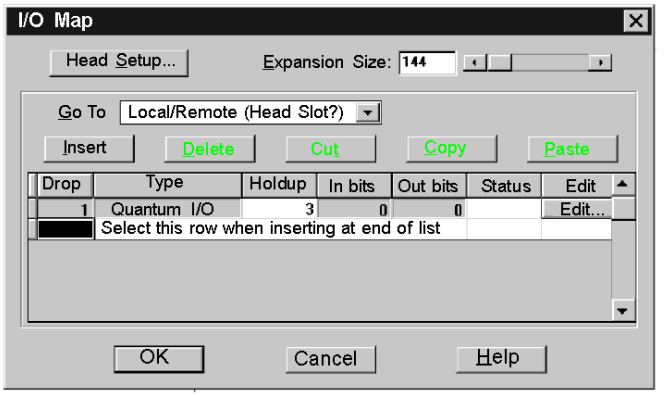
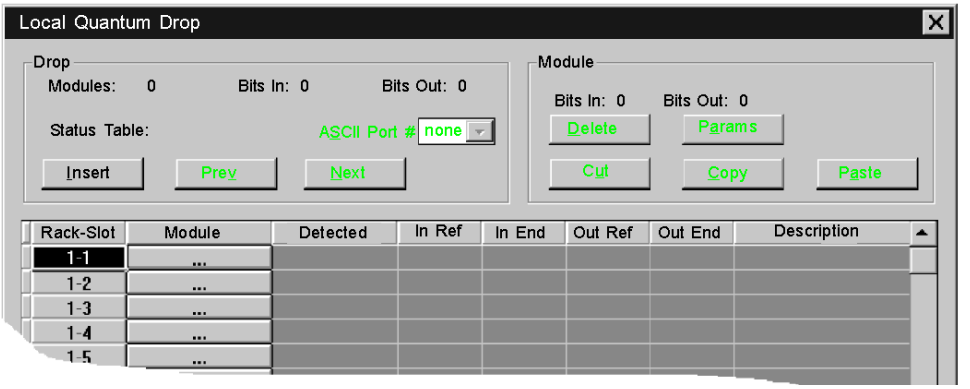
Overview

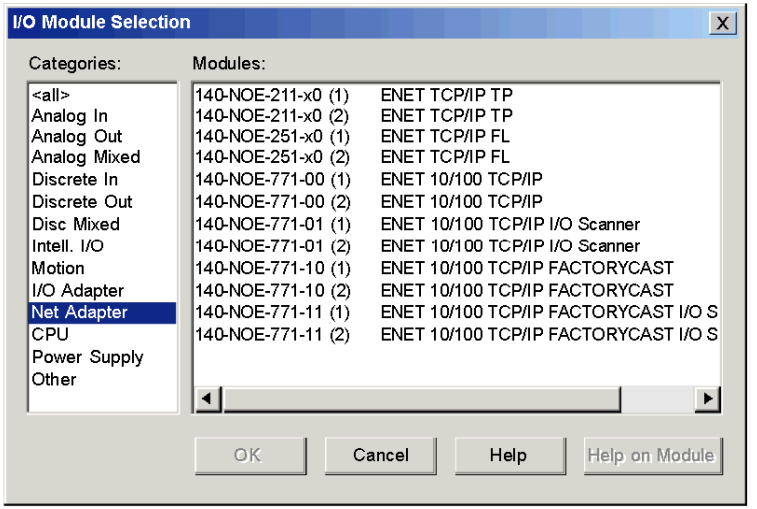
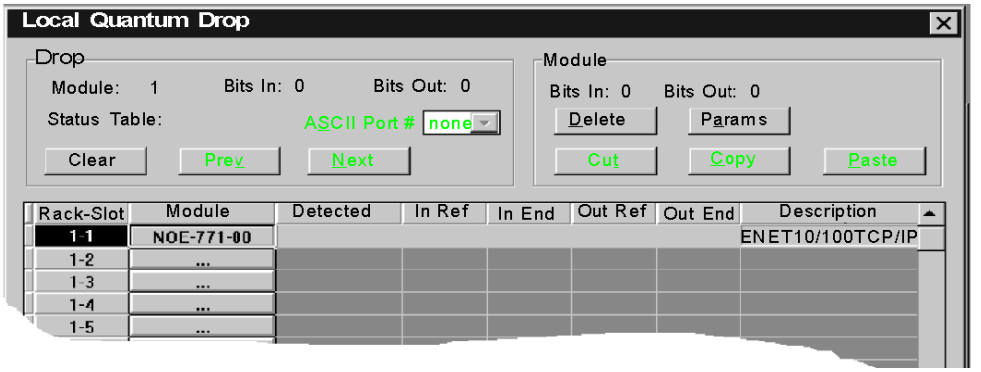
The following procedure describes how to create an I/O map for the NOEs in your system. The procedure determines the number of NOEs in the system and the slot numbers in which they reside.

Accessing and Editing an I/O Map

Follow the steps below to access and edit an I/O map from the PLC Configuration screen.

Step	Action
1	Click **Configure** → **I/O map**.  **Result**: The I/O Map dialog box opens (see below).

Step	Action
2	Click **Edit**.  **Result**: The Local Quantum Drop I/O dialog box opens. 

Step	Action
3	Click the **ellipsis (...) ...** button under the **Module** column. **Result**: The I/O Module Selection dialog box opens. 
4	Click **Net Adapter** in the Categories field, and click **140-NOE-771-00** in the Modules field. Click **OK**. **Result**: The Local Quantum Drop dialog box updates, and the NOE-771-00 is now listed under the Module column and described in the Description column. 
5	Repeat Steps 3 and 4 if you are adding other modules to the I/O map.
6	Click **OK** to return to the PLC Configuration screen.

Configuring the Ethernet Address Parameters

Next, you must configure the Ethernet address parameters from the Ethernet I/O Scanner screen as shown in Configuring the Ethernet Address Parameters *(see page 62)*.

Configuring the Ethernet Address Parameters

Overview

The following information describes how to configure Ethernet address parameters for the NOE 771 •• with Concept.

Introduction

⚠ CAUTION

UNINTENTIONAL OPERATION — DUPLICATE IP ADDRESS

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.

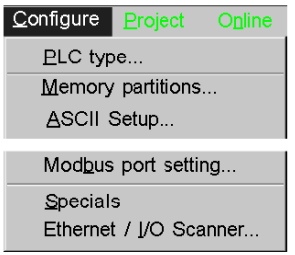
Failure to follow these instructions can result in injury or equipment damage.

The NOE 771 •• 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.

NOTE: The state must be offline to configure the NOE 771 •• module with Concept.

Configuring Ethernet Address Parameters

Follow the steps below to configure the Ethernet address parameters.

Step	Action
1	Click **Configure** → **Ethernet / I/O Scanner**.  **Result**: The Ethernet/ I/O Scanner dialog box opens.
2	Click the **Specify IP Address** button.
3	Type the new IP, Subnet Mask, and Gateway addresses in the respective fields.
4	Select the correct Internet frame type from the **Frame Type** list.
5	If the module's BootP server will assign Ethernet address parameters, click the **Use BootP Server** button. **Note**: If you click this button, the address parameter fields will not be available 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 use the IP Address derived from its MAC address.

NOTE: The MAC address is assigned at the factory, and is recorded on a front panel label, above the cable connector. This is a unique 48-bit global assigned address, which 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

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 MBP_MSTR instruction or an IEC Logic function. Operational statistics and error codes for reading and writing the controller information are included.

What's in this Chapter?

This chapter contains the following sections:

4.1 Using 984 Ladder Logic Communication Blocks

Overview

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

All NOE 771 •0 Quantum Ethernet TCP/IP modules allow the user to transfer data to and from nodes on a TCP/IP network through the use of an MSTR. 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. A specific code designates each operation.

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
Reset Option Module	10	supported
Read CTE (config extension)	11	supported
Write CTE (config extension)	12	supported

Number 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 for logic flow activation. In this case, when 1 active MSTR block releases the resources it has been using and becomes deactivated, the next MSTR operation that the ladder logic encounters can be activated.

MSTR Block for TCP/IP in Concept

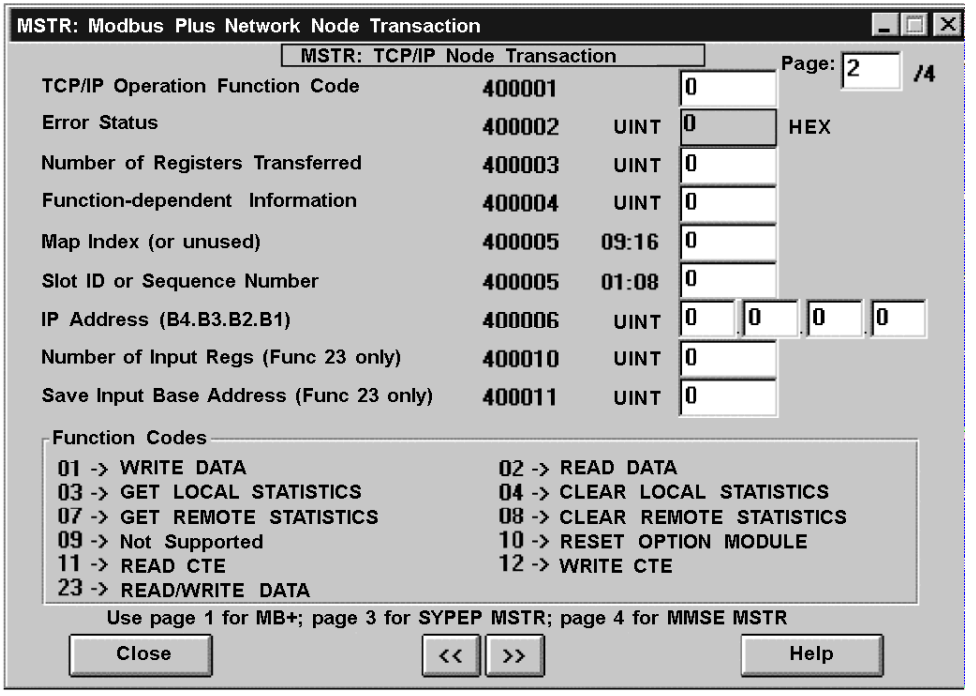
Overview

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

Installing the MSTR Block in TCP/IP

This is the MSTR Block as used for TCP/IP in Concept ladder logic. After the MSTR Block is inserted in the network, follow the steps below.

Step	Action
1	Position your cursor on the MSTR block.
2	Configure the control block and data area by double clicking.
3	Press **Ctrl+D** or click **Edit** → **DX Zoom**.
4	Go to page 2 for TCP/IP specific MSTR block and complete the required information.



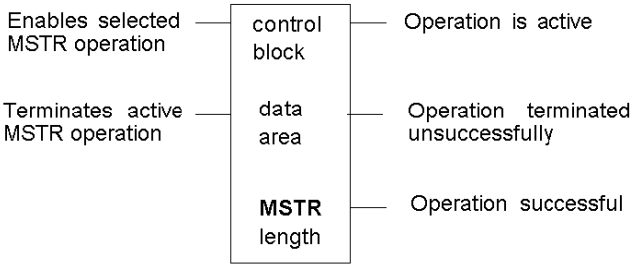
MSTR Ladder Logic Representation

Overview

The following information describes the ladder logic representation for MSTR.

Ladder Logic Diagram

The MSTR Block is represented in ladder logic diagrams, as shown below.



Inputs

The MSTR instruction has the following 2 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 3 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.

When all outputs are 0, 4 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 page 75)*, 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 1 of 10 MSTR operations legal for TCP/IP (1 to 4 and 7 to 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 (MSB) 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 (LSB) 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 get 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 page 75)*, 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 from 1 to 100.

MSTR Function Error Codes

Overview

The following information describes the error codes for MSTR operations.

Display in Control Block

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* when the MSTR routine runs over TCP/IP Ethernet.

Error Code (hex.)	Description
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 failure code.
30ss*	Modbus slave exception response.
4001	Inconsistent MODBUS slave response.
F001	Optional module not responding
F002	Module not fully initialized
* ss = subfield	

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

ss Hex. Value	Description
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	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	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

Hex Error Code	Meaning
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
5048	hardware length is invalid
5049	route specified cannot be found
504A	collision in select call: these conditions have already been selected by another task.
504B	task ID is invalid
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	nonoperational TCP connection
5057	incoherent configuration
Note: 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 topic describes these operations for MSTR:

- **read operation:** An MSTR read operation transfers data from a specified slave source device to a master destination device on the network. (It is operation type 1 in the displayed register of the top node.)
- **write operation:** An MSTR write operation transfers data from a master source device to a specified slave destination device on the network. (It is operation type 2 in the displayed register of the top node.)

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,

Register	Function	Content	
Displayed	Operation Type	1 = write, 2 = read	
First implied	Error status	Displays a hex value indicating an MSTR error.	
		Exception response, where response size is incorrect.	Exception code +3000
		Exception response where response size is incorrect.	4001
		Read Write	
Second implied	Length	Write = number of registers to be sent to slave. Read = number of registers to be read from slave.	
Third implied	Slave device data area	Specifies the starting %MW word (4x register) in the slave to be read from or written to (1 = 4001, 49 =40049).	
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.	

Read/Write Data

Introduction

In a single transaction, the MSTR read and write operations can transfer data from a master source device to a specified slave destination device, then transfer data from this specified slave source to the master. It uses a master transaction path and may require several cycles to complete. To program an MBP_MSTR block to perform a combined read/write operation, use function code 23.

The combined read/write operation can be used only with these two Quantum models:

- NOE 771 01 (version 3.0 or later)
- NOE 771 11 (version 3.0 or later)

Control Block Usage

Register	Content
CONTROL[1]	23 = read/write data.
CONTROL[2]	Indicates the error status.
CONTROL[3]	Number of registers to be sent to the slave.
CONTROL[4]	Specifies the %MW starting register in the slave to which the data will be written, e.g. 1 = %MW1, 49 = %MW49.
CONTROL[5]	Routing register: Most significant byte: network adapter module slot. Least significant byte: MBP on Ethernet Transporter (MET) mapping index.
CONTROL[6] ... CONTROL[9]	Each address contains 1 byte of the 32-bit IP address, where the MSB is in CONTROL[6] and the LSB is in CONTROL[9].
CONTROL[10]	Number of registers to be read from slave.
CONTROL[11]	Specifies the %MW starting register in the slave from which the data is read, e.g. 1 = %MW1, 49 = %MW49.

NOTE:

When configuring the MBP_MSTR block for a read/write data operation, note that

- The DATABUF output parameter is used to store, in the following sequence, both:
  1 the data to be written
  2 the data to be read
- The size of the DATABUF output parameter must equal the combined size of the data to be written and the data to be read; if the size is smaller, data will be overwritten and may be lost.
- Both the CONTROL and DATABUF parameters must be stored at located addresses, for example %MW addresses.

Get Local Statistics MSTR Operation

Overview

This topic describes the Get Local Statistics operation (operation type 3 in the display register of the top node). This operation obtains information related to the local node where the MSTR has been programmed. (Available Ethernet statistics are listed in the TCP/IP Ethernet Statistics table.)

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 (operation type 4 in the displayed register of the top node). This operation 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

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. (For more information, see TCP/IP Ethernet Statistics.)

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 %MW words (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

Introduction

The following information describes the Clear Remote Statistics operation (operation type 8 in the displayed register of the top node). This operation 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.

Reset Option Module MSTR Operation

Overview

The following information describes the Reset Option Module operation (operation type 10 in the displayed register of the top node). This operation 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

Introduction

The following information describes the Read CTE operation (operation type 11 in the displayed register of the top node). This operation 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 [1]	Operation Type	11
First implied [2]	Error status	Displays a hex value indicating an MSTR error, when relevant.
Second implied [3]	Length	Must indicate a length of between 12 and 37.
Third implied [4]	Not applicable	
Fourth implied [5]	Low byte	Quantum backplane slot address of the NOE module.
Fifth [6]... Eighth [9] 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 %MW words (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	Byte 4 (MSB) of the 32-bit IP address	
	Second implied	Byte 3 of the 32-bit IP address	
	Third implied	Byte 2 of the 32-bit IP address	
	Fourth implied	Byte 1 (LSB) of the 32-bit IP address	
Subnetwork mask	Fifth implied	Hi word	
	Sixth implied	Low word	
Gateway	Seventh implied	Byte 4 (MSB) of the 32-bit gateway address	
	Eighth implied	Byte 3 of the 32-bit gateway address	
	Ninth implied	Byte 2 of the 32-bit gateway address	
	Tenth implied	Byte 1 (LSB) of the 32-bit gateway address	
	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

Introduction

The following information describes the Write CTE operation (type 12 in the displayed register of the top node). This 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.

The Write CTE operation 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 [1]	Operation type	12
First implied [2]	Error status	Displays a hex value indicating an MSTR error, if relevant.
Second implied [3]	Length	Must indicate a length of between 12 and 37.
Third implied [4]	Not applicable	
Fourth implied [5]	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 [6] ... Eighth [9] 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 %MW words (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 = NOE771 00 3 = NOE771 10 4 = M1 5 = 140 NOE 771 01 6 = 140 NOE 771 11	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)

TCP/IP Ethernet Statistics

Introduction

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

Board Responses

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 under this topic)
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 for the:

- 140 NOE 771 ***x***1, versions 2.0, 3.0, 3.1, 3.3, and 3.6 or higher, and
- 140 NOE 771***x***0, versions 3.0, 3.3, and 3.4 or higher

Bit #	Definition
15	0 = Link LED off 1 = Link LED on
14	0 = Appl LED off 1 = Appl LED on
13	0 = twisted pair 1 = fiber
12	0 = 10 Mbit 1 = 100 Mbit
11 ... 8	Reserved
7 ... 4	Module Type (see table, below)
3	Reserved
2	0 = half duplex 1 = full duplex
1	0 = not configured 1 = configured
0	0 = PLC not running 1 = PLC/NOE running

NOTE: Bits are counted from right to left starting from bit 0 (low bit). For example, **PLC running** = 0x0001, **Application LED** = 0x4000, and **LED Connection**= 0x8000.

The following table describes the word bit definitions for board status for the:

- 140 NOE 771 ***x***1, versions 3.5, and
- 140 NOE 771***x***0, versions 1.02 and 2.0, and
- 140 CPU 651 ***x***0

Bit #	Definition
15 ... 12	Module Type
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

NOTE: Bits are counted from right to left starting from bit 0 (low bit). For example, **PLC running** = 0x0100, **Application LED** = 0x0040, and **LED Connection**= 0x0080.

Board Status Word Bit Definition by Module Type

The following table describes the values of the module types:

Value of Bits 7 ... 4 or 15 ... 12 (see tables above for bit range applicable to your module's software version)	Module Type
0	NOE 2x1
1	ENT
2	M1E
3	NOE 771 00
4	ETY
5	CIP
6	(reserved)
7	140 CPU 651 x0
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 170 ENT 110 01 and Momentum 170 ENT 110 00, see the *Momentum Ethernet Communication Adapter 170 ENT 110 01 and 170 ENT 110 00 User Guide* (870 USE 114).

For bit level detail for the 140 NOE 211, see the *TCP/IP Module User Guide* (840 USE 107).

4.2 Using IEC Logic Communication Blocks

Overview

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

The CREAD_REG block reads register data continuously from an addressed node via TCP/IP-Ethernet.

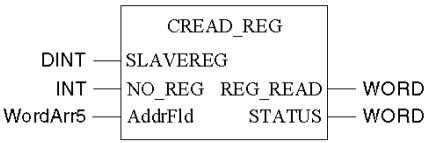
EN and ENO can be projected as additional parameters.

NOTE: About this function block:

- When programming this function, you must be familiar with the routing procedures used by your network.
- For technical reasons, this function block does not allow the use of ST and IL programming languages.

Representation

Block representation:



Parameter Description

Description of parameters:

Parameter	Data Type	Meaning
SLAVEREG	DINT	offset address of the first %MW word (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 %MW word (4x register) for read values
STATUS	WORD	error code

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 sixteen 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 (for example, MBP_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 ••) 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 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.

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

SLAVEREG is the start of the area in the addressed slave from which the source data is read. The source area always resides within the %MW word (4x register) area. SLAVEREG expects the source reference as offset within that area. (In 4x registers, the leading "4" must be omitted. For example, "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

NO_REG is the 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.

REG_READ

The REG_READ 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.

The STATUS parameter can be specified as direct address, located variable, or unlocated variable.

CWRITE_REG

Function Description

The CWRITE_REG block writes data to a register area continuously, transferring data from the PLC via TCP/IP Ethernet to an addressed slave.

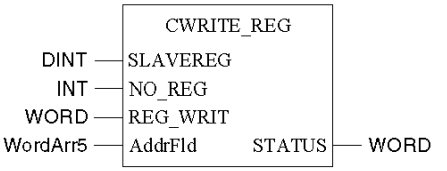
EN and ENO can be configured as additional parameters.

NOTE: About this function block:

- When programming this function, you must be familiar with the routing procedures used by your network.
- 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 %MW word (4x register) in the slave to be written to
NO_REG	INT	number of registers to be written to slave
REG_WRIT	WORD	first %MW word (4x register) of the source data area
AddrFld	WordArr5	data structure for transferring the TCI/IP address
STATUS	WORD	MBP_MSTR error code

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 sixteen write operations may be active at the same time. It makes no difference whether these operations are performed using this function block or others (for example, MBP_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

SLAVEREG is the start of the area in the addressed slave to which the source data are written. The destination area always resides within the %MW word (4x register) area. SLAVEREG expects the destination address as offset within that area. In 4x registers, the leading 4 must be omitted. For example, 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

NO_REG is the 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 STATUS parameter can be specified as direct address, located variable or unlocated variable.

REG_WRIT

The REG_WRIT 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

Upon request, the READ_REG block reads a register area once (rising edge of the REQ input). It reads data from an addressed slave via TCP/IP Ethernet.

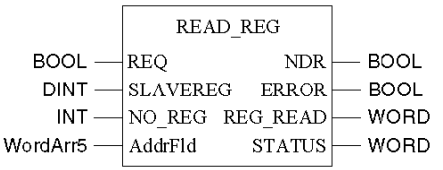
EN and ENO can be projected as additional parameters.

NOTE: About this function block:

- When programming this function, you must be familiar with the routing procedures used by your network.
- For technical reasons, this function block does not allow the use of ST and IL programming languages.

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 %MW word (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
REG_READ	WORD	first %MW word (4x 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 16 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 (for example, MBP_MSTR, CREAD_REG). All function blocks use 1 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 possible only when 1 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 REQ parameter can be specified as direct address, located variable, unlocated variable, or Literal.

SLAVEREG

SLAVEREG is the start of the area in the addressed slave from which the source data is read. The source area always resides within the %MW word (4x register) area. SLAVEREG expects the source reference as offset within that area. In 4x registers, the leading 4 must be omitted. For example, 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 NO_REG 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 NDR 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 ERROR 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 REG_READ parameter must be entered as a direct address or located variable.

STATUS

Error code, see Runtime errors.

The STATUS parameter can be specified as direct address, located variable or unlocated variable.

WRITE_REG

Function Description

Upon request, the WRITE_REG block writes a register area once (rising edge of the REQ input). It transfers data from the PLC via TCP/IP Ethernet to an addressed slave.

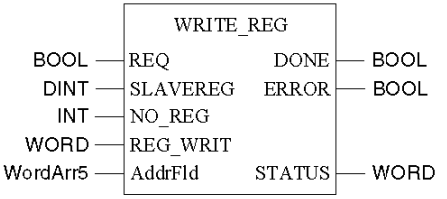
EN and ENO can be configured as additional parameters.

NOTE: About this function block:

- When programming this function, you must be familiar with the routing procedures used by your network.
- 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 %MW word (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 %MW word (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

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 sixteen 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 (for example, MBP_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 possible only when 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 REQ parameter can be specified as direct address, located variable or unlocated variable.

SLAVEREG

SLAVEREG is the start of the area in the addressed slave from which the source data is read. The source area always resides within the %MW word (4x register) area. SLAVEREG expects the source reference as offset within that area. In 4x registers, the leading 4 must be omitted. For example, 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

The REG_WRIT 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 DONE 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

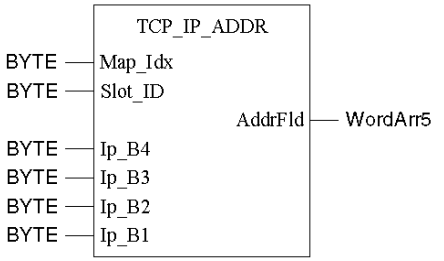
The TCP_IP_ADDR block enables the input of TCP/IP addresses for the READ_REG *(see page 96)*, CREAD_REG *(see page 90)*, WRITE_REG *(see page 99)*, and CWRITE_REG *(see page 93)* blocks. The address is transferred in the form of a data structure.

EN and ENO can be projected as additional parameters.

NOTE: When programming the TCP_IP_ADDR function, you must be familiar with your network's routing procedures.

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

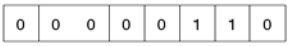
Elementary Description for WordArr5

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 (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

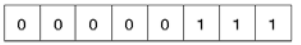
Map_Idx

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



Slot_ID

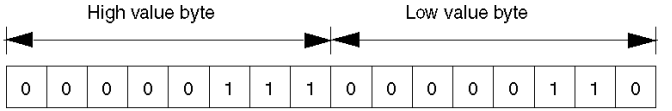
If an NOE in the rack of a Quantum controller is addressed as destination node, the value at the Slot_ID input represents the physical NOE slot. That is, if the NOE is plugged in at Slot 7 of the rack, the value appears as follows:



NOTE: When using an integrated Ethernet CPU module such as the 140 CPU 651 ***x***0, the slot ID must be 254 (FE hex) regardless of the CPU slot.

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. That is, 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 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 16 can be active at the same time. All function blocks use 1 data transaction path and require multiple cycles to complete a job.

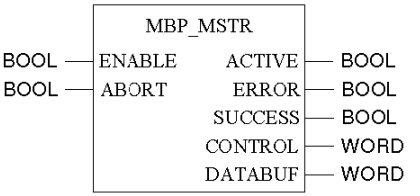
EN and ENO can be configured as additional parameters.

NOTE: About this function block:

- When only 1 read or write job is carried out in every cycle, TCP/IP communications are possible only between a Quantum PLC (NOE 771 ••) and a Momentum PLC (all TCP/IP CPUs and all TCP/IP I/O modules). If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.
- In FBD and LD sections, this function block can only be used on the program level, that is, not in derived function blocks (DFBs).
- When programming this function, you must be familiar with the routing procedures used by your network.
- 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
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 %MW word (4x register) of the MSTR control block
DATABUF	WORD	first %MW word (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.

The following table explains the valid function codes for the MBP_MSTR block:

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	Reset optional module	X
10	Read CTE (Config extension)	X
11	Write CTE (Config extension)	X
12	Send email	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 %MW word (4x register) specified is the first in a group of successive output/marker words in the data field. For operations providing data (for example, write operations), the data field is the data source. For operations receiving data (for example, read operations), 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 %MW words (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

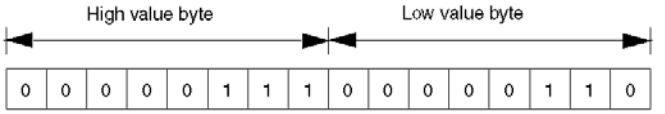
Control Block for TCP/IP Ethernet

The following table shows the control block for TCP/IP Ethernet:

Register	Contents
4x	Indicates an operations that is 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 an NOE in a Quantum controller rack is addressed as a 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. That is, if the NOE is plugged into 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: 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, -21

5

Overview

The material in this section presents the Global Data (Publish / Subscribe) utility available on the following 140 NOE 771 •• modules.

- 140 NOE 771 01
- 140 NOE 771 11
- 140 NOE 771 21

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 arranged in a distribution group for the purpose of application variable exchange and synchronization. Each Global Data device can publish up to 1 network (application) variable and subscribe up to 64 network (application) variables.

The Quantum NOE's embedded Global Data Configuration Web 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 1 network variable of up to 512 4x registers.
- A device can subscribe up to 64 network variables of up to 2 048 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.
- A NOE has only 1 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 1 subscriber is receiving the same data since only 1 transaction is necessary for all subscribers to receive the data.

This advantage offers 2 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 planning your installation before implementation. Work spent on planning saves time and money by reducing errors and unnecessary debugging time. Planning also ensures 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 Schneider calls 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.]	**Length (Registers)**	**Device Number**				**Variable Public. Status**
				1	2	...	3	
	1	*VALVE_STATUS*	*20*	*PUB*	*SUB*		*NONE*	OK
	2	*VALVE_CONTROL*	*10*	*SUB*	*NONE*		*PUB*	OK
	...							
	64	*PUMP_CONTROL*	*50*	*SUB*	*PUB*		*NONE*	OK
Device Publication Status:				OK	OK		OK	
Total Publication Size per Node:				20	50		10	
Total Subscription Size per Node:				60	20		0	

Group IP Address	*239.255.255.0*
Multicast Filtering Enabled	*OFF*
Default 4x Address for Health	*400100*
Distribution Period	*10*
Health Timeout	*1000*
Data Zone	*400200*

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 / Unity product line symbol.

Table of Global Data Limits

Parameter	Limit
Maximum number of publish variables per device	1
Maximum size for the publish variable	512 Registers = 512 Words (16 bits) = 1 024 Bytes
Maximum number of subscription variables per device	64 (63 if this device is publishing)
Maximum size for the subscribe variables per device	2048 registers = 2048 Words (16 bits) = 4 096 Bytes

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.

Table of Global Data Planning Spreadsheet

Parameter	Description
Parameter Checking	Reserved.
Variable Id	Represents the Data ID on the NOE's Global Data Configuration Web page.
Symbol	Symbolic name for Global Data exchange.
Length (Registers)	Length of Global Data information. Number of 4x registers.
Device Number	Number of devices for the Global Data network. Up to 64.
Variable Public. Status	Automatic information of the correct publication status of the Global Data network. Only by using the Microsoft Excel™ spreadsheet. Information per symbol.
Device Publication Status	Automatic information of the correct publication status of the Global Data network. Only by using the Microsoft Excel™ spreadsheet. Information per device.
Total Publication Size per Node	Publication size for the specific node. The maximum publication size is 512 registers per node.
Total Subscription Size per Node	Subscription size for the specific node. The maximum subscription size is 2 048 registers per node.
Group IP Address Enabled	IP address for multicast networking. Identifies the stations distribution group. The address range is from 224.0.0.0 to 239.255.255.255.
Multicast Filtering Enabled	A check box for Ethernet switches that support multicast filtering.
Default 4x Address for Health	4x register address for the Health bits. This is the register where the Health bits are stored. It has the size of four 4x registers.
Distribution Period	The minimum number of controller scan times before an update will occur.
Health Timeout	The maximum time between received subscriptions before a subscription is declared unhealthy (faulty). The value is measured in milliseconds and can be set to a value that ranges from 50 through 1 000 ms (increase in units of 50 ms)
Data Zone	The starting address for the data. These are the registers where the data information are stored.

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:

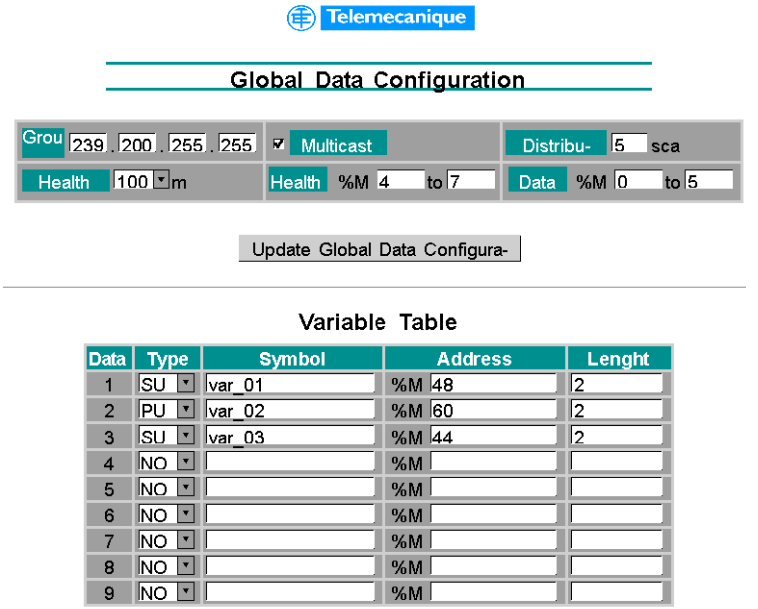
- Distribution period
- Multicast filtering
- Health bit location
- Global Data base address
- Group IP address

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 the Quantum home page, click **Diagnostics**.
2	You will be requested to supply a 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.

Step	Action
5	Click the **Configure Global Data** link. 
6	The **Global Data Configuration** page appears.  In this screen, you may make changes to the configuration.

Configuring Global Data

After you have completed the Modelling System Configuration process using the second method, Copy Configuration, 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, select the **Multicast filtering** check box.
6	Enter %MW word 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 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 %MW words. The ending %MW word field is automatically updated. If you are using the second method, **Copy Configuration**, update **Length** the first time only.
6	When you are finished, click the **Update Global Data Configuration** 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 Diagnostics** page. Follow these links:I **Diagnostics and Online Configuration** I **NOE Diagnostics** I **Global Data**

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 -- 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. Select 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 1 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, -11 and -21 only

6

Overview

This chapter discusses the NOE 771 -00, -01, -11 and -21 modules' I/O scanner capabilities.

What's in this Chapter?

This chapter contains the following topics:

I/O Scanner Concepts

Overview

The I/O scanner is a feature of the NOE 771 0•, -•1, and CPU 651 •0 modules, which allows repeated reading and/or writing to input/output devices.

You can configure the scanner with the Schneider Electric programming packages or directly by using the internal NOE I/O Scanner Web site (NOE 771 -0• and -•1 only). In both ways, you can configure data and transfer it between network nodes without using the MSTR instruction.

I/O Scan List

The I/O scan list is a configuration table that identifies the targets to 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.

There can be multiple instances of the I/O scan list. The individual scan lists for each module are identified by the Quantum backplane slot number where the NOE is installed.

Device Control Block

See Enable/Disable I/O Scanner *(see page 123)*.

Health Block

Each health block bit corresponds to an entry in the I/O scanner table. Each entry in the table represents 1 logical device.

The bits contain the health status for the Modicon Quantum I/O scanner.

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 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.
Ethernet I/O Scanner	Provides high performance cyclic communication service to the controller.

Enhanced Modbus I/O Scanner Features

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

Parameter	Value
Maximum Number of Devices	64: 140 NOE 771 00 (Version 2.2 or earlier) 128: 140 NOE 771 00 (Version 3.0 or later), 140 NOE 771 01, and 140 NOE 771 11 only 128: HE CPU 651•0
Maximum Number of Input Words	4 000
Maximum Number of Input Words	4 000
Health Timeout Value	Individual setting (10 ms to 2 000 ms in 10 ms increments)
Last Value (Input)	Global setting (Zero or Hold)
IP Address	IPv4 Address
Unit ID	User configurable.Default value: 255.
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.

Quantum CPU Type	Number of NOEs Supported
140 CPU 311 10	2
140 CPU 434 12A	6
140 CPU 534 14A	6
140 CPU 651 50	6
140 CPU 651 60	6
140 CPU 671 60	6

Using the I/O Scanner with an IP Router

NOTE: [The I/O scanners in the NOE 771 •1 and HE CPU 651 •0 modules send out requests with a Time To Live (TTL) of 10, which allows passage through multiple routers.

Enable/Disable I/O Scanner

Device Control Block

The Enable/Disable I/O Scanner function reduces network traffic volume. Use the Device Control Block bits to enable/disable devices as follows.

Registers	The Device Control Block consists of registers either 8 words or 4 double words. Contents of the registers are mapped in the controller's memory. Each bit corresponds to an entry in the table (see the tables below.)
Disabling Devices	Each I/O scanner device can be disabled. To disable individual devices: **1.** Select the Device Control Block check box. **2.** Set the associated bit = 1.
Mapping Device Control Block Bits to I/O Scanner Entry Numbers (#)	See the table below for mapping entry numbers to bits. Each entry number represents a logical device on the network.
Setting Bits	If Device Control Block bit is set to • 0 = Device is enabled. • 1 = Device is disabled.

Mapping Device Control Block Bits to I/O Scanner Entry Numbers (#)

Single Word (W) Register (%MD*x*:4)																	
W1 (%MW x+1)	Table Entry #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W2 (%MW x+2)	Table Entry #	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W3 (%MW x+3)	Table Entry #	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							Word 4 through Word 7 (Table Entry 49 through 112)										
W8 (%MW x+8)	Table Entry #	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

NOTE: Bits are counted from right to left starting from bit 0 (low bit). Examples: to configure %MD1:4 as a device control block in the I/O Scanner table, use %MW2, bit 15 to enable or disable table entry 1. Use %MW3, bit 15 to enable or disable table entry 17.

Double Word (DW) Register (%MD*x*:4)																	
DW1 (%MDx[0])	Table Entry #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DW2 (%MDx[1])	Table Entry #	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Table Entry #	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
	Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							Word 3 through Word 4 (Table Entry 49 through 112)										
DW5 (%MDx[4])	Table Entry #	113	114	115	116	117	118	119	120	12	122	123	124	125	126	127	128
	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

NOTE: Bits are counted from right to left starting from bit 0 (low bit). To enable or disable the Device Control Block for I/O Scanner table entries 17 through 31, use %MDx[1], not %MDx[0]. For example, to configure %MD1:4 as a device control block in the I/O scanner table, use %MD1[0], bit 31 to enable or disable table entry 1. Use %MD1[1], bit 15 to enable or disable table entry 17.

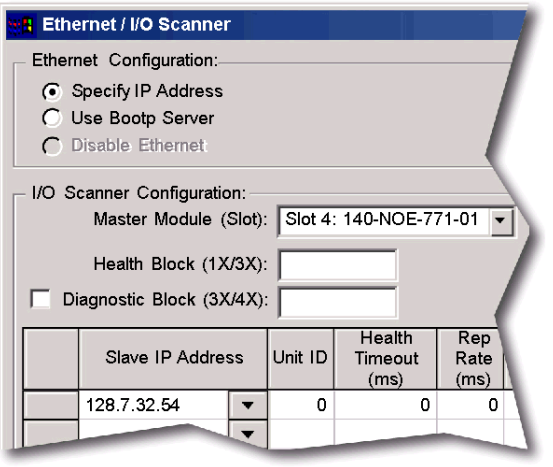
Configuring the I/O Scan List Using Concept

Overview

After the NOE 771 -00, -01, or -11 has been configured using Concept *(see page 51)*, 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

In the **Ethernet / I/O Scanner** dialog box, 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 0, the health bit is set to true once communications are established, and it is never cleared.

Rep Rate

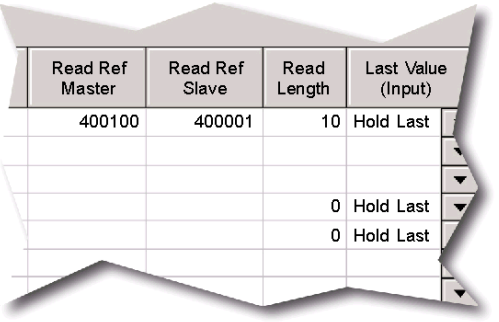
Use this column to specify the lower bound in milliseconds (ms) between transactions to this node. Valid values are 0 to 50 000 ms (1 min.). The NOE module takes this value and rounds up to a multiple of 17 ms. The update of I/O is synchronized to the CPU scan. If the CPU scan is greater than the configured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

For example, if you specify 10 ms, the number is rounded up to 17 ms. If the controller's scan time is 5 ms, the time between transactions must be greater than or equal to 1 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.

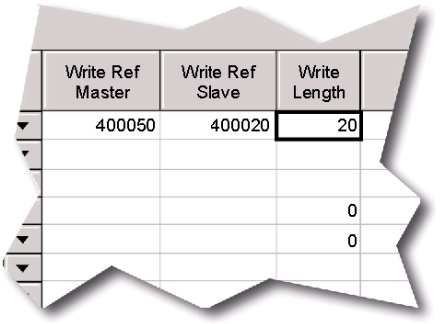


Read Ref Master	Read Ref Slave	Read Length	Last Value (Input)
400100	400001	10	Hold Last
		0	Hold Last
		0	Hold Last

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.



Write Ref Master	Write Ref Slave	Write Length
400050	400020	20
		0
		0

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 1, the remote device is healthy. It the health bit is 0, 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

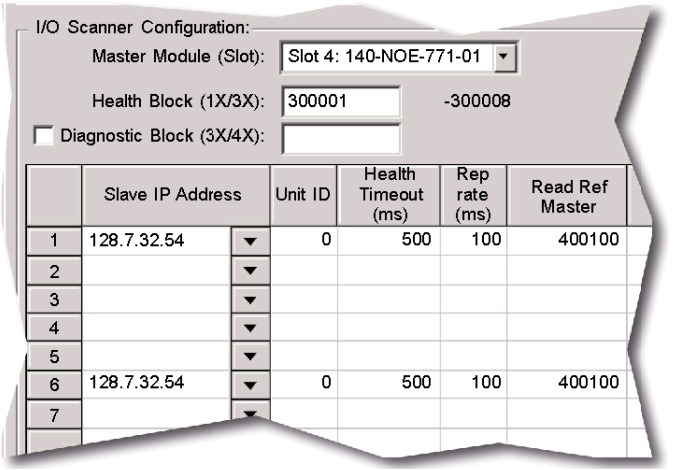
Word 2 Bit Positions															
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

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 field.



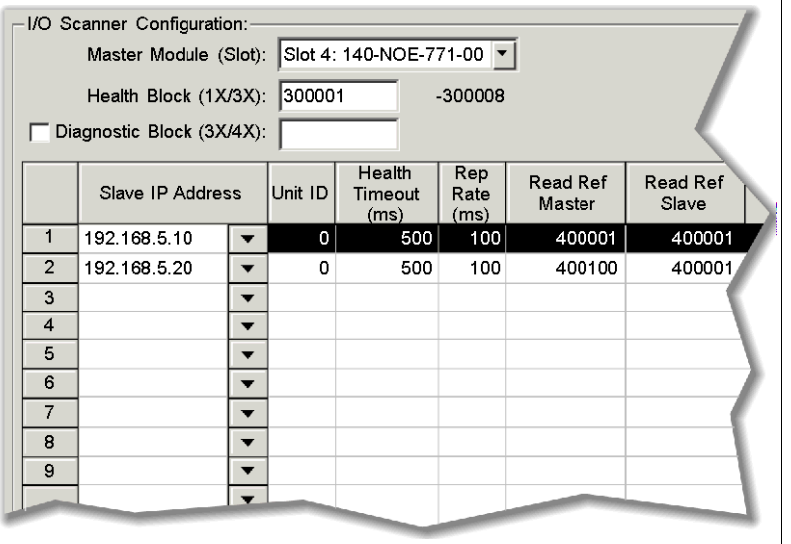
Completing the I/O Configuration

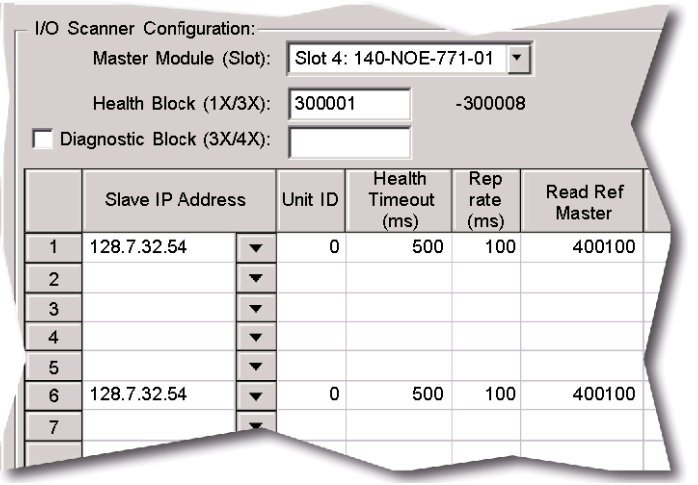
Overview

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.

Step	Action
1	Select the row you want to copy by clicking the row number at the far left. 
2	Click **Copy**.

Step	Action
3	Select the row where you would like to paste the data by clicking the row number at the far left. 
4	Click **Paste**.

Cut and Paste

To move a row within the configuration list, follow the directions for copying, except click **Cut** instead of **Copy**.

Delete

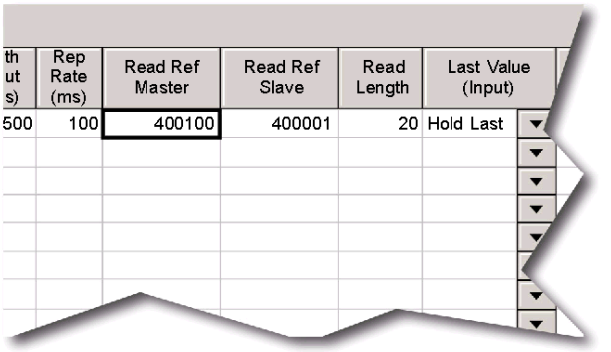
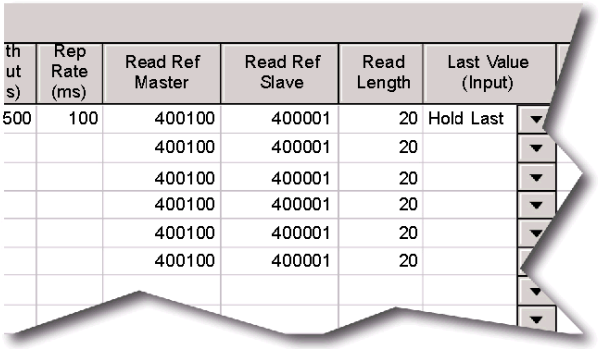
To delete a row from the configuration list, select the row by clicking the row number at the far left. Then click **Delete**.

Sort

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

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

Step	Action
1	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 1 contiguous block of cells with the data to be copied in the first row. You cannot select 2 separate blocks. 
2	Click **Fill Down**. **Result**: The data from the first row is copied into the selected cells. 

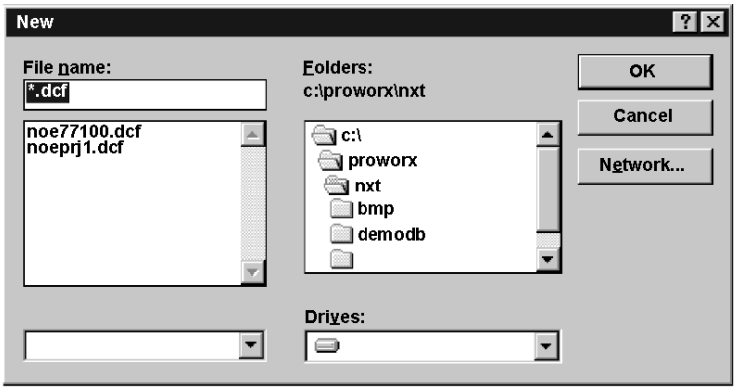
Configuring the I/O Scan List Using ProWORX NxT

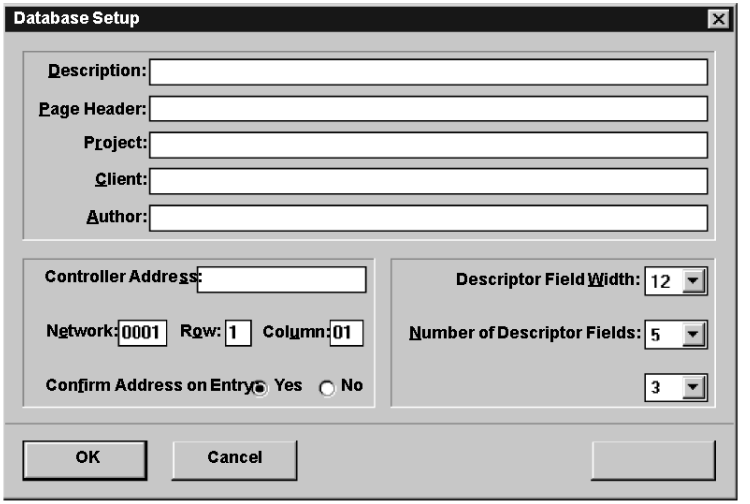
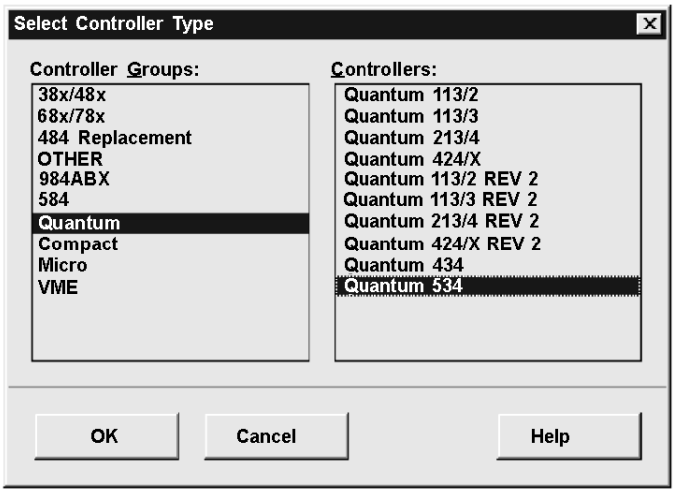
Overview

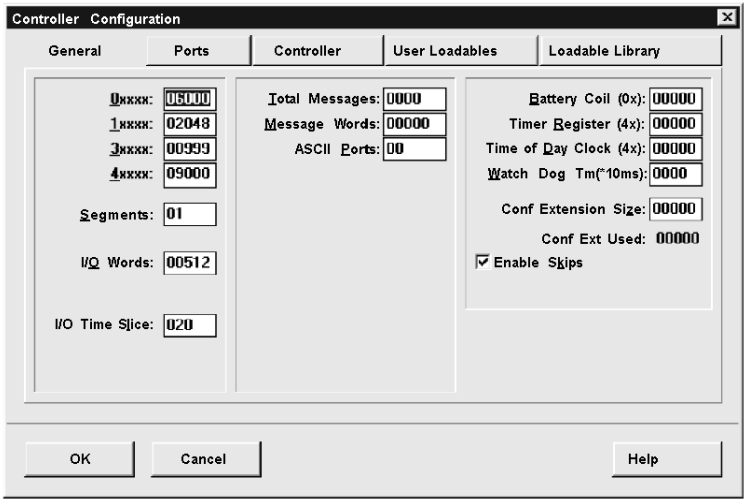
This section discusses how to configure the NOE 771 module from your programming panel using ProWORX NxT program. This process assumes you have switched to an Ethernet network so you can choose I/O scanner. You can now configure data blocks to be transferred between controllers on a TCP/IP network.

Selecting Your PLC

The following steps describe how to select a PLC.

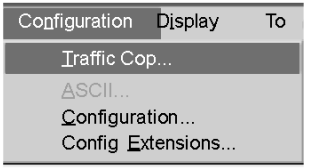
Step	Action
1	Open ProWORX NxT on your programming panel (PC).
2	Click **File** → **New**.  **Result**: The New dialog box appears.
3	Type a file name in the **File name** field. Select the drive from the **Drives** list. Select the directory in the **Folders** field where you want to save the new database. Click **OK**. 

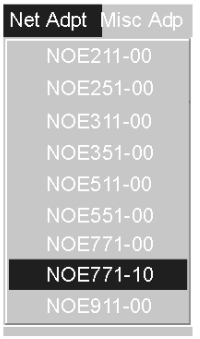
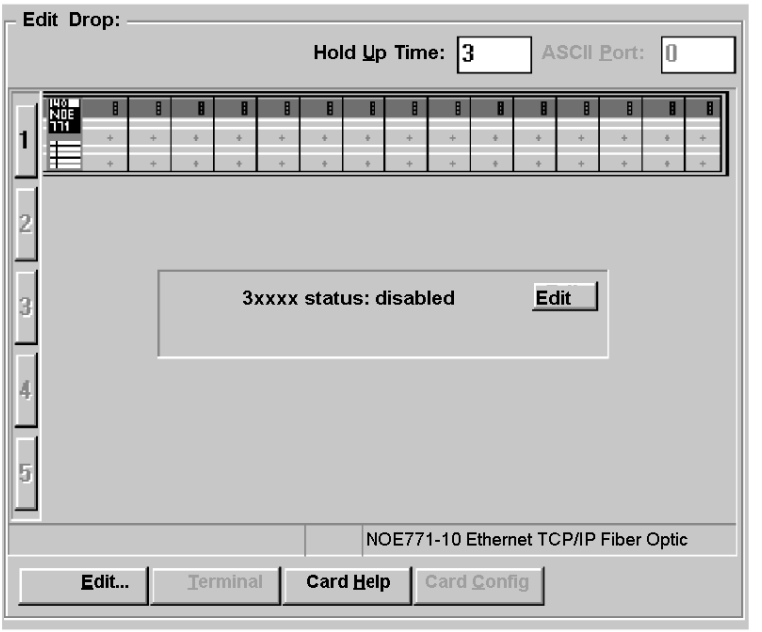
Step	Action
4	**Result**: The Database Setup dialog box appears. Fill in the fields as necessary. Click **OK**. 
5	**Result**: The Select Controller Type dialog box appears. From the **Controller Groups** list on the left, select **Quantum**. From the **Controllers** list on the right, select the CPU that is installed in your Quantum backplane. Click **OK**. 

Step	Action
6	**Result**: The Controller Configuration dialog box appears. You must define a value in the **Conf Extension Size** field on the right side of the Controller Configuration screen. This value is the amount of memory you require. Click **OK**. 

Accessing and Editing the Traffic Cop

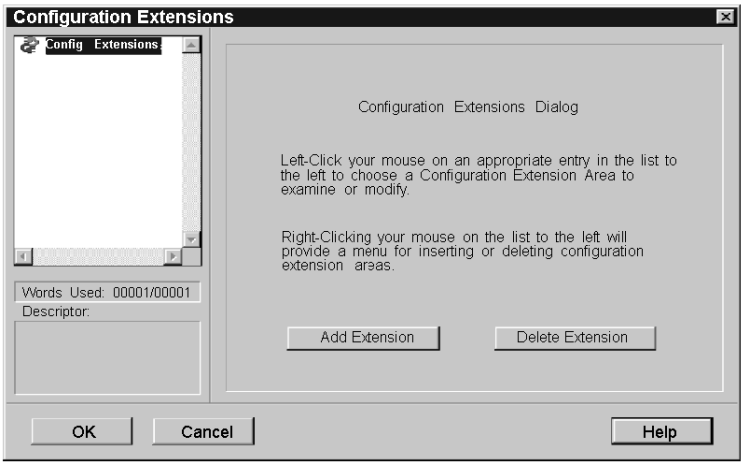
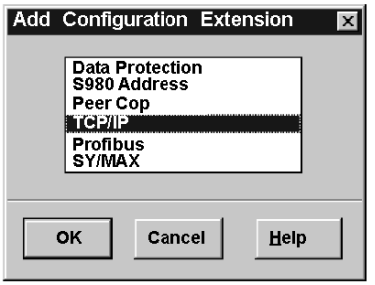
The following steps describe how to access and edit the Traffic Cop.

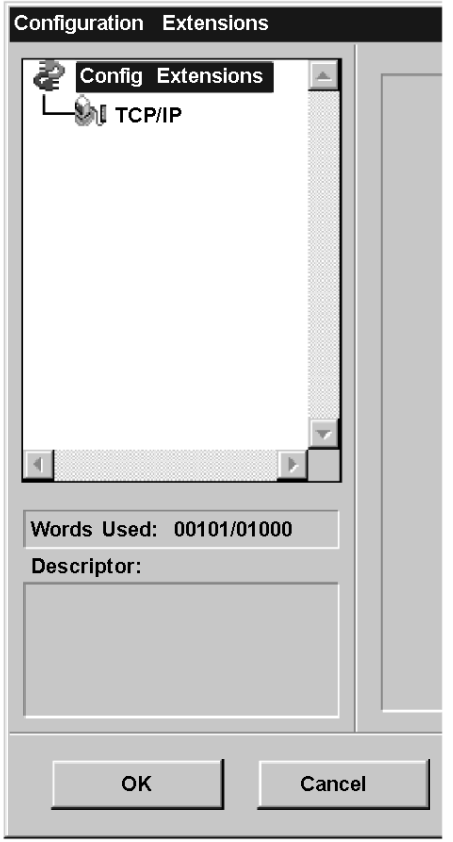
Step	Action
1	Click **Configuration** → **Traffic Cop**. 
2	From the Quantum Traffic Cop menu on the left, click the **+** sign to expand the Traffic Cop tree. Choose the rack and slot where you want the NOE 771 module inserted.

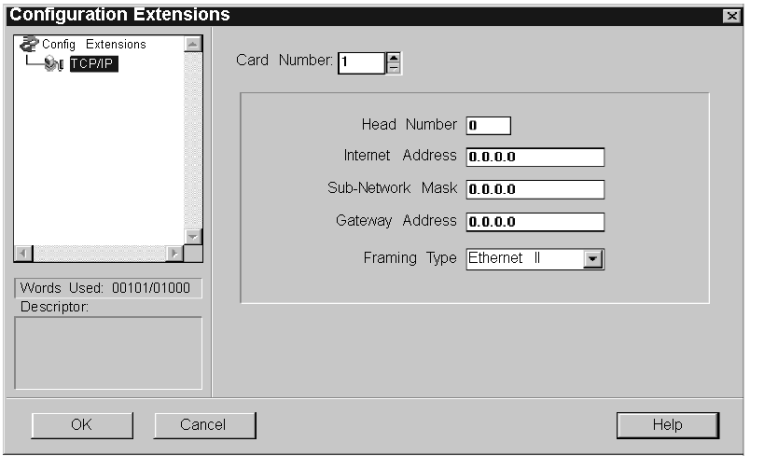
Step	Action
3	In the **Net Adpt** list, click the respective **NOE771-••** module.  **Result**: The NOE 771 module is inserted into the specified location, in the Edit Drop field of the Traffic Cop screen. 
4	Click **OK**.

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	Click **Configuration** → **Config Extensions**.  **Result**: The Configuration Extensions dialog box appears. 
2	Click the **Add Extension** button. **Result**: The Add Configuration Extension dialog box appears. 

Step	Action
3	Select **TCP/IP** from the list, and click **OK**. **Result**: The TCP/IP Configuration Extension is added to the left menu of the Configuration Extensions screen. 

Step	Action
4	Click **TCP/IP** in the Config Extensions menu. **Result**: The details of the TCP/IP configuration appear on the right side of the Configuration Extension screen. 
5	Click the up / down arrows to enter the card number (rack number) in the Card Number field. 
6	Complete the fields for each card number. The **Head Number** field represents the rack in which the NOE 771 module is located. Click **OK**.

Setting Up the I/O Scanner

At this point, you are ready to set up the I/O Scanner. The I/O Scanner provides data transfer between 2 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 I/O groups to be scanned.
- Configure the transaction parameters.
- Set the hardware clock for when the data is to be collected.

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

Step	Action
1	In the Network Editor, click **Configuration** → **Config Extensions**. The Configurations Extensions dialog box appears.
2	In the Config Extensions tree, right-click **Config Extensions** → **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**: All 1xxxxx addresses are based on a 16-bit boundary. **Example**: 100001, 100017, 100033, etc.

Configuring the Transaction Parameters

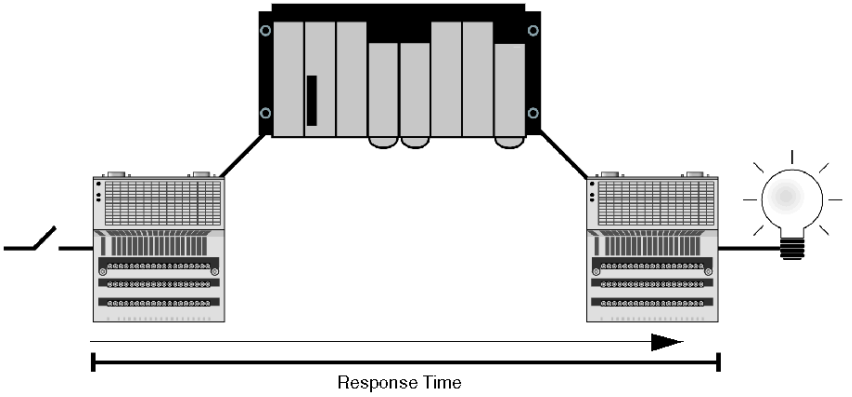
The following steps describe how to configure the transaction parameters.

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

I/O Scanner Response Times: Remote Input to Remote Output

Measurement Setup

The set of curves below illustrates Quantum PLC response times when a signal is sent from a remote input module to a remote output module through a Quantum PLC:

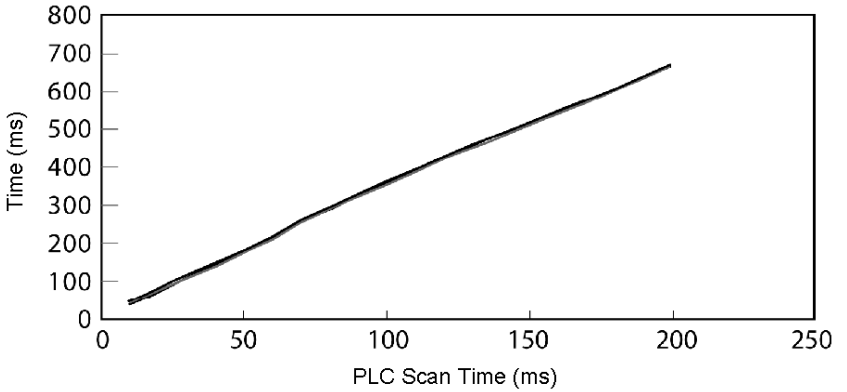


The signal is:

1 triggered by a Momentum input module with a response time of ~2 ms
2 scanned into the Quantum PLC at a repetitive rate of 0 ms
3 copied to another internal variable within the PLC
4 written to a Momentum output module with a response time of ~2 ms

140 CPU 434 12A with 140 NOE 771 x1 Module

The 140 CPU 434 12A with an NOE 771 x1 (v3.3) was used for the following measurements:



The plot shows four lines representing the number of scanned devices:

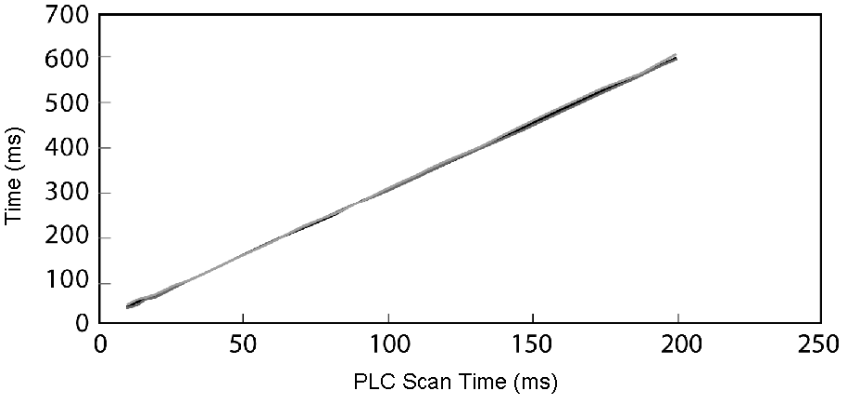
- 1 device
- 8 devices
- 16 devices
- 32 devices

The four lines are indistinguishable at this scale because the response times are so similar. You can see the similarity in the following table of data from which the above graph was plotted:

Number of Devices to Scan	**Time from Scanned Device Input to Scanned Device Output (ms)**				
434 12A + NOE 771 x1 (v3.3)	10 ms scan	20 ms scan	50 ms scan	100 ms scan	200 ms scan
1 device	41	73	179	358	665
8 devices	42	75	180	360	666
16 devices	44	77	182	361	668
32 devices	46	79	185	364	671

140 CPU 65 150 with NOE 771 x1 (v3.3)

The 140 CPU 65 150 with an NOE 771 x1 (v3.3) is used used for the following measurements:



The plot shows four lines representing the number of scanned devices:

- 1 device
- 8 devices
- 16 devices
- 32 devices

The four lines are indistinguishable at this scale because the response times are so similar. You can see the similarity in the following table of data from which the above graph was plotted:

Number of Devices to Scan	**Time from Scanned Device Input to Scanned Device Output (ms)**				
65150 + NOE 771x1 (v3.3)	10 ms scan	20 ms scan	50 ms scan	100 ms scan	200 ms scan
1 device	35	61	153	302	602
8 devices	36	62	154	303	603
16 devices	38	64	155	305	606
32 devices	40	66	157	307	609

Embedded Web Pages

7

Overview

This chapter presents the contents of the embedded Web pages contained in the Quantum 140 NOE 771 •• 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

Each Quantum 140 NOE 771 •• 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.

Pages on the embedded Web site display the following information.

- configurable menus of the address server (BootP, DHCP, and 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 1•, FactoryCast / Real Time, modules offer these additional pages.

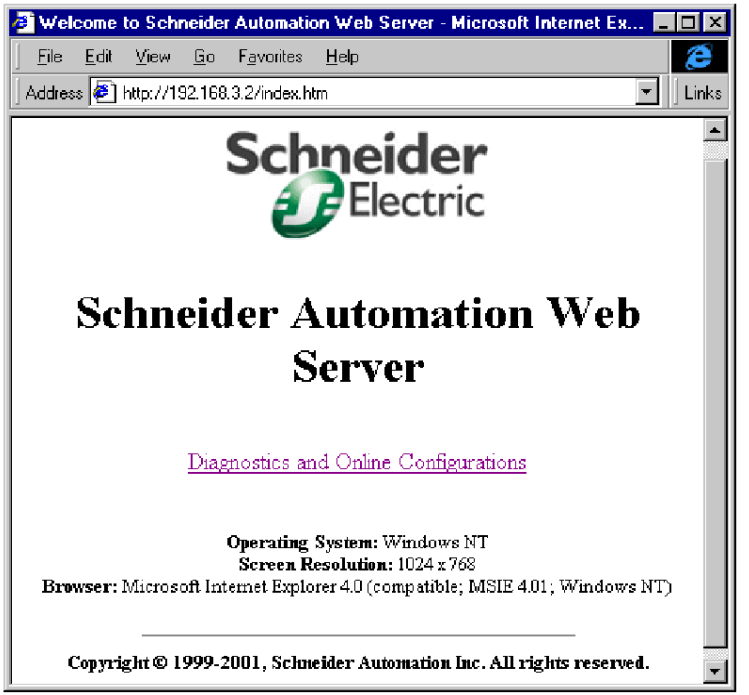
- configuration and status for Global Data (Publish / Subscribe)
- bandwidth monitoring
- I/O Scanner status
- Modbus messaging status

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

For information about the additional functionality provided by the FactoryCast system in the 140 NOE 771 1• modules, see the *FactoryCast User's Guide* (31001229).

Accessing the Module's Home Page

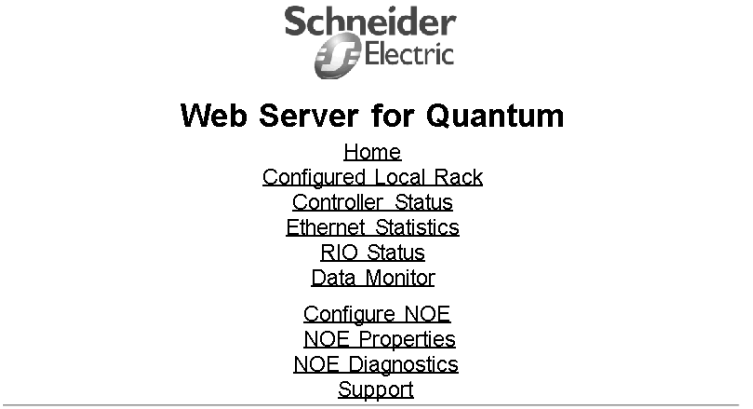
Before you can access the module's home page, you need its full IP address or URL from your system administrator. Type the address or URL in the address or location box in the browser window. After you do this, the Schneider Automation Web Utility home page displays. The following steps describe how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

Step	Action
1	Click the **Diagnostics and Online Configurations** link. 
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 the **Data Editor / Data Monitor**. **Note**: The default User Name is USER, and the default password is USER. We recommend that the system administrator change both during module installation.

Quantum Welcome Page

Overview

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



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
Graphic Editor (only 771 01, -11, -21)	Displays the Graphic tool for creating input / output user screens (only with FactoryCast)
Data Monitor / Data Editor	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

Configured Local Rack Page

Overview

The Configured Local Rack page displays a visual representation of the current configuration.

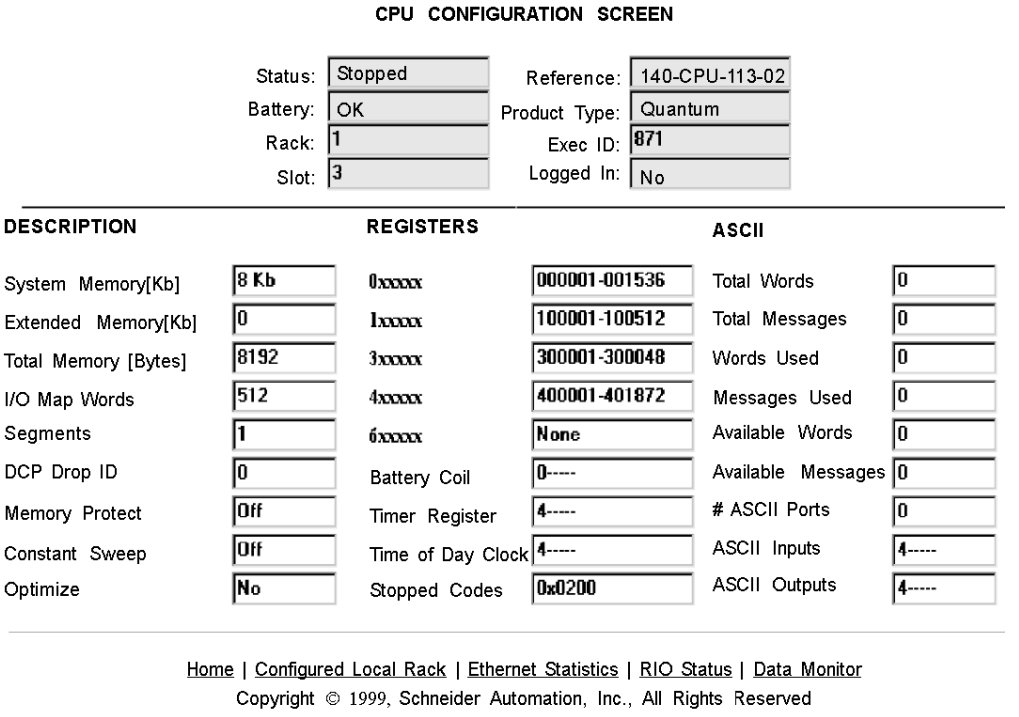
The following table details the links on the Configured Local Rack page. To view the pages related to each of these topics, click the respective topic.

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
Graphic Editor (only 771 01, -11, -21)	Displays the Graphic tool for creating input / output user screens (only with FactoryCast)
Data Monitor / Data Editor	Allows access to the Quantum PLC Data with editing capabilities

CPU Configuration Screen Page

Overview

In the following figure, the top 8 fields identify the CPU configuration. More information about these fields follows.



Description Fields

The following table describes the description fields on the CPU Configuration screen.

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.

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.

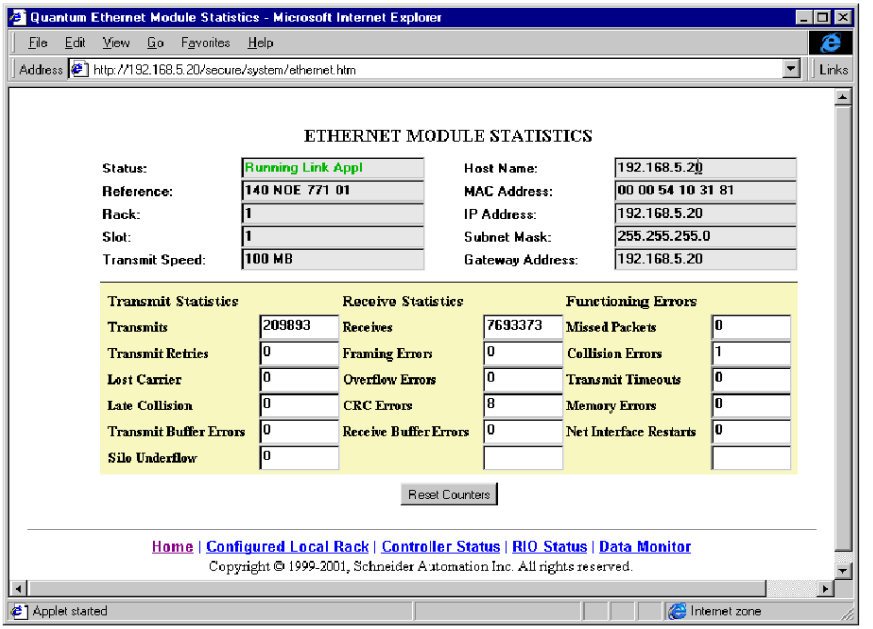
Link	Results
Home	Displays the Quantum Welcome Page
Configured Local Rack	Displays the Quantum Local Rack with NOE and CPU
Ethernet Statistics	Displays the Ethernet Module Statistics with the Reset Counters link
RIO Status	Displays the Remote I/O Communications Status
Graphic Editor (only 771 01, -11, -21)	Displays the Graphic tool for creating input / output user screens (only with FactoryCast)
Data Monitor	Allows access to the Quantum PLC Data with editing capabilities

Ethernet Module Statistics Page

Overview

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 0 by clicking the **Reset Counter** button.



Ethernet Module Statistics Page Links

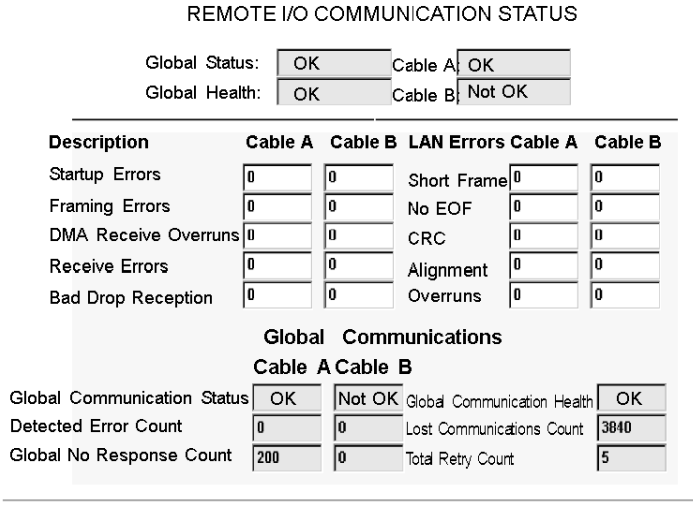
The following table describes the links on the Ethernet Module Statistics page.

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
Graphic Editor (only 771 01, -11, -21)	Displays the Graphic tool for creating input / output user screens (only with FactoryCast)
Data Monitor	Allows access to the Quantum PLC Data with editing capabilities

Remote I/O Communication Status Page

Overview

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 1•, the FactoryCast Web server.

For more information about the Remote I/O communications status, see the *Remote I/O Cable System Planning and Installation Guide* (890 USE 101).

Remote I/O Communications Status Page Links

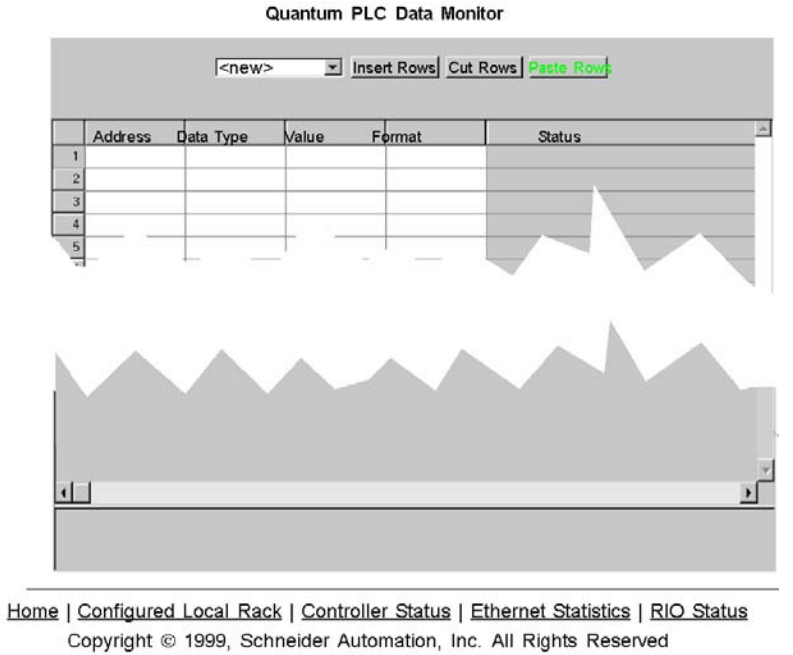
The following table describes the information that is available for each cable, using the links on the Remote I/O Communication Status page.

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

Overview

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



You can add, delete, and copy Quantum PLC data, as follows.

- Click the **Insert Rows** button to insert additional rows of data.
- Click the **Cut Rows** button to delete specific rows of data.
- Click the **Paste Rows** button to copy rows of data.

PLC Data Monitor Page Links

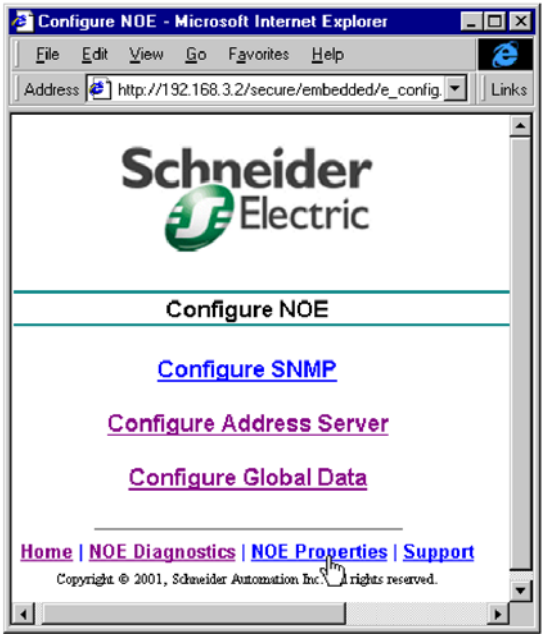
The following table describes the links on the Quantum PLC Data Monitor page.

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
Graphic Editor (only 771 01, -11, -21)	Displays the Graphic tool for creating input / output user screens (only with FactoryCast)

Configure NOE Page

Overview

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

Overview

SNMP may already be configured on your NOE 771 ••. If 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 ••.

Step	Action
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

When you click the **Update SNMP** button, a new page displays with 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 •0 Transparent Factory modules.

NOTE: The chapter Address Server Configuration / Faulty Device Replacement *(see page 171)* describes how to configure the address server for the 140 NOE 771 •1, FactoryCast Web server modules, and also describes the BootP process.

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 is used for the 140 NOE 771 •0 modules.

Ethernet Configuration

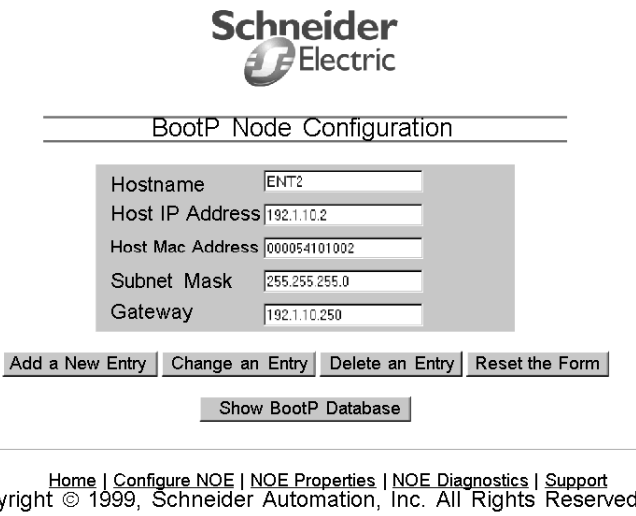
No BootP Database File Exists. Please click button to configure it.

Configure BootP Entry

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

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**: The **BootP Node Configuration** screen appears. 
2	If this is an initial BootP configuration, fill in the fields on the **BootP Node Configuration** screen, and click the **Add a New Entry** button.

Initial Configuration

The following table describes the fields to complete 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** screen.
2	Click the **Change an Entry** button. **Result**: A new entry will be made at the bottom of the Database Table, and you will get a successful entry message.
3	Click **Configure NOE** to return to the Configure NOE screen.
4	Click **Configure BootP**.
5	Enter the information to be old information.
6	Click **Delete an Entry**.

Deleting the BootP Database File

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

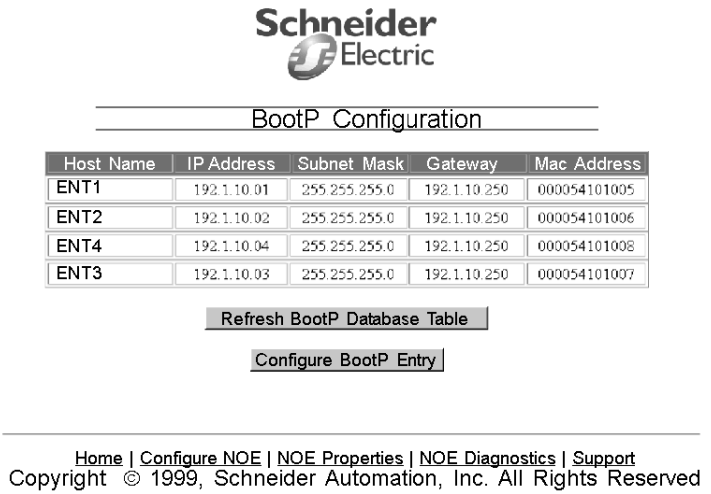
Step	Action
1	Enter the new information for the item to be deleted.
2	Click the **Delete an Entry** button. **Result**: A **delete successful** message is displayed.
3	Click **Configure NOE** to return to Configure NOE screen.
4	Click **Configure BootP**.
5	Click **Refresh BootP Database Table** to view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

Resetting the Form

To clear the fields on the BootP Node Configuration screen, click 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, click the **Show BootP Database** button.



Configure BootP Page Links

The following table describes the links on the BootP Node Configuration screen.

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

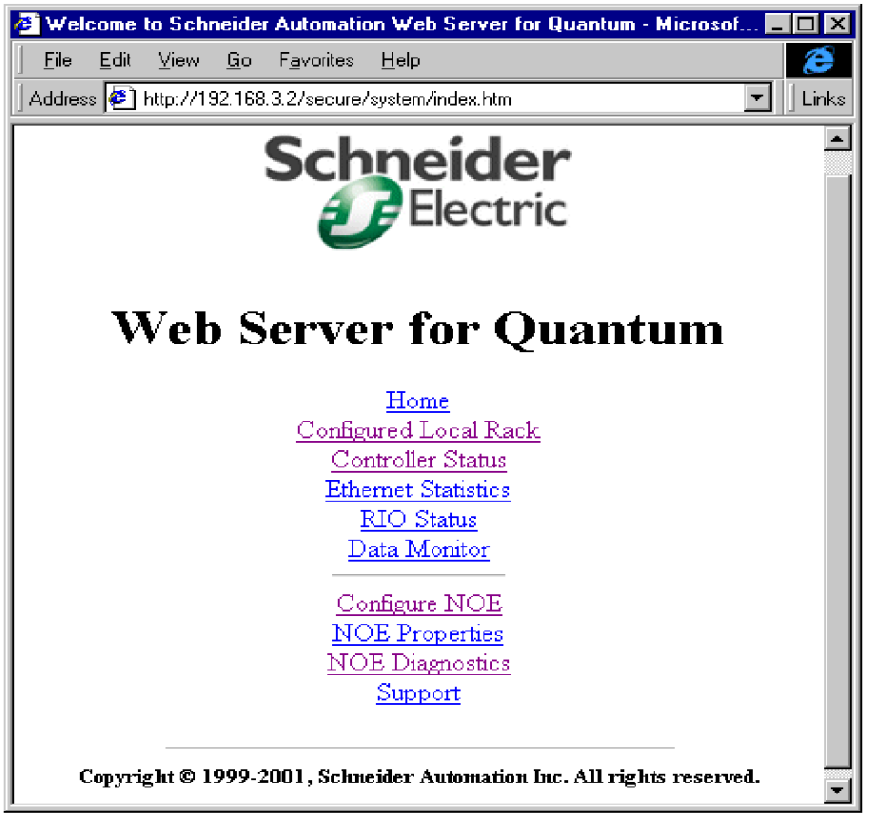
The 140 NOE 771 •1 embedded Web server provides Web pages that you may use to diagnose Transparent Factory / Real Time services. These services are listed below:

- 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.

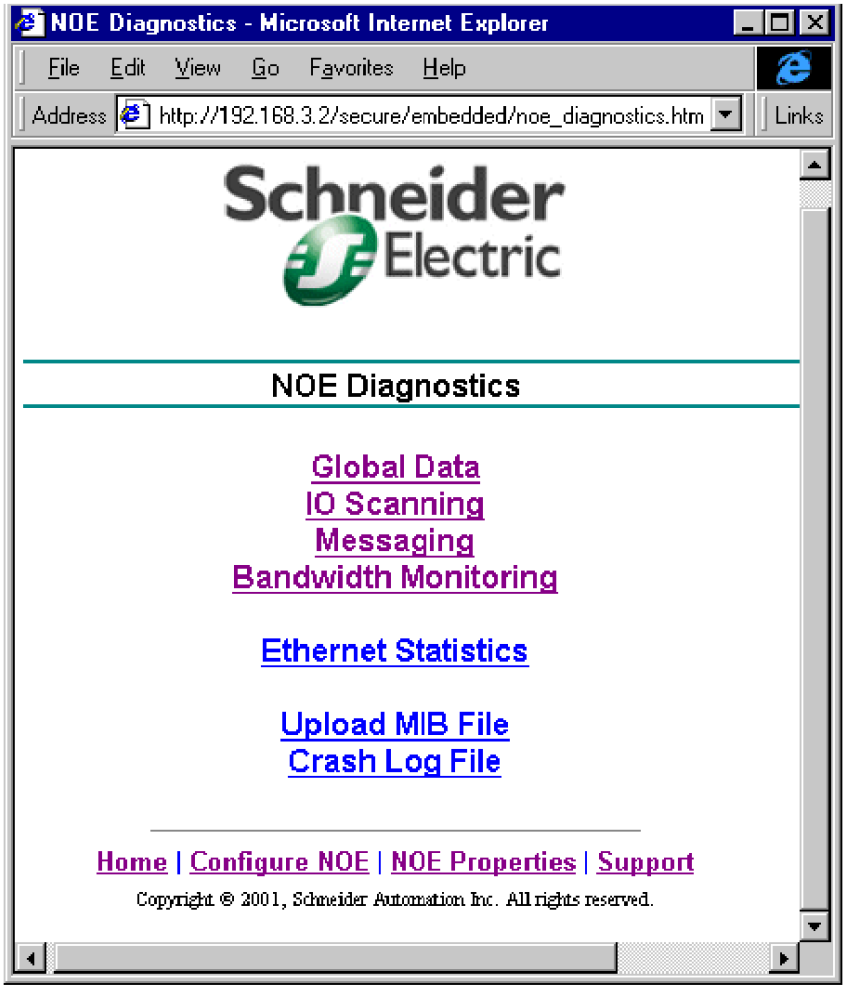
Accessing Web Diagnostics

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



Click the **NOE Diagnostics** link, which opens the **NOE Diagnostics** screen.

On the NOE Diagnostics screen, click the desired service link.



The first 4 links are the diagnostics.

- global data
- I/O scanning
- messaging
- bandwidth monitoring

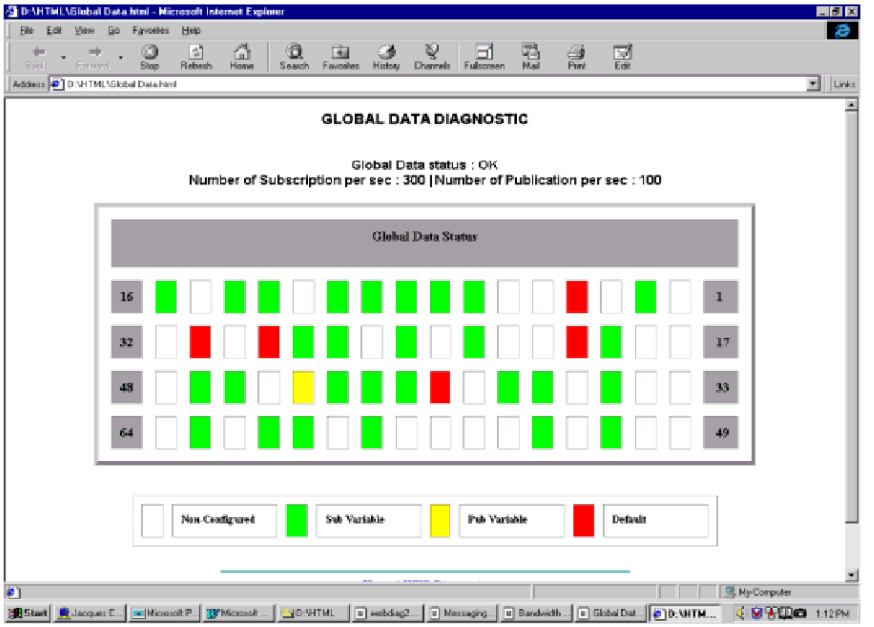
The last 3 links access other functions.

- Ethernet statistics
- upload MIB file
- crash log file

Global Data Diagnostics Page

The **Global Data Diagnostics** screen displays information generated by the global data service. At the top of the page the following 3 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.

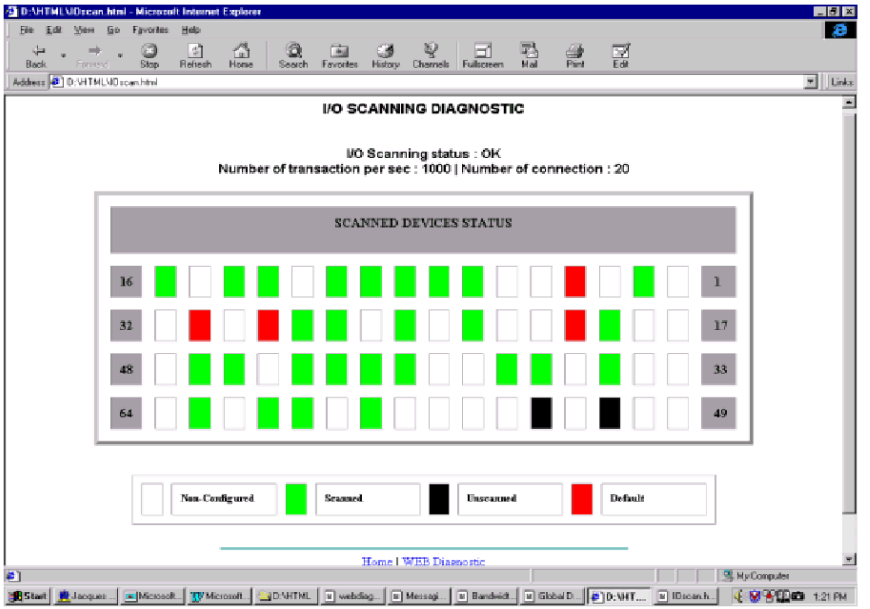
This page also displays the complete status for all variables published and subscribed within the same distribution group. Every variable is identified by its data ID. At the bottom of the page, 4 color-coded boxes indicate the status of the variables.

- white = all non configured variables
- green = healthy subscribed variables
- black = healthy published variables
- red = 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 3 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, 4 color-coded boxes indicate the status of the devices.

- white = all non configured devices
- green = all scanned devices
- black = all temporary unscanned devices
- red = all devices in a default state

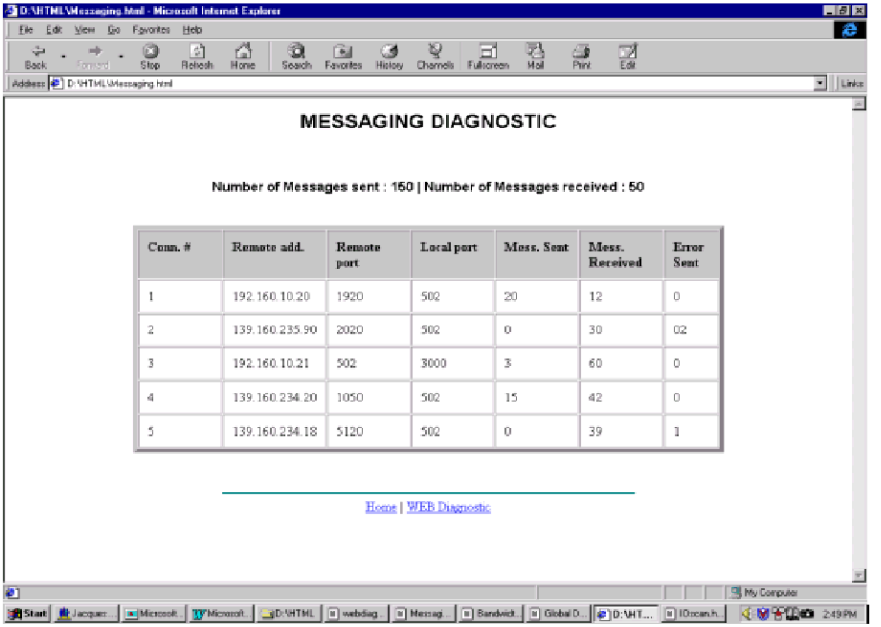
If red displays in the **Fault** variable box, you should check the system for problems.

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:

- 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.

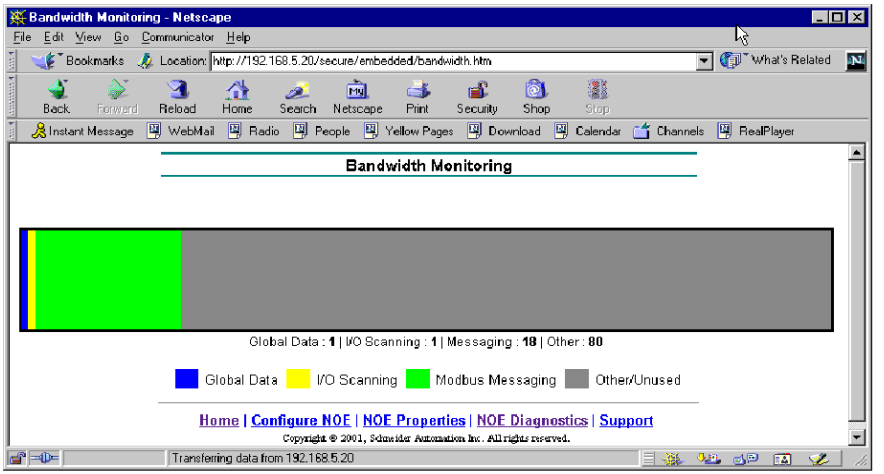


For each connection (numbered from 1 to 64) the following information is given:

- 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 and distribution of NOEs on your system.



The page reports 4 statistics one each for the 3 services and 1 for **other**.

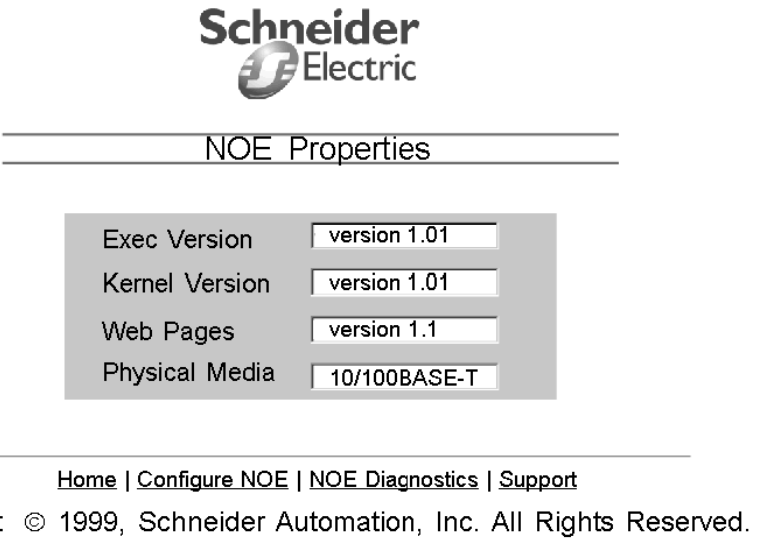
- blue: global data (usage expressed as a % of maximum number of messages/second)
- yellow: I/O scanner (usage expressed as a % of maximum number of messages/second)
- green: Modbus messaging (usage expressed as a % of maximum number of messages/second)
- gray: other/unused (usage expressed as a % of maximum number of messages/second.)

Total percentages equal 100%.

NOE Properties Page

Overview

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



NOTE: The NOE Properties screen is *information only*. The fields cannot be changed.

NOE Properties Page Links

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

Link	Results
Home	Returns you to the Quantum Welcome screen
Configure NOE	Provides the ability to configure and change the NOE through the Ethernet Configuration screen
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

Overview

The Crash Diagnostics page displays a crash log file when a crash has occurred and a status message when no crash has occurred.

Click **Clear Crash Log File** to clear the file.

Crash Log Diagnostics Links

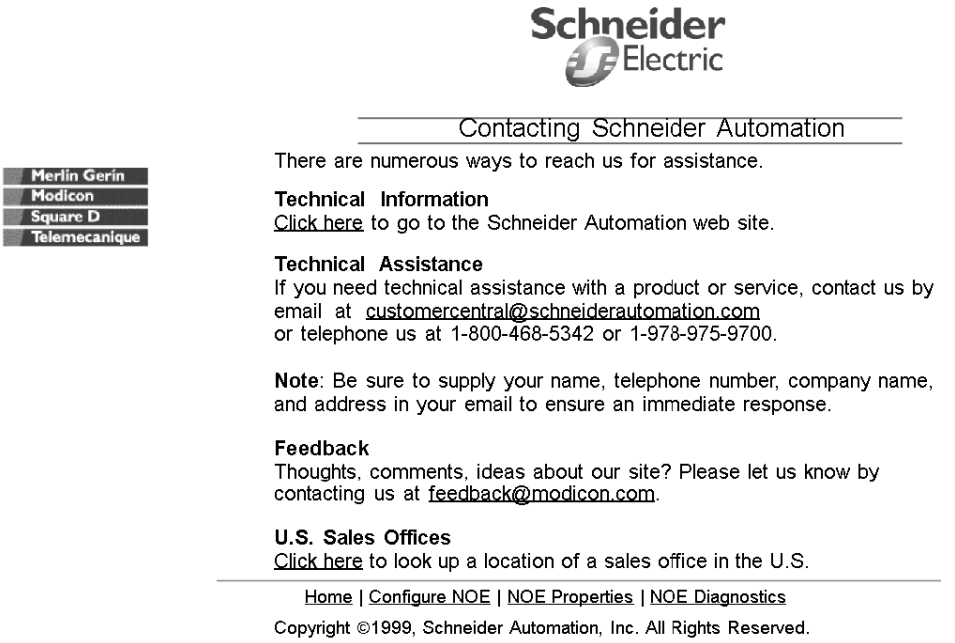
The following table describes the links on the Crash Log Diagnostics page.

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 figure shows the Contacting Schneider Automation page, which contains information about how to obtain support for the NOE 771 •• modules.



Address Server Configuration / Faulty Device Replacement

8

Overview

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 BootP server behavior
   Enter the MAC address and IP configuration. The NOE BootP server will provide the IP configuration when the device sends a BootP request.
2. Faulty Device Replacement (FDR) behavior
   Enter the role name or the MAC address of the device. The device will send its role name or the MAC address 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 a FDR-compliant device is to download its 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 entering either the MAC address or the role name on the **Address Server Node Configuration** page. You may enter only 1 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 the user assigns to a device, a logical name that has a meaning within the application.

Some examples of a role name are:

- **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 a method of handling device replacement without disrupting the system or interrupting service. If a device fails, 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 working 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 3 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 get the 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 to enter the role name into the new 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 the 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 work correctly the following must be observed.

- Address class and subnet class configured for the devices must match.
- Address class of the NOE and of the devices must be the same.

Operating on a Corporate Network

NOTE:

- 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 have to 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

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

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 an entry, the following alert message appears.

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

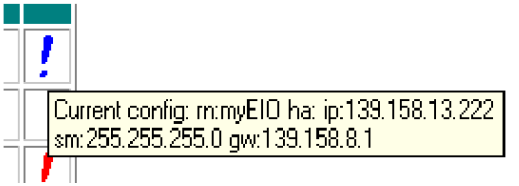
Error Icon If an entry is not loaded in the DHCP server or loaded with a different configuration, an icon of an exclamation point displays in the seventh column. The icon informs you of the difference between the current and stored information.

- 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 displays one of the following messages.

- 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. Fill in these 3 fields: **Device IP address**, **Subnet Mask**, and **Gateway**.

Select either the **Role Name** or **Device MAC address** field. When you select 1 field, the other is not available.

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, click 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, click 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 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 3 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 4 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

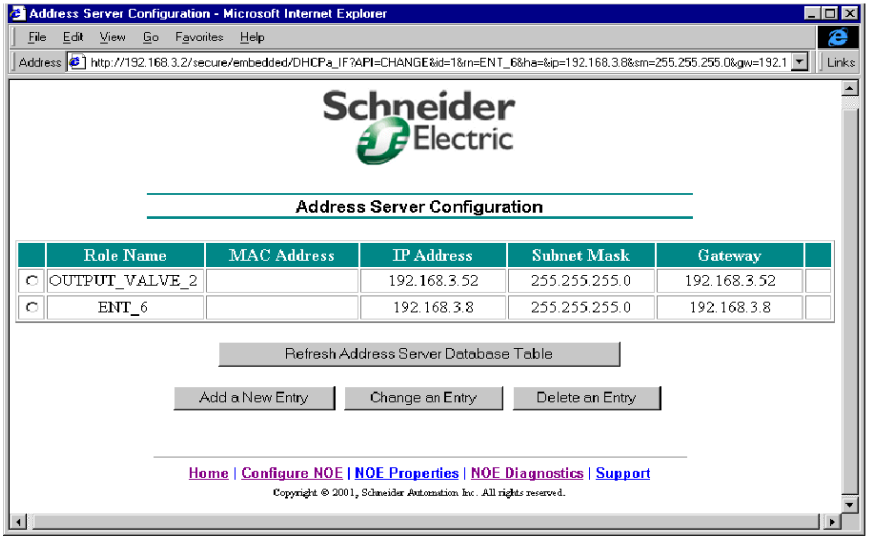
To configure the address server you use Web pages generated by the embedded Web server. The first page that appears is the **Address Server Configuration** page. The first column contains buttons used for selecting devices. The Address Server Configuration page displays configuration information for each device in the system and has 7 columns in the table.

Displayed on this page is information about:

- role name
- MAC address
- IP address
- subnet mask
- gateway

An 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.



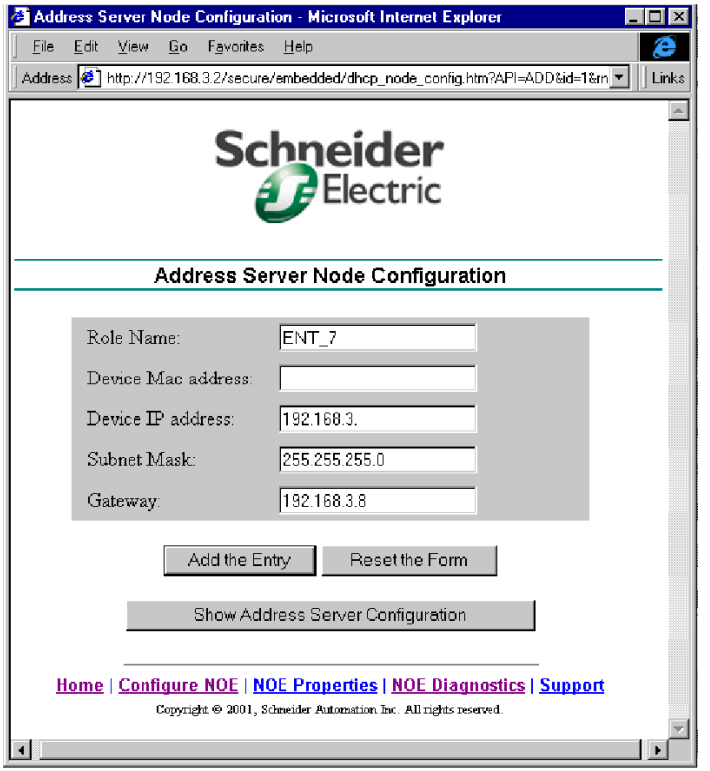
Choosing Options

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

- select an entry
- add an entry

- change an entry
- delete an entry

Below are descriptions of each options.

Option	Description
Select an entry	When the page displays, by default, no entries are selected. Use the buttons in the first column to select an entry. You can only select 1 entry at a time.
Add an entry	When you click the **Add a New Entry** button, 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 4 of the 5 fields of the dialog box. Only the role information field is blank. We recommend you enter a role name, for example *ENT_7*. If you did not select a device, default values appear in all the fields.
Change an entry	Before using this button, you must select an entry in the database by selecting 1 of the buttons in the first column. If you do not select an entry, an error message appears. When you click the **Change an Entry** button, the Address Server Node Configuration page appears. The information displayed is for the device selected. 

Option	Description
Delete an entry	Before selecting this button, you must select an entry in the database by choosing 1 of the buttons in the first column. If you do not select an entry, an error message appears. The entry you select will be removed from the database. Before completely deleting an entry, a warning message appears. ● Click **Yes** if you want to delete the entry. ● Click **No** if you do not want to delete the entry. If you click **Yes**, a dialog box appears, asking you to confirm the deletion.  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 to z (lower case)
  - A to Z (upper case)
  - _ (underscore)
- bad **MAC address**
  The valid MAC address characters are:
  - 0 to 9
  - A to F
- wrong **IP address**
- wrong **subnet mask**
- wrong **gateway**
- double entry
  Each entry must have a unique role name or MAC address. If you enter a duplicate role name or MAC address, the system highlights the error.

Erroneous errors are not loaded into the DHCP server. Therefore, you must correct the errors before loading.

There are 2 ways to correct the errors.

- Through the Web page: Make the changes on the Web page and submit the change.
- Through the address server configuration file: Make the changes in the file and reboot the server.

Hot Standby

9

Overview

The Quantum Ethernt module offers a Hot Standby configuration available for Quantum controllers.

What's in this Chapter?

This chapter contains the following topics:

Quantum Hot Standby for Unity Pro

The Hot Standby Solution

⚠ CAUTION

COMMUNICATION FAILURE

Whenever possible, use of a switch (not a hub) to connect the NOE modules to each other or to the network.

Schneider Electric offers switches. Contact a local sales office for more information.

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

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Design your application so that unmonitored modules support communication only to noncritical parts of the application.

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

In the hot standby solution, two identically configured PLC systems are set up to control the same application. One PLC, the primary, runs the application and updates the other secondary (standby) PLC. The standby maintains awareness of the application status but does not perform any control functions. In the event of a failure in the primary PLC, the standby PLC takes over the primary PLC responsibilities. When the PLC that has failed becomes operational again, it comes back up in the hot standby system as the new secondary PLC.

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 2 NOEs to the same switch minimizes the probability of a communication failure.

NOTE: Schneider Electric recommends that you use a switch (not a hub) 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 1 of 2 Hot Standby actions.

If the NOE . . .	Then
Detects that the new Hot Standby state is either primary or standby	The NOE changes the IP address.
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 switch 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 following table identifies the Ethernet services that are available and unavailable in a hot standby solution.

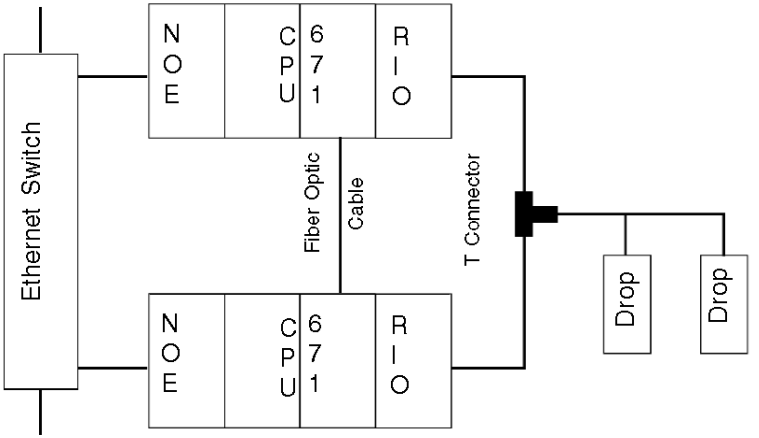
Service	NOE 771 Family
I/O Scanning	Available
Global Data	Available
Modbus Messaging	Available
FTP/TFTP	Available
SNMP	Available
HTTP Server	Available
DHCP	Unavailable

NOTE: Only the 140 NOE 771 01 or 140 NOE 771 11 (TCP/IP Ethernet modules) support a Modicon Quantum Hot Standby with Unity V2.0 system.

Hot Standby Topology

Hot Standby Interconnection

The following diagram shows a Hot Standby system and the relationship between the 2 redundant systems. Two 140 CPU 671 60 controllers are connected via a link created with fiber optic cable. The RIOs are connected both to each other (through the fiber optic cable) and to the RIO drops.



NOTE: The following three items are required.

1. Two identical systems
2. identical order of modules in each rack
3. identical software revisions

The NOEs are connected to the same switch. 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 to connect to the same switch:

- If a failure to communicate between the NOEs occurs, the time to swap increases.
- 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 a BOOTP server is not 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 gives you a connection to the NOE, and you can then 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 Module

The module can be configured to work in conjunction with the Modicon Quantum Hot Standby with Unity controller. Since the primary CPU and secondary controllers must have an identical configuration, the configured IP addresses will be the same. The module's IP address is either the configured IP address or the configured IP address + 1. The current local Hot Standby state determines the IP address.

In the offline state, the IP address is determined by whether or not the other controller is in transition to the primary CPU state.

This table shows the IP address assignments:

Hot Standby State	**IP Address**
Primary CPU	Configured IP address
Standby CPU	Configured IP address + 1
Transition from primary to offline	Configured IP address, if peer controller **does not** go to primary CPU
Transition from standby to offline	Configured IP address + 1

IP Address Restriction

Do not use either **broadcast IP address** or **broadcast IP address - 2** to configure the module.

IP Address Transparency

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

For a Quantum Hot Standby configuration:

- Do not use the address IP + 1.
- Do not use consecutive IP addresses of the configured IP address.
- Do not configure the primary CPU address as nnn.nnn.nnn.254. This causes the standby CPU IP address to be: nnn.nnn.nnn.255. The module would then return the diagnostic code **Bad IP configuration**.

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

When a switchover occurs, a new primary CPU PLC takes the IP address of the old primary CPU PLC. When the PLC that has stopped becomes operational again and rejoins the hot standby system, it takes the IP address of the secondary PLC. A new primary CPU NOE must have the same IP address as the former primary CPU NOE. The IP address in the secondary NOE (an NOE in the secondary state) is IP address + 1.

The NOEs integrated into the Modicon Quantum Hot Standby with Unity configuration coordinate this swapping IP address with the management of Ethernet services used.

NOE Operating Modes and Modicon Quantum Hot Standby with Unity

The NOE Modes

The NOE module modes are:

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

The Modicon Quantum Hot Standby with Unity and the NOE operating modes are synchronized by the conditions described in the following table:

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

Any 1 of 4 events will affect the NOE operating mode. These 4 events occur when:

- the NOE is powered up
- an NOE executes a Hot Standby switch over
- an NOE goes to offline mode
- a new application is downloaded to the NOE

Power Up and IP Address Assignment

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 CPU	configured IP address
secondary	configured IP address + 1
unassigned to offline	See the *Offline Mode at Power-up Sequence* in the next table.

If two NOEs power up simultaneously, a *resolution algorithm*:
- determines the primary CPU NOE
- assigns the configured IP address to that primary CPU NOE
- assigns the configured IP address + 1 to the secondary NOE

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

Power Up and Ethernet Services

The following table shows how the status of an NOE service is affected by the Modicon Quantum Hot Standby with Unity HSBY 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 CPU	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 table describes the manner in which NOEs coordinate the Hot Standby switchover.

Step	Action
1	NOE A is running in the primary CPU PLC and NOE B is in the secondary PLC in a hot standby configuration.
2	NOE A detects that its PLC has changed from primary CPU to offline mode.
3	NOE A changes its HSBY state from primary CPU to offline with the same Ethernet services running, and starts its watchdog timer (with 500 ms timeout setting). It waits for a UDP request to swap IP addresses from NOE B.
4	NOE B detects that its PLC has changed state from secondary to primary CPU.
5	NOE B stops all its Ethernet services, sends a UDP request to NOE A for the synchronization of the IP address swap, starts its watchdog timer (with 500 ms timeout setting), and waits for an UDP response from NOE A.
6	Once NOE A receives the UDP request from NOE B (or after its watchdog timer times out), it stops all its Ethernet services. If it has received a UDP request, NOE B sends a UDP response to NOE A; if its watchdog timer has timed out, NOE B does not send a UDP response. NOE A then swaps its IP address and starts secondary services.
7	NOE B swaps IP addresses and starts Ethernet services as primary CPU.
8	After NOE A senses that its local controller changes state from offline to standby, it takes the secondary IP address.
9	NOE B now becomes the primary CPU NOE.
10	NOE B opens all client connections and listens for all server connections and reestablishes those connections.
11	NOE A listens for all server connections and reestablishes those connections.

NOTE: During the Hot Standby switchover, there is a loss of communication during 500 ms between the PLC and the HMI and/or Unity Pro.

Going to Offline

When either the CPU stops or the Hot Standby state goes to offline mode, 2 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 CPU to offline	Configured IP address, if other controller **does not** go to primary CPU
Standby CPU to offline	Configured IP address + 1

Address Swap Times

Description

The following table details address swap times, such as the time to close connections, time to swap IP addresses, or time to establish connections.

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 swap times, please see the *Quantum NOE 771xx Ethernet Modules User Guide* (840 USE 116).	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 re-establishment connection	500 ms + the time of the client re-establishment connection
FTP/TFTP server	The time of the client re-establishment 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 re-establishment connection	500 ms + the time of the client re-establishment connection

Network Effects of Modicon Quantum Hot Standby with Unity Solution

Overview

The Modicon Quantum Hot Standby with Unity Pro 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 Modicon Quantum Hot Standby with Unity solution.

Browsers

NOTE: In Modicon Quantum Hot Standby with Unity Pro configuration, the NOE's I/O Scanner must be enabled.

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

Remote Clients

Hot Standby switchover affect remote clients.

An NOE will reset under the following conditions:

Remote Connection Request during Hot Standby : If a remote client establishes a TCP/IP connection during a Hot Standby switchover, the server closes the connection using a TCP/IP reset.

Hot Standby Switchover during Remote Connection Request : If a remote client makes a connection request and a Hot Standby switchover 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 switchover, the NOE will reset all client connections using a TCP/IP reset.

I/O Scanning Service

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION - DEVICES GO TO THEIR FALL-BACK STATES DURING SWITCHOVER

Configure Ethernet output devices to their Hold Last Value fallback state whenever possible. Output devices that support only a Set to Zero fallback state may produce a pulse during switchover.

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

I/O scanning provides the repetitive exchange of data with remote TCP/IP nodes I/O devices. While the PLC is running, the primary CPU NOE sends Modbus read/write, read, or write requests to remote I/O devices, and transfers data to and from the PLC memory. In the secondary controller, the I/O scanning service is stopped.

When the Hot Standby switchover occurs, the primary CPU NOE closes all connections with I/O devices by sending a TCP/IP reset. The I/O scanning service in this NOE is standby CPU.

After the switchover, the new primary CPU NOE re-establishes the connection with each I/O devices. It restarts the repetitive exchange of data with these re-connections.

The module provide the I/O scanning feature. Configure this feature with the Unity Pro software.

Using either method, the configuration and transfer of data between network addresses can be done without using the MSTR/IEC function block.

NOTE:

You must account for the following Ethernet I/O scanning considerations during a switchover.

- If MSTR/IEC function block is used for TCP/IP, only some of the opcode will be used. Therefore, the block does not complete its transaction, and it returns error code 0•8000.
- While the NOE is in the process of performing the transaction, a new MSTR/IEC function block may become active.
- The output states of the scanned I/Os will follow the state defined in the last value option configured in the I/O scanning table of the NOE module (in Unity Pro software).
  These 2 states are either:
  a. set to 0
  b. hold last

With the above considerations, we recommend using switchover with Ethernet I/O scanning for less critical applications.

Global Data (Publish/Subscribe) Service

The Hot Standby primary CPU NOE is 1 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 1 multicast address.

In this network service, the Modicon Quantum Hot Standby with Unity controllers are viewed like only 1 station. The primary CPU 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 switchover occurs, the primary CPU NOE stops the global data service. The NOE does not publish the local variable during a switchover. And after the switchover, the new primary CPU NOE starts to publish application variables and to receive the subscription variables.

FTP/TFTP Server

The 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. Modicon Quantum Hot Standby with Unity allows only 1 active FTP/TFTP client session per NOE module.

When the Hot Standby switchover occurs, the primary CPU and secondary NOEs close the FTP/TFTP connection. If you send an FTP/TFTP request during the switchover, 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

Overview

This chapter describes how to use the NOET 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

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 -- you must program an MSTR block in ladder logic. For details about the MSTR block, see *Transferring Data Using Communication Blocks, page 65*.

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, see *Ethernet Developers Guide, page 245*.

NOTE: In its capacity as server, the Ethernet module can accept only 32 connections at a 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 **OK**. **Result**: The Welcome dialog box appears.
4	Click **Next**, and follow the instructions that appear in each of the dialog boxes to complete the installation. **Note**: Each installation dialog box has **Back** and **Next** buttons that allow you to move back to the previous dialog box or forward to the next dialog box.

Establishing a Connection with an Ethernet Module

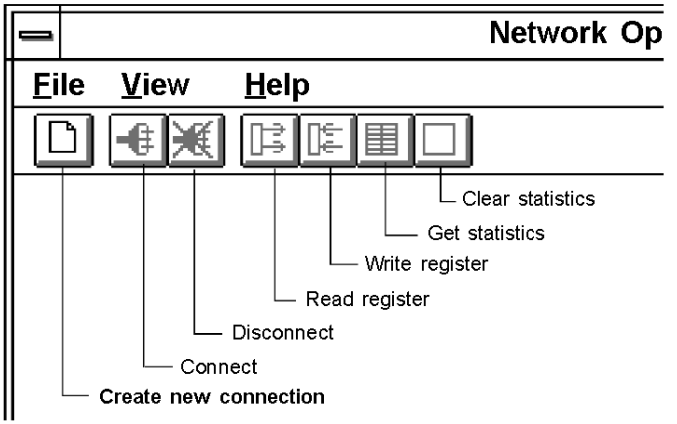
Overview

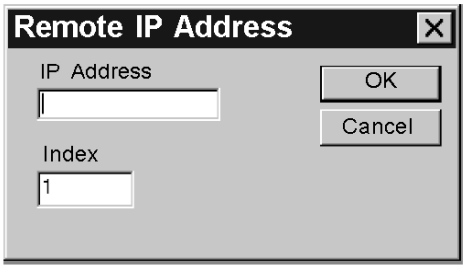
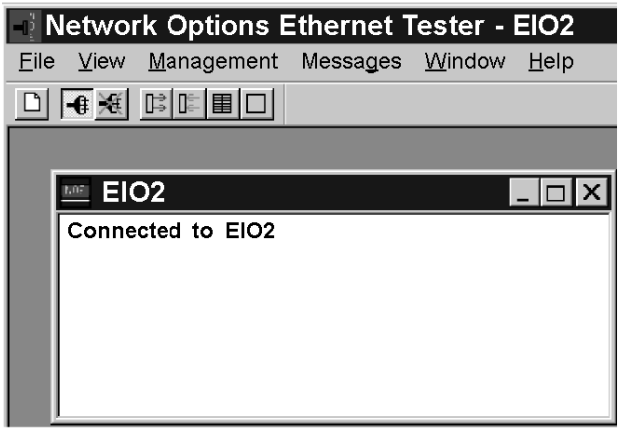
The following information describes how to use the NOET when connecting to an Ethernet adapter.

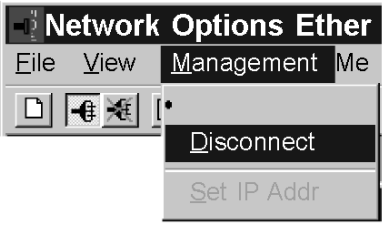
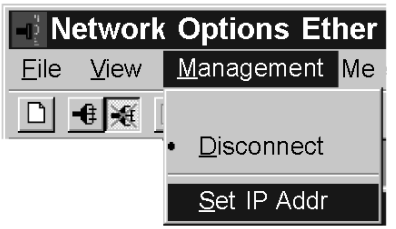
To establish a connection with an Ethernet module using the NOET, you must know the module's IP network address or role name.

Connecting with an Ethernet Module

Follow the steps below to establish a connection with an Ethernet module using the NOET.

Step	Action
1	Click **File** → **New**.  Or, click the **Create new connection** toolbar icon.  **Result**: The Remote IP Address dialog box appears.

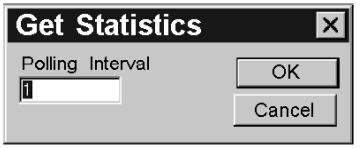
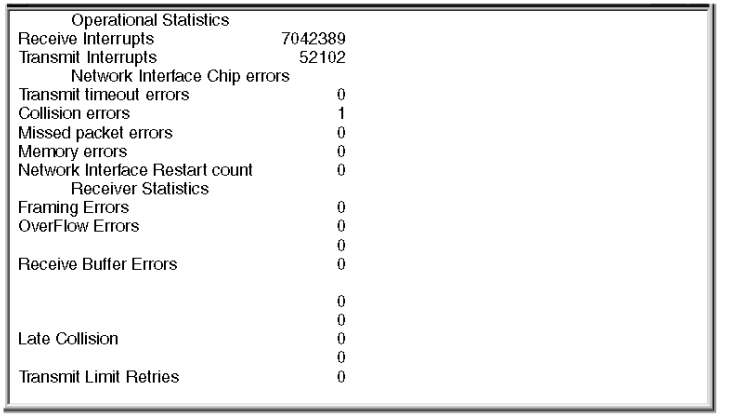
Step	Action
2	Type the adapter's IP network address or role name in the **IP Address** field.  **NOTE:** With a bridge, the IP of the bridge is entered and the bridge index is placed in the **Index** field. This valus is normally a Modbus or Modbus Plus address (depending on the type of bridge being used); in a CEV20030 or CEV20040, this could be the index number for a predefined routing path in the bridge. After clicking **Enter**, the tool will attempt to open a connection to the target IP. If this devices is present and accepts the connection, a dialog will appear that allows the user to define the offset and length of the Modbus message, Click **OK**. 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.

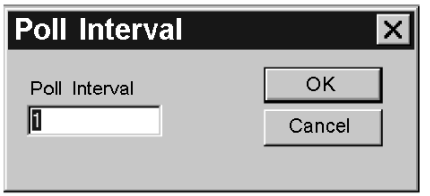
Step	Action
4	When you are ready to disconnect, click **Management** → **Disconnect**, or click the **Disconnect** toolbar icon. 
5	Click **Management** → **Set Ip Addr**, after disconnecting from 1 module, to reassign its dedicated connection. Type the new IP network address or role name in the box provided. 

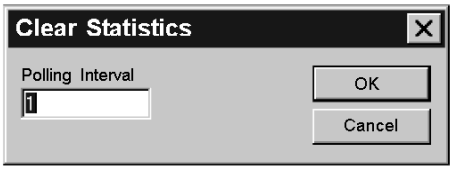
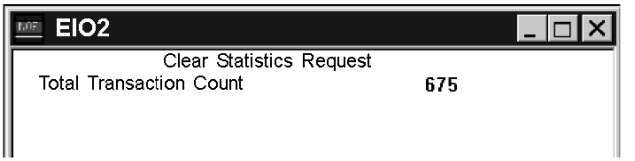
Getting and Clearing Statistics

Getting and Clearing Statistics

Follow the steps below to get statistics from the Ethernet module using your NOET.

Step	Action
1	Establish a connection with the Ethernet module.
2	• Click **Messages** → **Get Stats**. - or - • Click the **Get Statistics** toolbar icon. 
3	Enter the number of seconds between transactions in the **Polling Interval** field, and click **OK**. 
4	Complete statistics for the module will be printed in the window for this connection. 

Step	Action
5	Click **Messages** → **Poll Interval** to change the polling interval without interrupting communication with the Ethernet module. 
6	Enter the new number in the **Poll Interval** field, and click **OK**. 

Step	Action
7	• Click **Messages** → **Clear Stats** to clear statistics. - or - • Click the **Clear Statistics** toolbar icon. 
8	The following figure shows the Clear Statistics dialog box. Click **OK**.  **Result**: The Clear Statistics Request for the connection displays. 

Statistics

Overview

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 1,519 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 8 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.

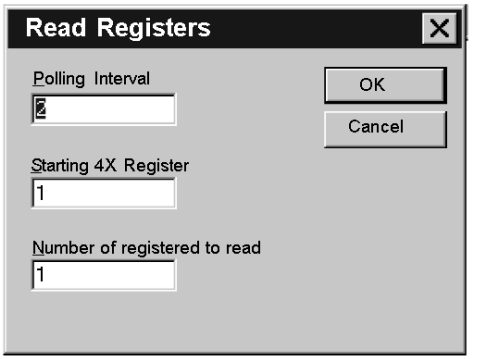
- **Transmit buffer errors**: The number of times the end packet flag on the current buffer was not set and the Ethernet controller did not own the next buffer. A transmit buffer error causes a restart.
- **Silo Underflow**: The number of times a packet was truncated due to data late from memory. A Silo 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* (840 USE 101) for details.

Reading Registers

Reading Registers

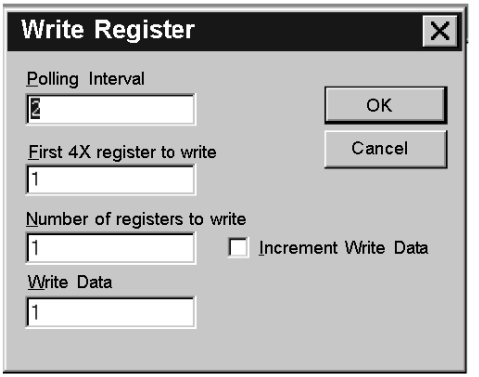
The following procedure describes how to read registers from the Ethernet adapter using your NOET.

Step	Action
1	Establish a connection with the Ethernet adapter.
2	• Click **Messages** → **Read Register**. - or - • Click the **Read Register** toolbar icon. 
3	In the **Polling Interval** field, type the number of seconds between transactions. 
4	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.
5	In the **Number of registers to read** field, type in the number of registers to read.
6	Click **OK**. **Result**: 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

Writing Registers

The following procedure describes how to write registers from the Ethernet module using your NOET.

Step	Action
1	Establish a connection with the Ethernet module.
2	• Click **Messages** → **Write Register**. - or - • Click the **Write Register** toolbar icon. 
3	In the **Polling Interval** field, type the number of seconds between transactions. 
4	In the **First 4x register to write** field, type the number of the first 4x register you want to write. When typing the 4x register number, omit the leading 40 or 400.
5	In the **Number of registers to write** field, type the number of registers to write.
6	In the **Write Data** field, type the data to be written to those registers.
7	Select the **Increment Write Data** check box if you want the data you have entered to increase by 1 with each transaction.
8	Click **OK**. **Result**: 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 NOET displays 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* (840 USE 101).

Using the Test Button

Overview

The NOET is an option that allows you to test data.

There are 3 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.

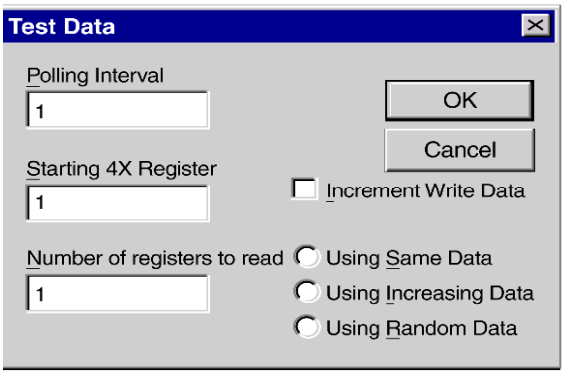
Test Function and Options

Click **Messages** → **Test Data** to access the test option.

- or -

Click the **Test** toolbar icon.

Result: The **Test Data** dialog box opens.



In the Test Data dialog box, type values in the following 3 fields.

- **Polling Interval**
- **Starting 4X Register**
- **Number of registers to read**

Select the **Increment Write Data** check box if you want the data you have entered to increase by 1 with each transaction.

If you select the **Increment Write Data** check box, you must choose 1 of 3 options for incrementing write data. Click the appropriate 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

Overview

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

Network management software allows a network manager to:
- monitor and control network components
- isolate problems and find their causes
- query 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 SNMP, which is the standard protocol used to manage a 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 that SNMP has access to must be defined and given a unique name. Both the manager and agent program must agree on the names and the meanings of the fetch and store operations. The set of all objects that SNMP can access is known as an 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 an SNMP Manager

If you have a SNMP manager already working, you may continue to use that SNMP manager. If you are selecting a SNMP manager, there are many on the market, and you may use any of these managers. You must use a SNMP, Version 1 compliant manager.

If you do not currently use a SNMP manager in your organization and are evaluating them for purchase, we recommend that you consider the HiVision, with the ConnexView Add-On, developed for use with Schneider Automation PLCs.

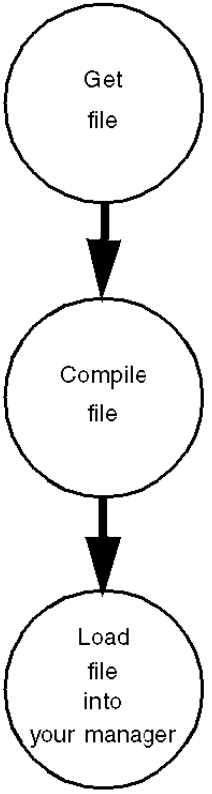
Please contact your Schneider Electric sales office *(see page 29)* for availability and pricing of HiVision and ConnexView.

Using an SNMP Manager

Follow the steps below to obtain a SNMP manager.

Step	Action
1	Get Schneider `.mib` file from the NOE Web page. You will find the `.mib` file as a packed file under **/wwwroot/SchneiderTFE.zip** on your NOE module.
2	Compile `.mib` file in the compiler.
3	Load the compiled `.mib` file to the SNMP 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. Search for the words `Network Management for Microsoft Networks Using SNMP`.
- Use an Internet search engine to search for an SNMP introduction, tutorial, or 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 FAQs. In addition, a search of most online retail book sites will yield a substantial list of titles.

ASN.1 Naming Scheme

Overview

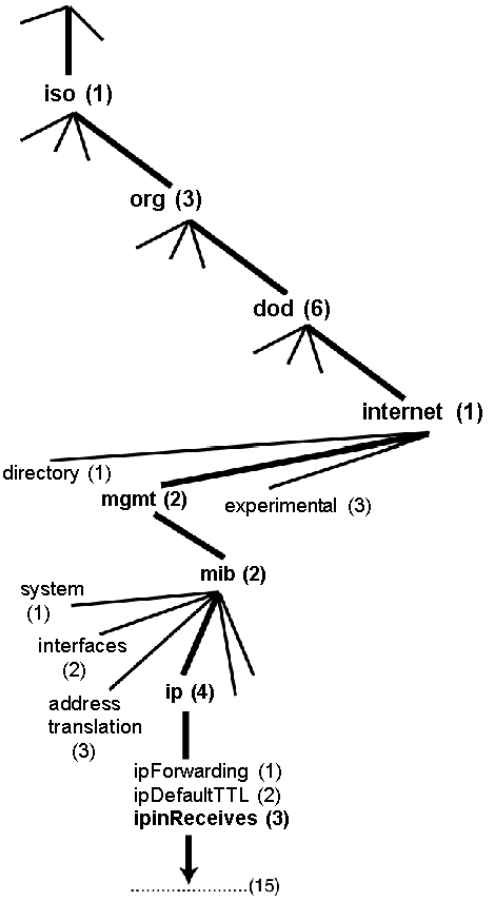
Abstract Syntax Notation One (ASN.1) is a formal language for abstractly describing messages to be exchanged between distributed computer systems.

Example

Objects in an MIB are defined with the ASN.1 naming scheme. The naming scheme assigns each object a long prefix, which guarantees that the name will be unique.

Example: An integer, which counts the number of IP datagrams that a device has received, is named *iso.org.dod.internet.mgmt.mib.ip.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
- 3 = identified organization — 1 of the branches under the ISO root
- 6 = U.S. Department of Defense (DOD) — 1 of the children under branch1.3
- 1 = the Internet subtree under 1.3.6
- 2 = the mgm branch — (1 of 7) 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 (1 of 11)
- 3 = ipinReceives — the MIB object

Configuring a NOE with SNMP

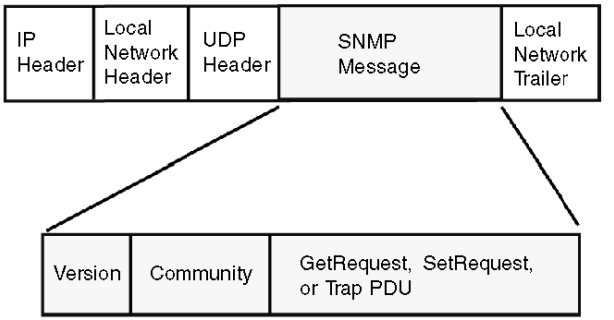
The Object Identifier (OID)

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.

The SNMP message is the innermost part of a typical network transmission frame, as shown below.



The PDUs within the SNMP initiate the communication between the manager and the agents.

The SNMP installed on your NOE module uses the following 3 PDUs.

- GetRequest
- SetRequest
- Trap

GetRequest PDU

The GetRequest (shortened to Get) PDU is used by the SNMP manager to retrieve the value of 1 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 1 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, which the manager and agent use. 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 2 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, reboot the module using hot swap to set the changes.

Configuring an NOE with TFE Private MIB

Introduction

An MIB is an element used in network management. Network management services are based on the need to monitor and 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 *(see page 217)* and SETs *(see page 217)* to retrieve system information and set system environment variables.

NOTE: The TFE private MIB is available only in the 140 NOE 771 -01, -11 and -21, 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 PEN from the 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:

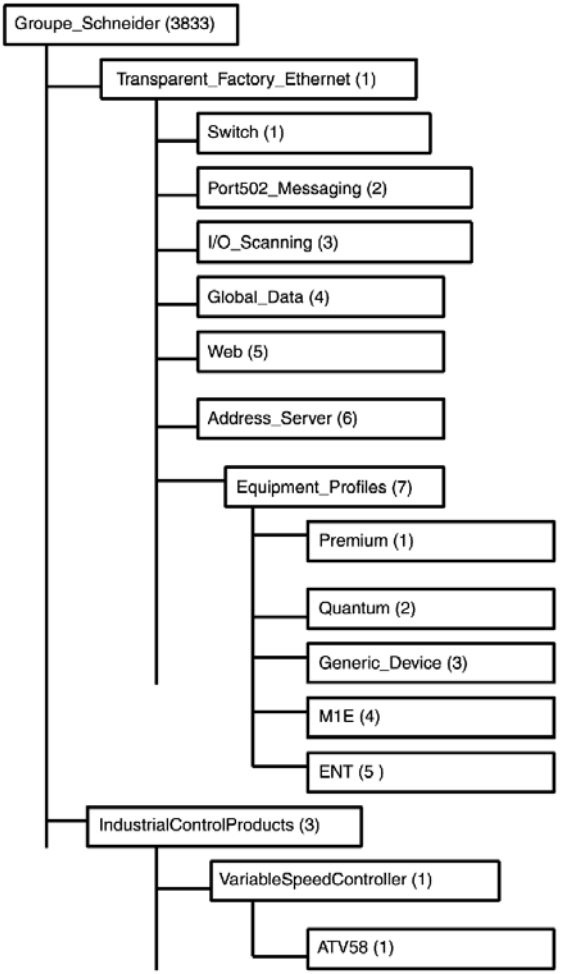
```
iso(1)
        org(3)
                dod(6)
                        internet(1)
                                private(4)
                                        enterprise(1)
                                                GroupeSchneider(3833)
                                                        Transparent_Factory_Ethernet(1)
```

Under the Groupe Schneider 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 enterprise 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 IndustrialControlProducts (3) subtree)

When a new device is created, a corresponding object description is created in the ASN.1 format. The ASN.1 files 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, X-Way)
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 1 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

Service	Description
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 1 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 address server 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
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 0.
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. **Example**: The PLC Premium family is 100.

Service	Description
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 4 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.
- send messages to the 2 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 Ethernet Private 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 an `.mib` text file.

Maintenance

12

Overview

This chapter details information about system maintenance, including accessing and clearing the crash log and downloading the new NOE exec.

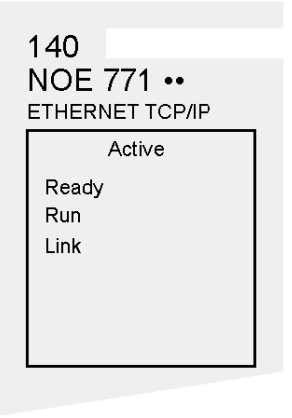
What's in this Chapter?

This chapter contains the following topics:

Responding to Errors

Detecting Errors

When faults occur, the NOE 771 •• 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 on the following pages in the order noted. If no remedy suggested in this manual overcomes the error, call your local service representative or contact Schneider Electric customer service *(see page 29)*.

Certain error codes are recorded in the MSTR block. For instructions about how to read and interpret these codes through ProWORX NxT, Modsoft, or Concept, please refer to *MSTR Function Error Codes, page 71*.

Responding to an Active LED Error Indicator

If the Active LED fails to light, the NOE 771 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 working; 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 2 network option modules (including NOE, NOM, NOP, and CRP 811 modules) have been installed in the backplane with a 140 CPU 113 or 213 (not more than 6 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 later to support the Ethernet module. Earlier versions do not recognize the module.
6	If steps 4 and 5 above check out okay, replace the NOE 771 module.

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 okay, replace the NOE 771 module.

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 okay, replace the NOE 771 module.

Kernel LED Error

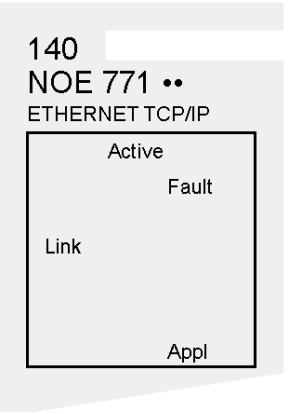
The following table describes the Kernel LED errors that may occur and how to respond to them.

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, page 197*).

Fault LED

The Fault LED will flash briefly, following an error as the module attempts to recover.

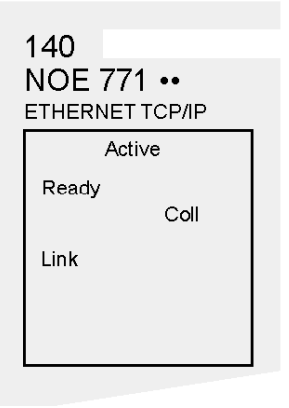
The following figure shows the Fault LED.



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

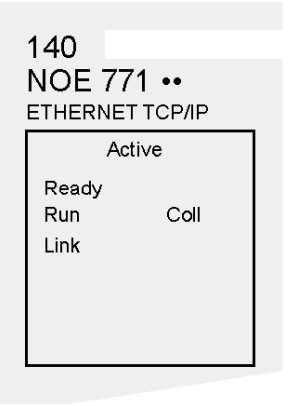
If the Collision LED fails to light, use the following procedure.

Step	Action
1	Make sure that the cable has been installed and is working properly.
2	Verify that the Ethernet hub/switch is working 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, we recommend you segment 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.

Number of Flashes in Sequence	Action
3	Check Ethernet connection.
4	Change IP address.
5	Provide IP address.
6	Connect using default IP address and configure.
7	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 *(see page 233)*.

Reading and Clearing the Crash Log

Overview

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 *(see page 29)*, 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 advanced diagnostics. The crash log is one of the tools used to solve complex problems.

If the Appl indicator is on, 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 page 143)* or through FTP.

Follow the steps below to access the crash log through 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 file: `get crash.log`

Clearing the Crash Log

The crash log can be cleared from the Embedded Web pages *(see page 143)* or through FTP.

Follow the steps below to access the crash log through FTP.

Step	Action
1	Log into the module's FTP server.
2	Change the directory to `wwwroot/conf/diag`.
3	Perform an FTP to delete the crash log file: `rm crash.log`

Downloading a New NOE Exec

Introduction

Use the following tools to download a new NOE Exec.

- Schneider Automation programming packages (see corresponding manuals)
- FTP

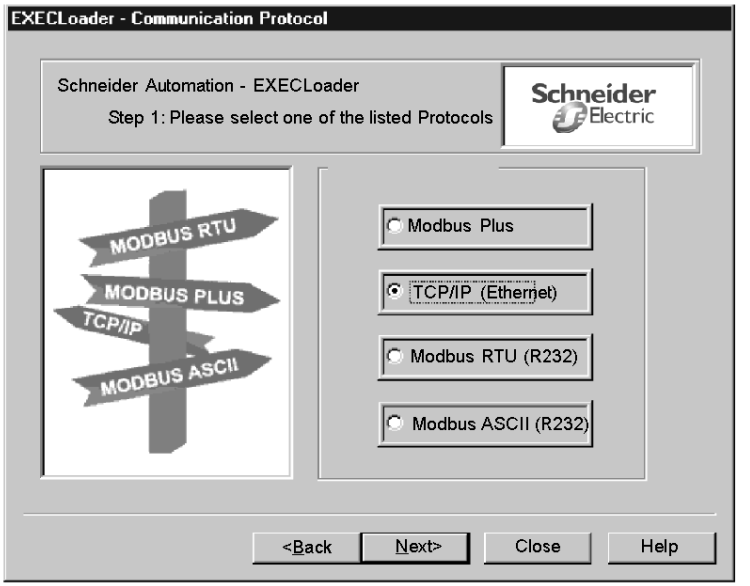
The Concept EXECLoader

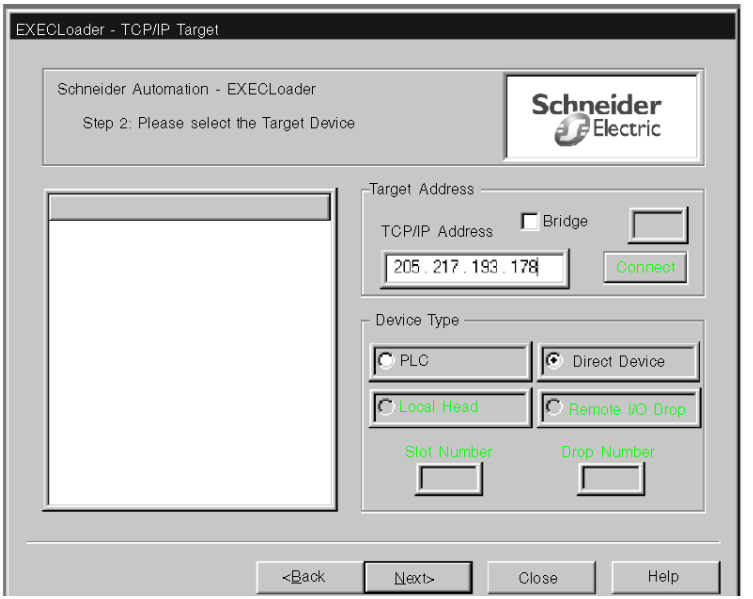
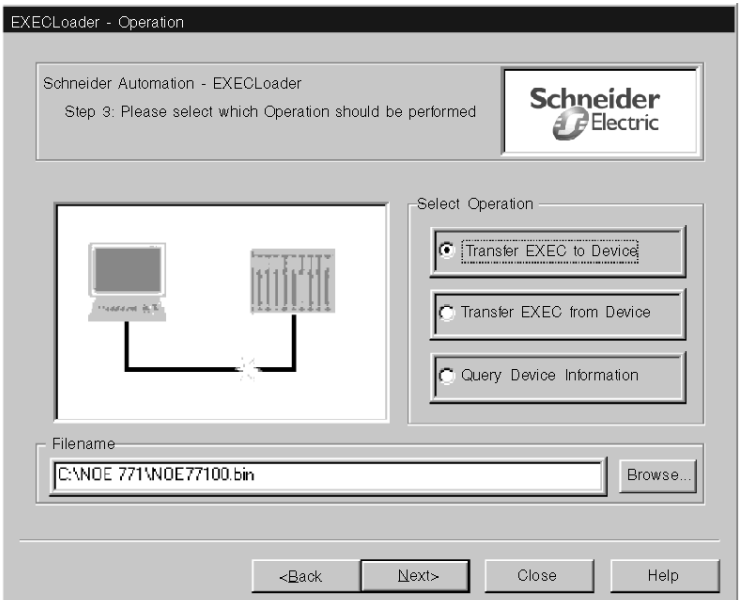
Overview

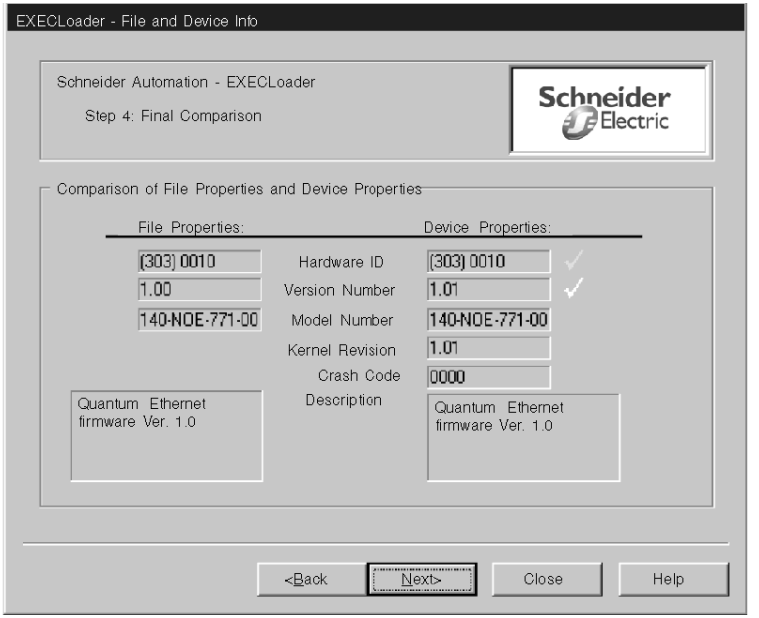
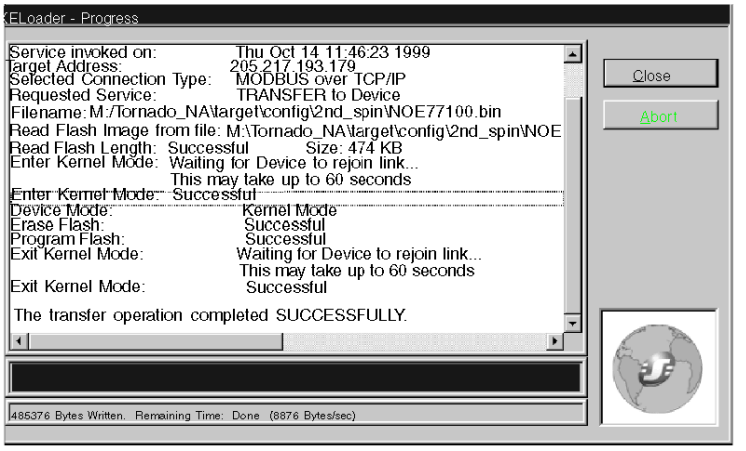
This section describes the use of the Concept EXECLoader to download a new NOE Exec.

Downloading NOE Exec

Follow the steps below to download a new NOE Exec using the Concept EXECLoader.

Step	Action
1	Activate the Exec Loader program.
2	Click **Next** to start the Exec Loader process.
3	Click the **TCP/IP (Ethernet)** button, and click **Next.** 

Step	Action
4	Click the **Direct Device** button, and click **Next**. 
5	Click the **Transfer EXEC to Device** button. 

Step	Action
6	Click **Browse** to select the file name, and click **Next**.
7	The EXECLoader - File and Device Info dialog box opens. Click **Next**. 
8	The EXECLoader - Progress dialog box opens. 
9	When the process is complete, click **Close**.

Downloading a New NOE Exec using FTP

Exec Version

Please check the current NOE Exec file version on the NOE Properties Web page.

Follow these links: **Web Server** → **Diagnostics and Online Configuration** → **NOE Properties**.

Do not change a new version of the NOE Exec file.

Procedure

⚠ CAUTION

POTENTIAL SYSTEM CRASH

Be sure you reboot the module after downloading a new NOE Exec using FTP. An Exec update using FTP may cause a crash to the system.

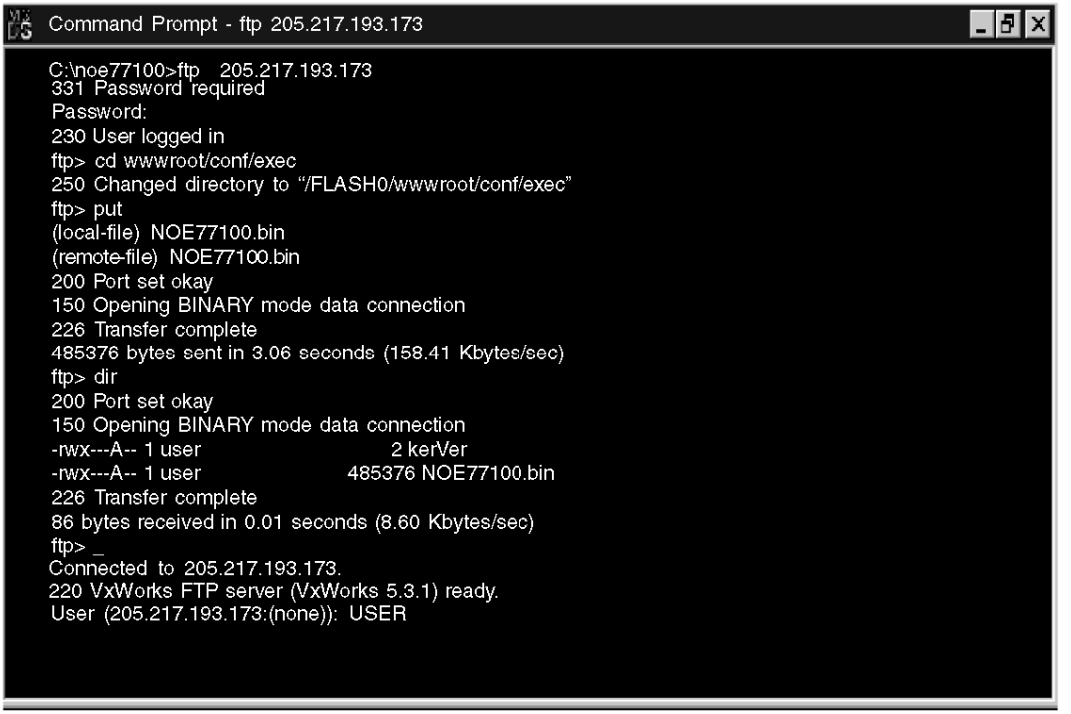
Failure to follow these instructions can result in injury or equipment damage.

Follow the steps below to download a new NOE Exec using 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**. **Note**: Pay attention that the NOE771xx.bin is the local path on your PC (default path: `c:\`).
6	At the local file prompt, type `NOE771xx.bin`, and press **Enter**.
7	At the remote file prompt, type `NOE771xx.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 extension in lowercase as shown in the figure below. **Example**: `NOE771xx.bin`

Sample FTP Session

The following FTP session was used to download an NOE Exec.



NOTE: The NOE Kernel can not be downloaded using FTP.

Downloading a New NOE Kernel

Overview

Version 2.00 of the NOE Exec adds a new feature to allow the updating of the low level kernel within the NOE 771 •• firmware. Follow the procedure below to ensure the proper installation of new kernel firmware.

NOTE: The NOE kernel can not be downloaded using FTP.

⚠ CAUTION
NOE OPERATION You must follow the kernel download procedure below. Failure to do so will render the NOE inoperable. **Failure to follow these instructions can result in injury or equipment damage.**

Kernel Version

Please check the current NOE kernel version on the NOE Properties Web page.

Follow these links: **Web Server** → **Diagnostics and Online Configuration** → **NOE Properties**.

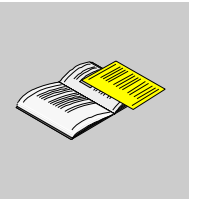
Do not change a new version of the NOE kernel.

Procedure

Follow these steps to download a NOE kernel.

Step	Action
1	Check the current version of the NOE's Exec firmware.
2	If the Exec is not at 2.00 or later, you must update the Exec first. After loading the new Exec and before loading the kernel, make sure to cycle power to the NOE.
3	Loading the kernel firmware is performed using the EXECLoader in the same manner as the Exec firmware.
4	After the transfer is 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



Overview

The appendices provide supplementary reference information for the Quantum NOE 771 •• series of modules.

What's in this Appendix?

The appendix contains the following chapters:

Specifications

A

Specifications

Specification Table

Communication ports	One auto-sensing 10/100Base-T shielded twisted pair (RJ-45 connector) port and 1 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 later
	Modlink, Ver. 2.0, or later
	Modsoft, Ver. 2.6, or later
	ProWORX NxT, Ver. 2.1, or later
Firmware	
CPU type and version	Quantum executive, Ver. 2.0, or later
NOE upgradeable	Field upgradeable using FTP or programming panel
Operating Conditions	
Temperature	0 to +60° C
Humidity	0 to 95% Rh non condensing @ 60° C
Altitude	15,000 ft (4,500 m)
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

Overview

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 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

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 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 CSample_View class manages the TCP/IP connection, sends requests to remote nodes, and displays either the 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 through its application programming interface and through 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 the 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 following socket attributes.

- 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.

- **Invoke Identifier**: This 2-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 2-byte field identifies the protocol type. Currently, the only protocol supported is Modbus.
- **Command Length**: This 2-byte field is the size of the rest of the message.
- **Destination Identifier**: This 1-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

Timers

CSample_View periodically receives 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 CMainFrame's 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 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 you select 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, 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 page 254)* 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 CMainFrame 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 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 0.

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, 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, the header has been received. OnTcpIpRead sets the number of bytes received to 0, 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, OnTcpIpRead changes the receive state to RX_BODY. However, if either transaction identifier does not match or the protocol is not MODBUS, 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, 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, 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, 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, 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, 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

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**: Displays the results of a get statistics request.
- **ClearStatsPaint**: Displays the results of a clear statistics request.
- **ReadRegPaint**: Displays the results of a read register request.
- **WriteRegPaint**: 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

Overview

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 MBAP is a layer-7 protocol providing peer-to-peer communication between 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 MBAP can be transmitted over any communication system that supports messaging services. However, the current Quantum implementation transports MBAP 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 MBAP 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 7 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 2 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 MBAP, called mb_pdu, contains 2 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

Overview

There are several classes of service that are part of the MBAP. 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 you to define parameter values which affect the PLC's register files, I/O map, communication port configuration and scan attributes, to name a few.

Device Execution State Control

This service class allows you to start/stop the PLC scan execution. You must be in an application login context, which is obtained through other Modbus services.

Modbus Application Protocol PDU Analysis

Analysis

The MBAP 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 MBAP whose messages are received at a well-known port. The current maximum size of the mbap_pdu for this class of services is 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
    services    len               [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 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.

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 MBAP does not support either broadcast or multicast at the application layer.

Schneider Electric's Quantum PLCs use broadcast addressing because they use ARP to locate the destination node. The client interface to the MBAP service on the PLC, the MSTR block, requires you to provide the destination IP address. Also the embedded stack uses 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 MBAP.

Reference Documents

Overview

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

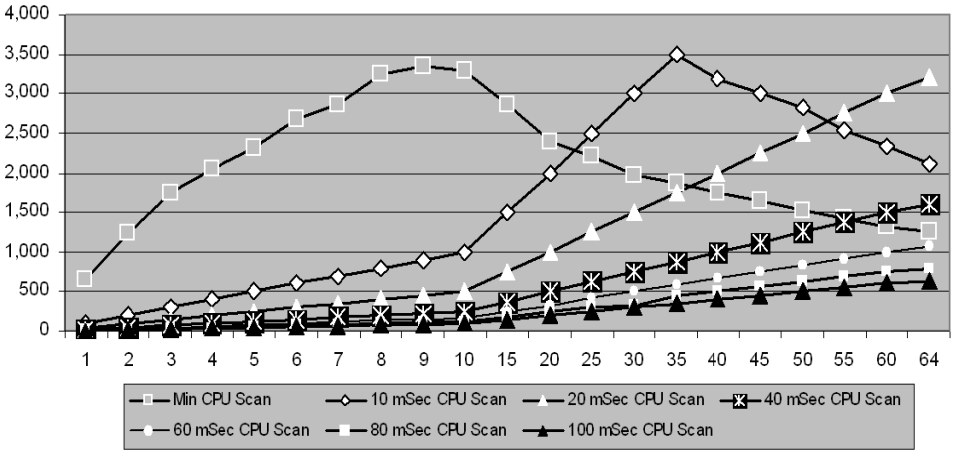
140 NOE 771 -00, -01, and -11 I/O Scanner Performance

Overview

The following information describes the performance of the 140 NOE 771 -00, -01, and -11 I/O scanner with various Quantum CPUs.

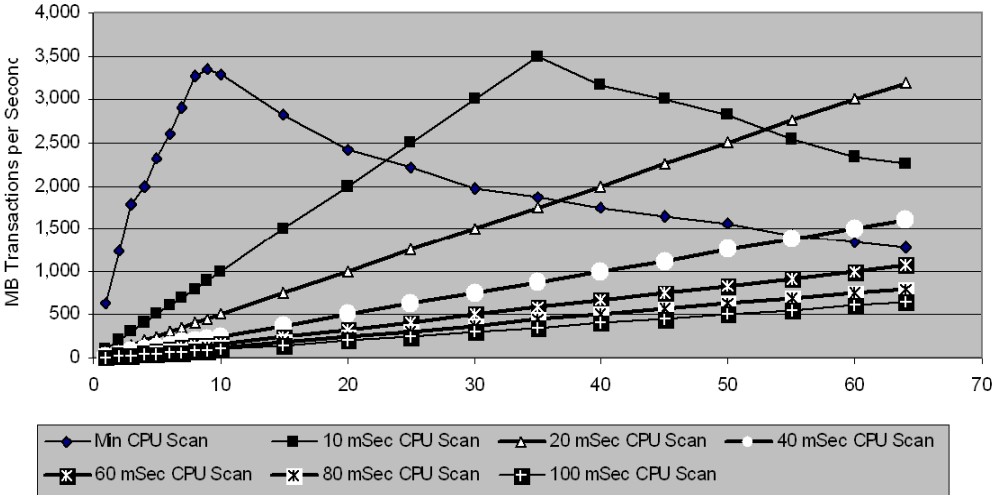
Quantum 113 CPU

The following figure shows the performance of the 140 NOE 771 -00, -01, and -11 I/O scanner with a Quantum 113 CPU.



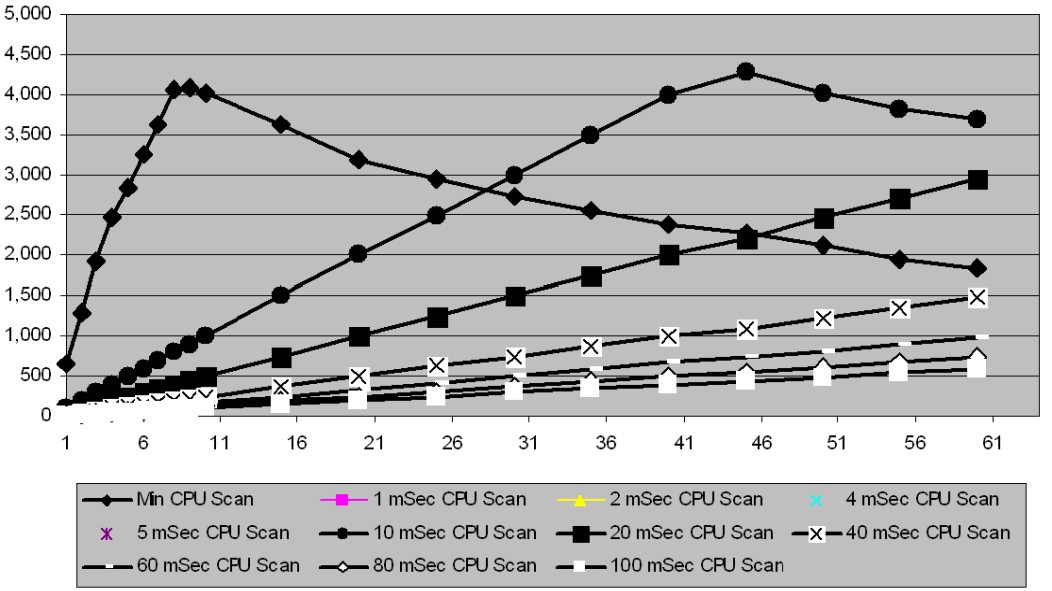
Quantum 213 CPU

The following figure shows the performance of the 140 NOE 771 -00, -01, and -11 I/O scanner with a Quantum 213 CPU.



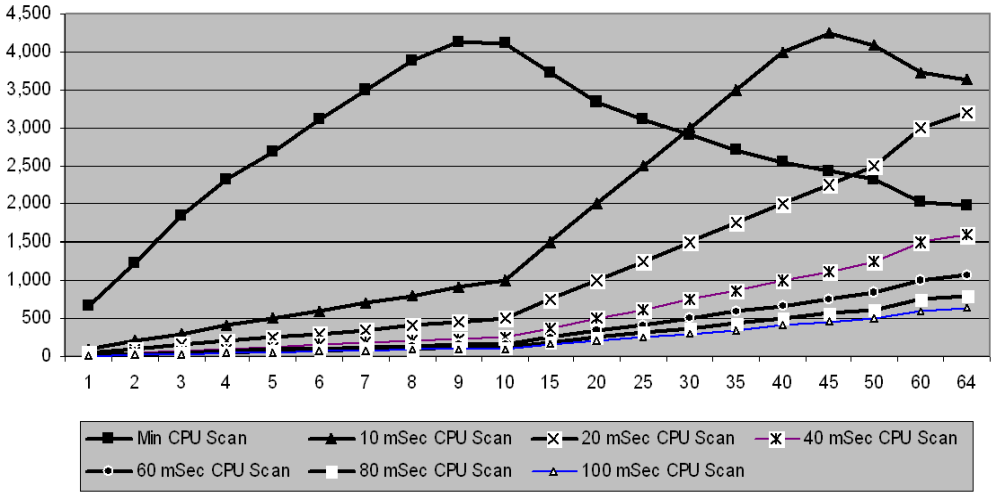
Quantum 424 CPU

The following figure shows the performance of the 140 NOE 771 -00, -01, and -11 I/O scanner with a Quantum 424 CPU.

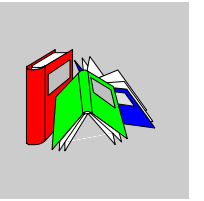


Quantum 534 CPU

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 •• 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 Sserver

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.

FDR

Faulty Device Replacement. The method of handling device replacement without disrupting the system or interrupting service.

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.

IANA

Internet Assigned Numbers Authority

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.

ISO

International Organization for Standardization

L

LAN

Local Area Network

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. A layer-7 protocol providing peer-to-peer communication between PLCs and other host-based nodes on a LAN.

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.

PDU

Protocol Data Unit

peer cop

Software that allows you to configure data blocks to be transferred between controllers on a Modbus Plus network.

PEN

Private Enterprise Number

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

role name

Within Faulty Device Replacement, a role name is a logical name that the user assigns to a device, a logical name that has a meaning within the application.

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).

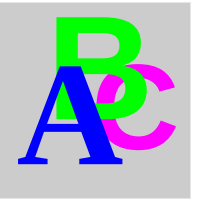
Winsock

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



Symbols

0-9

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

W