




ProLinx[®]
MNETC
ProLinx Gateway
Modbus TCP/IP Client

May 11, 2011

Your Feedback Please

We always want you to feel that you made the right decision to use our products. If you have suggestions, comments, compliments or complaints about our products, documentation, or support, please write or call us.

How to Contact Us

ProSoft Technology

5201 Truxtun Ave., 3rd Floor

Bakersfield, CA 93309

+1 (661) 716-5100

+1 (661) 716-5101 (Fax)

www.prosoft-technology.com

support@prosoft-technology.com

Copyright © 2011 ProSoft Technology, Inc., all rights reserved.

MNETC Driver Manual

May 11, 2011

ProSoft Technology[®], ProLinx[®], inRAx[®], ProTalk[®], and RadioLinx[®] are Registered Trademarks of ProSoft Technology, Inc. All other brand or product names are or may be trademarks of, and are used to identify products and services of, their respective owners.

ProSoft Technology[®] Product Documentation

In an effort to conserve paper, ProSoft Technology no longer includes printed manuals with our product shipments. User Manuals, Datasheets, Sample Ladder Files, and Configuration Files are provided on the enclosed CD-ROM in Adobe[®] Acrobat Reader file format (.PDFs). These product documentation files may also be freely downloaded from our web site: www.prosoft-technology.com

Important Installation Instructions

Power, Input, and Output (I/O) wiring must be in accordance with Class I, Division 2 wiring methods, Article 501-4 (b) of the National Electrical Code, NFPA 70 for installation in the U.S., or as specified in Section 18-1J2 of the Canadian Electrical Code for installations in Canada, and in accordance with the authority having jurisdiction. The following warnings must be heeded:

WARNING - EXPLOSION HAZARD - SUBSTITUTION OF COMPONENTS MAY IMPAIR SUITABILITY FOR CLASS I, DIV. 2;

WARNING - EXPLOSION HAZARD - WHEN IN HAZARDOUS LOCATIONS, TURN OFF POWER BEFORE REPLACING OR WIRING MODULES

WARNING - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NON-HAZARDOUS.

THIS DEVICE SHALL BE POWERED BY CLASS 2 OUTPUTS ONLY.

ProLinx® Products Warnings

WARNING – EXPLOSION HAZARD – DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NON-HAZARDOUS.

AVERTISSEMENT – RISQUE D'EXPLOSION – AVANT DE DÉCONNECTER L'EQUIPMENT, COUPER LE COURANT OU S'ASSURER QUE L'EMPLACEMENT EST DÉSIGNÉ NON DANGEREUX.

ProLinx Gateways with Ethernet Ports

Series C ProLinx™ Gateways with Ethernet ports do **NOT** include the HTML Web Server. The HTML Web Server must be ordered as an option. This option requires a factory-installed hardware addition. The HTML Web Server now supports:

- 8 MB file storage for HTML files and associated graphics files (previously limited to 384K)
- 32K maximum HTML page size (previously limited to 16K)

To upgrade a previously purchased Series C model

Contact your ProSoft Technology distributor to order the upgrade and obtain a Returned Merchandise Authorization (RMA) to return the unit to ProSoft Technology.

To order a ProLinx Plus gateway with the -WEB option

Add **-WEB** to the standard ProLinx part number. For example, **5201-MNET-MCM-WEB**.

Markings

Label Markings

<cULus>

E183151

Class I Div 2

Groups A,B,C,D T6

-30°C ≤ Ta ≤ 60°C

<Ex>

II 3 G

EEx nA IIC T6

-20°C ≤ Ta ≤ 50°C

II – Equipment intended for above ground use (not for use in mines).

3 – Category 3 equipment, investigated for normal operation only.

G – Equipment protected against explosive gasses.

Agency Approvals and Certifications

Contents

Your Feedback Please.....	2
How to Contact Us.....	2
ProSoft Technology® Product Documentation.....	2
Important Installation Instructions.....	3
ProLinX® Products Warnings.....	3
ProLinX Gateways with Ethernet Ports.....	3
To upgrade a previously purchased Series C model.....	3
To order a ProLinX Plus gateway with the -WEB option.....	3
Markings.....	4
1 Functional Overview	9
1.1 Modbus TCP/IP (MNET) Port.....	10
1.2 Gateway Internal Database.....	11
1.2.1 Modbus TCP/IP Client Access to Database.....	11
2 Protocol Functional Specifications	13
2.1 Modbus TCP/IP Specifications.....	13
3 Modbus TCP/IP Client Protocol Configuration	15
3.1 Installing ProSoft Configuration Builder Software.....	15
3.1.1 Using the Online Help.....	16
3.2 Configure the Gateway.....	17
3.2.1 Configuring Module Parameters.....	17
3.2.2 Printing a Configuration File.....	17
3.3 MNET Servers.....	18
3.3.1 Float Flag.....	18
3.3.2 Float Start.....	18
3.3.3 Float Offset.....	18
3.3.4 Output Offset.....	19
3.3.5 Bit Input Offset.....	19
3.3.6 Holding Register Offset.....	19
3.3.7 Word Input Offset.....	19
3.4 [MNET CLIENT x].....	20
3.4.1 ARP Timeout.....	20
3.4.2 MBAP Port Override.....	20
3.4.3 Minimum Command Delay.....	21
3.4.4 Response Timeout.....	21
3.4.5 Retry Count.....	21
3.4.6 Float Flag.....	21
3.4.7 Float Start.....	21
3.4.8 Float Offset.....	21
3.5 MNET Client x Commands.....	22
3.5.1 Command List Overview.....	22
3.5.2 Commands Supported by the Module.....	23
3.5.3 Command Entry Formats.....	24
3.5.4 Enable.....	25

3.5.5	Internal Address.....	25
3.5.6	Poll Interval	25
3.5.7	Reg Count.....	26
3.5.8	Swap Code	26
3.5.9	Node IP Address.....	26
3.5.10	Service Port	27
3.5.11	Slave Address.....	27
3.5.12	Modbus Function	27
3.5.13	MB Address in Device	28
3.6	Ethernet Configuration.....	29
4	Client Error/Status Data	31
4.1	MNET Error and Status Data.....	31
4.1.1	MNET Server (Port 2000)	31
4.1.2	MNET Server (Port 502)	32
4.2	MNET Client Error/Status Data.....	33
4.2.1	MNET Client [0].....	33
4.2.2	MNET Client [1].....	33
4.2.3	MNET Client [2].....	34
4.2.4	MNET Client [3].....	34
4.2.5	MNET Client [4].....	34
4.2.6	MNET Client [5].....	35
4.2.7	MNET Client [6].....	35
4.2.8	MNET Client [7].....	35
4.2.9	MNET Client [8].....	36
4.2.10	MNET Client [9].....	36
4.3	Client Command Errors	37
4.3.1	Standard Modbus Exception Code Errors	37
4.3.2	Module Communication Error Codes	38
4.3.3	MNET Client Specific Errors.....	38
4.3.4	Command List Entry Errors	38
4.4	Ethernet LED Indicators.....	39
5	Modbus Protocol Specification	41
5.1	Read Coil Status (Function Code 01).....	42
5.2	Read Input Status (Function Code 02)	43
5.3	Read Holding Registers (Function Code 03).....	44
5.4	Read Input Registers (Function Code 04)	45
5.5	Force Single Coil (Function Code 05)	46
5.6	Preset Single Register (Function Code 06)	47
5.7	Diagnostics (Function Code 08)	48
5.7.1	Sub-function Codes Supported.....	48
5.7.2	Modbus Exception Responses	53
5.8	Force Multiple Coils (Function Code 15)	56
5.9	Preset Multiple Registers (Function Code 16).....	57
6	Support, Service & Warranty	59
	Contacting Technical Support	59
6.1	Return Material Authorization (RMA) Policies and Conditions.....	61
6.1.1	Returning Any Product.....	61

6.1.2	Returning Units Under Warranty	62
6.1.3	Returning Units Out of Warranty	62
6.2	LIMITED WARRANTY	63
6.2.1	What Is Covered By This Warranty	63
6.2.2	What Is Not Covered By This Warranty	64
6.2.3	Disclaimer Regarding High Risk Activities	64
6.2.4	Intellectual Property Indemnity	65
6.2.5	Disclaimer of all Other Warranties	65
6.2.6	Limitation of Remedies **	66
6.2.7	Time Limit for Bringing Suit	66
6.2.8	No Other Warranties	66
6.2.9	Allocation of Risks	66
6.2.10	Controlling Law and Severability	66

Index

67

1 Functional Overview

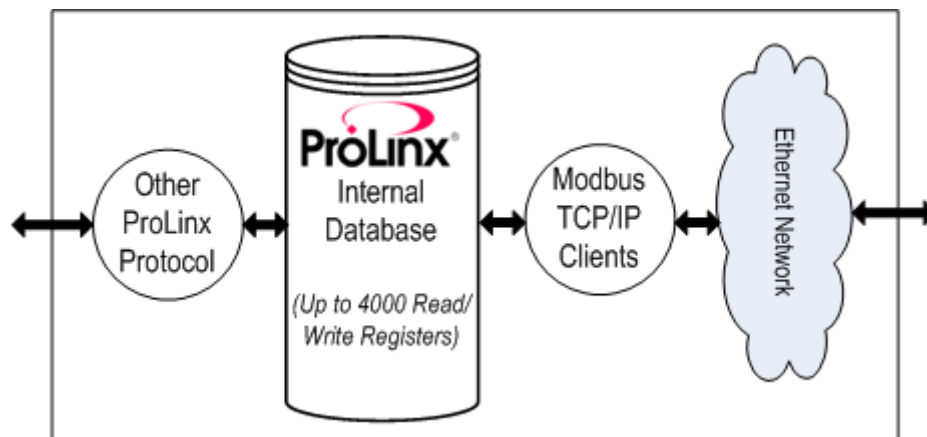
In This Chapter

- ❖ Modbus TCP/IP (MNET) Port..... 10
- ❖ Gateway Internal Database..... 11

The ProLinx Modbus TCP/IP (MNETC) Client driver can be used to interface many different protocols into the Schneider Electric Quantum family of processors as well other software solutions supporting the protocol. The MNET driver supports

- 5 MNET Modbus TCP/IP servers (port 2000)
- 5 MBAP Modbus TCP/IP servers (port 502)
- 10 Modbus TCP/IP clients (MNET or MBAP), each supporting up to 16 Modbus TCP/IP commands.

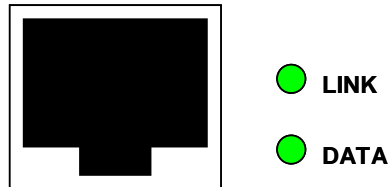
The Ethernet driver interfaces with a common internal database in the module. This permits the sharing of data across many different protocols and networks. The following illustration shows the functionality of the MNET driver.



1.1 Modbus TCP/IP (MNET) Port

The module supports up to ten Client connections on the TCP/IP network to interface with processors (and other server-based devices) using a user-constructed command list of up to 16 entries. The module's internal database is used as the source for write commands to the remote processors. Data collected from the processors using read commands is placed in the module's database.

Data in the module's internal database is accessible for read and write operations by any node on the network supporting the MBAP (Service Port 502) or MNET (Service Port 2000) TCP/IP protocols. The MBAP protocol is a standard Ethernet implementation defined by Schneider Automation and used on their Quantum Processor. This open protocol is a modified version of the serial Modbus protocol. The MNET protocol embeds a Modbus RTU serial protocol message in a TCP/IP packet. The module supports up to ten active Client connections with up to sixteen commands for each client.



1.2 Gateway Internal Database

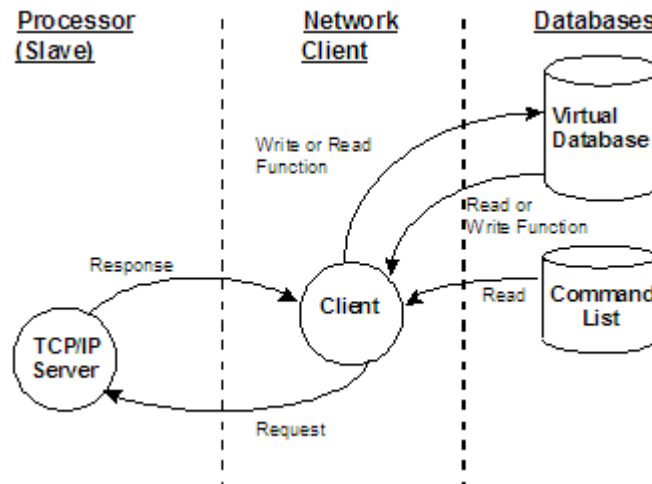
The internal database is central to the functionality of the gateway. This database is shared between all the ports on the gateway and is used as a conduit to pass information from devices on one network to devices on either connected network. This permits data from devices on one communication port or network to be viewed and controlled by devices on another port or network.

In addition to data from the Client and server ports, status and error information generated by the gateway can also be mapped into the internal database.

1.2.1 Modbus TCP/IP Client Access to Database

The client functionality exchanges data between MNETC module's internal database and data tables established in one or more Quantum processors or other server based devices. The command list, defined in the user configuration, determines what data is to be transferred between the module and each of the servers on the network. No ladder logic is required in the processor (server) for Client functionality, except to assure that sufficient data memory exists.

The following illustration describes the flow of data between the Ethernet Clients and the internal database.



2 Protocol Functional Specifications

In This Chapter

❖ Modbus TCP/IP Specifications 13

2.1 Modbus TCP/IP Specifications

Specification	Description
Modbus TCP/IP Client	
General	The client driver supports the active reading and writing of data with Modbus TCP/IP compatible devices. Ten client connections available (connect up to 10 servers/devices with 16 commands)
Configurable parameters	Number of active commands (Up to 16 for client) Min Command Delay Response Timeout Retry Count
Modbus Commands supported	1: Read Output Status 2: Read Input Status 3: Read Multiple Data Registers 4: Read Input Registers 5: Write Single Bit 6: Write Single Data Register 15: Write Multiple Bits 16: Write Multiple Data Register
Command List	Up to 16 commands on the Client port, each fully configurable for function, Server IP address, Service Port, register to/from addressing and word/bit count and processor slot number
Status Data	Error codes available on an individual command basis.
Polling of command list	User configurable polling of commands, including disabled, continuous and on change of data (write only)

3 Modbus TCP/IP Client Protocol Configuration

In This Chapter

❖ Installing ProSoft Configuration Builder Software.....	15
❖ Configure the Gateway.....	17
❖ MNET Servers.....	18
❖ [MNET CLIENT x].....	20
❖ MNET Client x Commands.....	22
❖ Ethernet Configuration	29

3.1 Installing ProSoft Configuration Builder Software

You must install the *ProSoft Configuration Builder (PCB)* software to configure the gateway. You can always get the newest version of *ProSoft Configuration Builder* from the ProSoft Technology website.

To install ProSoft Configuration Builder from the ProSoft Technology website

- 1 Open your web browser and navigate to *http://www.prosoft-technology.com/pcb*
- 2 Click the **DOWNLOAD HERE** link to download the latest version of *ProSoft Configuration Builder*.
- 3 Choose **SAVE** or **SAVE FILE** when prompted.
- 4 Save the file to your *Windows Desktop*, so that you can find it easily when you have finished downloading.
- 5 When the download is complete, locate and open the file, and then follow the instructions on your screen to install the program.

If you do not have access to the Internet, you can install *ProSoft Configuration Builder* from the *ProSoft Solutions Product CD-ROM*, included in the package with your gateway.

To install ProSoft Configuration Builder from the Product CD-ROM

- 1 Insert the *ProSoft Solutions Product CD-ROM* into the CD-ROM drive of your PC. Wait for the startup screen to appear.
- 2 On the startup screen, click **PRODUCT DOCUMENTATION**. This action opens a *Windows Explorer* file tree window.
- 3 Click to open the **UTILITIES** folder. This folder contains all of the applications and files you will need to set up and configure your gateway.
- 4 Double-click the **SETUP CONFIGURATION TOOL** folder, double-click the **PCB_*.EXE** file and follow the instructions on your screen to install the software on your PC. The information represented by the "*" character in the file name is the *PCB* version number and, therefore, subject to change as new versions of *PCB* are released.

Note: Many of the configuration and maintenance procedures use files and other utilities on the CD-ROM. You may wish to copy the files from the *Utilities* folder on the CD-ROM to a convenient location on your hard drive.



3.1.1 Using the Online Help

Most of the information needed to help you use ProSoft Configuration Builder is provided in a Help System that is always available whenever you are running ProSoft Configuration Builder. The Help System does not require an Internet connection.

To view the help pages, start ProSoft Configuration Builder, open the **HELP** menu, and then choose **CONTENTS**.

3.2 Configure the Gateway

3.2.1 Configuring Module Parameters

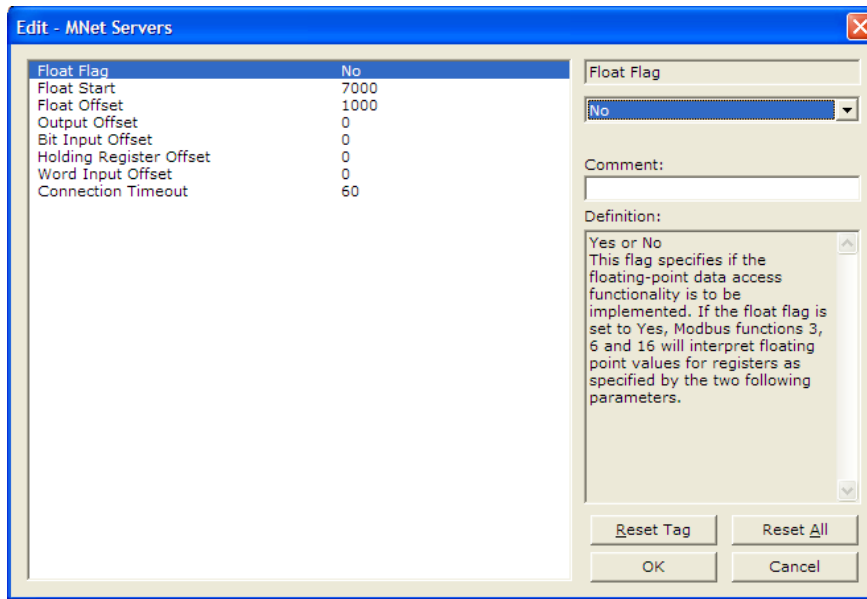
- 1 Click the **[+]** sign next to the module icon to expand gateway information.
- 2 Click the **[+]** sign next to any  icon to view gateway information and configuration options.
- 3 Double-click any  icon to open an *Edit* dialog box.
- 4 To edit a parameter, select the parameter in the left pane and make your changes in the right pane.
- 5 Click **OK** to save your changes.

3.2.2 Printing a Configuration File

- 1 Select the module icon, and then click the right mouse button to open a shortcut menu.
- 2 On the shortcut menu, choose **VIEW CONFIGURATION**. This action opens the *View Configuration* window.
- 3 In the *View Configuration* window, open the **FILE** menu, and choose **PRINT**. This action opens the *Print* dialog box.
- 4 In the *Print* dialog box, choose the printer to use from the drop-down list, select printing options, and then click **OK**.

3.3 MNET Servers

This section contains database offset information used by the server when accessed by external Clients. These offsets can be utilized to segment the database by data type.



3.3.1 Float Flag

YES or NO

This flag specifies if the floating-point data access functionality is to be implemented. If the float flag is set to **YES**, Modbus functions 3, 6, and 16 will interpret floating-point values for registers as specified by the two following parameters.

3.3.2 Float Start

0 to 65535

This parameter defines the first register of floating-point data. All requests with register values greater than or equal to this value will be considered floating-point data requests. This parameter is only used if the *Float Flag* is enabled. For example, if a value of **7000** is entered, all requests for registers 7000 and above will be considered as floating-point data.

3.3.3 Float Offset

0 to 9999

This parameter defines the start register for floating-point data in the internal database. This parameter is used only if the *Float Flag* is enabled. For example, if the *Float Offset* value is set to **3000** and the *Float Start* parameter is set to **7000**, data requests for register 7000 will use the internal Modbus register 3000.

3.3.4 Output Offset

0 to 3999

This parameter specifies the offset address in the internal Modbus database for network requests for Modbus function 1, 5, or 15 commands. For example, if the value is set to 100, an address request of 0 will return the value at register 100 in the database.

3.3.5 Bit Input Offset

0 to 3999

This parameter specifies the offset address in the internal Modbus database for network requests for Modbus function 2 commands. For example, if the value is set to 150, an address request of 0 will return the value at register 150 in the database.

3.3.6 Holding Register Offset

0 to 4999

This parameter specifies the offset address in the internal Modbus database to with network requests for Modbus functions 3, 6, or 16 commands. For example, if the value is set to 50, an address request of 0 will return the value at register 50 in the database.

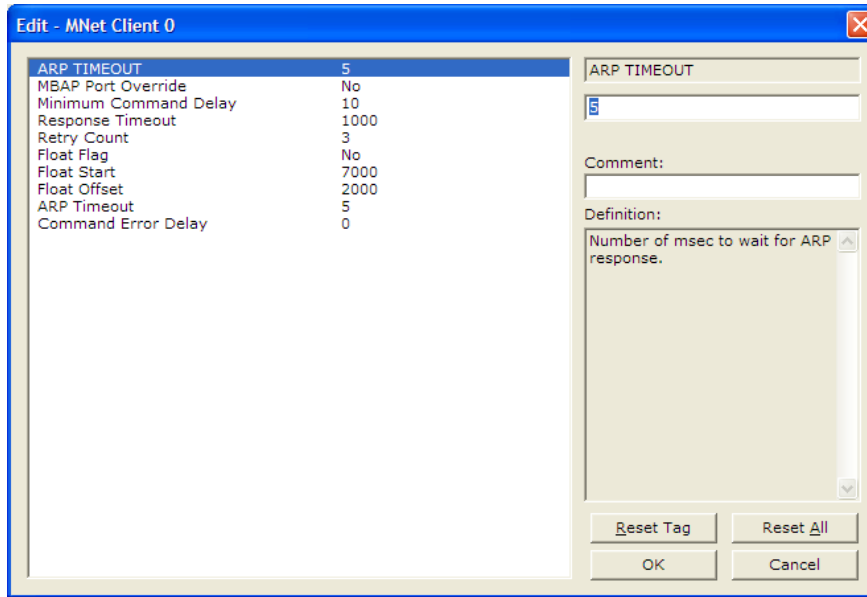
3.3.7 Word Input Offset

0 to 3999

This parameter specifies the offset address in the internal Modbus database for network requests for Modbus function 4 commands. For example, if the value is set to 150, an address request of 0 will return the value at register 150 in the database.

3.4 [MNET CLIENT x]

The [MNET CLIENT x] section of the **CFG** file specifies the parameters for the client to be emulated on the module. The command list for the client is entered in a separate section.



3.4.1 ARP Timeout

1 to 60

This parameter specifies the number of seconds to wait for an ARP reply after a request is issued.

3.4.2 MBAP Port Override

Y or N

If this parameter is set to Y (Yes), all messages generated by the client driver will be MBAP format messages to all Server Port values.

If this parameter is set to N (No) (default value), or is omitted from the configuration file, all messages sent to Server Port 502 will be MBAP format messages, and all other Server Ports values will use the encapsulated Modbus message format (MNET).

Each client is configured independently in the configuration file.

This parameter applies to firmware version 1.05 and above. For downward compatibility, you may omit this parameter from the client's configuration.

3.4.3 Minimum Command Delay

0 to 65535 milliseconds

This parameter specifies the number of milliseconds to wait between the initial issuance of a command. This parameter can be used to delay all commands sent to slaves to avoid "flooding" commands on the network. This parameter does not affect retries of a command as they will be issued when failure is recognized.

3.4.4 Response Timeout

0 to **65535** milliseconds

This is the time in milliseconds that a Client will wait before re-transmitting a command if no response is received from the addressed server. The value to use depends on the type of communication network used, and the expected response time of the slowest device on the network.

3.4.5 Retry Count

0 to **10**

This parameter specifies the number of times a command will be retried if it fails.

3.4.6 Float Flag

YES or **No**

This flag specifies if the floating-point data access functionality is to be implemented. If the float flag is set to **YES**, Modbus functions 3, 6, and 16 will interpret floating-point values for registers as specified by the two following parameters.

3.4.7 Float Start

0 to **65535**

This parameter defines the first register of floating-point data. All requests with register values greater than or equal to this value will be considered floating-point data requests. This parameter is only used if the *Float Flag* is enabled. For example, if a value of **7000** is entered, all requests for registers 7000 and above will be considered as floating-point data.

3.4.8 Float Offset

0 to **9999**

This parameter defines the start register for floating-point data in the internal database. This parameter is used only if the *Float Flag* is enabled. For example, if the *Float Offset* value is set to **3000** and the *Float Start* parameter is set to **7000**, data requests for register 7000 will use the internal Modbus register 3000.

3.5 MNET Client x Commands

The *MNET Client x Commands* section of the configuration sets the Modbus TCP/IP Client command list. This command list polls Modbus TCP/IP server devices attached to the same Ethernet network as the Modbus TCP/IP Client port. The module supports numerous commands. This permits the module to interface with a wide variety of Modbus TCP/IP protocol devices.

The function codes used for each command are those specified in the Modbus protocol. Each command list record has the same format. The first part of the record contains the information relating to the MNETC communication module, and the second part contains information required to interface to the Modbus TCP/IP server device.

3.5.1 Command List Overview

In order to interface the gateway with Modbus TCP/IP server devices, you must construct a command list. The commands in the list specify the server device to be addressed, the function to be performed (read or write), the data area in the device to interface with, and the registers in the internal database to be associated with the device data. The Client command list supports up to 16 commands.

The command list is processed from top (command #1) to bottom. A poll interval parameter is associated with each command to specify a minimum delay time in tenths of a second between the issuances of a command. If the user specifies a value of **10** for the parameter, the command will be executed no more frequently than once a second.

3.5.2 *Commands Supported by the Module*

The format of each command in the list depends on the Modbus Function Code being executed.

The following table lists the functions supported by the module.

Function Code	Definition	Supported in Client	Supported in Server
1	Read Coil Status	X	X
2	Read Input Status	X	X
3	Read Holding Registers	X	X
4	Read Input Registers	X	X
5	Set Single Coil	X	X
6	Single Register Write	X	X
7	Read Exception Status		X
8	Diagnostics		X
15	Multiple Coil Write	X	X
16	Multiple Register Write	X	X
22	Mask Write 4X		X
23	Read/Write		X

Each command list record has the same general format. The first part of the record contains the information relating to the communication module and the second part contains information required to interface to the Modbus TCP/IP Client server device.

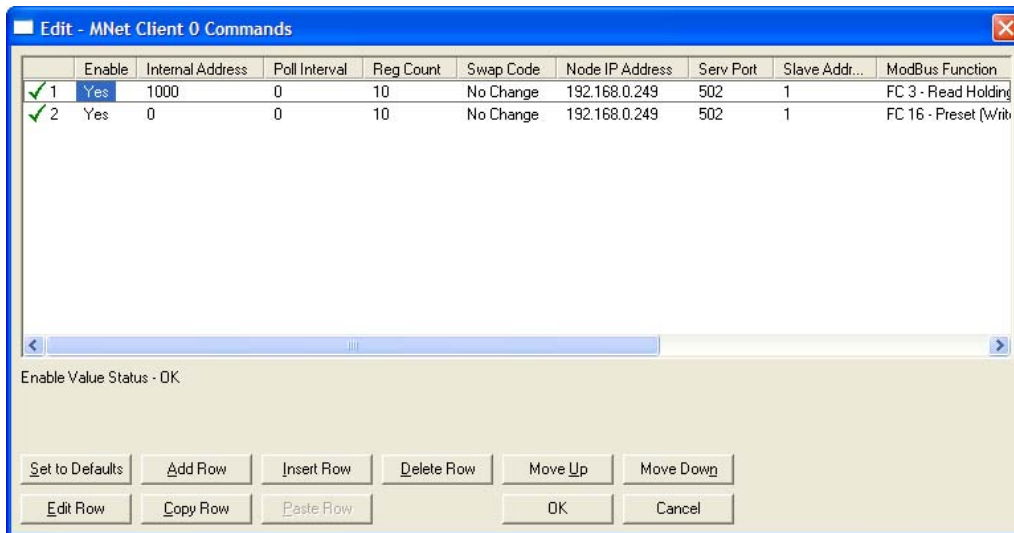
3.5.3 Command Entry Formats

The following table shows the structure of the configuration data necessary for each of the supported commands.

1	2	3	4	5	6	7	8	9	10
Enable Code	Internal Address	Poll Interval Time	Count	Swap Code	IP Address	Serv Port	Slave Node	Function Code	Device Modbus Address
Code	Register (bit)	1/10th Seconds	Bit Count	0	IP Address	Port #	Address	Read Coil (0x)	Register
Code	Register (bit)	1/10th Seconds	Bit Count	0	IP Address	Port #	Address	Read Input (1x)	Register
Code	Register	1/10th Seconds	Word Count	Code	IP Address	Port #	Address	Read Holding Registers (4x)	Register
Code	Register	1/10th Seconds	Word Count	0	IP Address	Port #	Address	Read Input Registers (3x)	Register
Code	1 bit	1/10th Seconds	Bit Count	0	IP Address	Port #	Address	Force (Write) Single Coil (0x)	Register
Code	1 bit	1/10th Seconds	Word Count	0	IP Address	Port #	Address	Preset (Write) Single Register (4x)	Register
Code	Register (bit)	1/10th Seconds	Bit Count	0	IP Address	Port #	Address	Force (Write) Multiple Coil (0x)	Register
Code	Register	1/10th Seconds	Word Count	0	IP Address	Port #	Address	Preset (Write) Multiple Register (4x)	Register

The first part of the record is the gateway information, which relates to the ProLinx gateway, and the second part contains information required to interface to the server device.

Command list example:



3.5.4 *Enable*

No (0) or **YES** (1)

This field defines whether or not the command is to be executed.

Value	Description
No (0)	The command is disabled and will not be executed in the normal polling sequence.
Yes (1)	The command is executed each scan of the command list if the Poll Interval Time is set to zero (0). If the Poll Interval time is set, the command will be executed when the interval timer expires.

3.5.5 *Internal Address*

0 to **65535** (for bit-level addressing)

or

0 to **4999** (for word-level addressing)

This field specifies the database address in the gateway's internal database to use as the destination for data brought in by a read command or as the source for data to be sent out by a write command. The database address is interpreted as a bit address or a 16-bit word (register) address, depending on the Modbus Function Code used in the command.

- For Modbus functions 1, 2, 5, and 15, this parameter is interpreted as a bit-level address.
- For Modbus functions 3, 4, 6, and 16, this parameter is interpreted as a word-level or register-level address.

3.5.6 *Poll Interval*

0 to **65535**

This parameter specifies the minimum interval between issuances of a command during continuous command execution (*Enable* code of **1**). The parameter is entered in tenths of a second. Therefore, if a value of **100** is entered for a command, the command executes no more frequently than every 10 seconds.

3.5.7 Reg Count

Regs: 1 to 125

Coils: 1 to 800

This parameter specifies the number of 16-bit registers or binary bits to be transferred by the command.

- Functions 5 and 6 ignore this field as they apply only to a single data point.
- For functions 1, 2, and 15, this parameter sets the number of bits (inputs or coils) to be transferred by the command.
- For functions 3, 4, and 16, this parameter sets the number of registers to be transferred by the command.

3.5.8 Swap Code

NONE

SWAP WORDS

SWAP WORDS & BYTES

SWAP BYTES

This parameter defines if and how the order of bytes in data received or sent is to be rearranged. This option exists to allow for the fact that different manufacturers store and transmit multi-byte data in different combinations. This parameter is helpful when dealing with floating-point or other multi-byte values, as there is no one standard method of storing these data types. The parameter can be set to rearrange the byte order of data received or sent into an order more useful or convenient for other applications. The following table defines the valid *Swap Code* values and the effect they have on the byte-order of the data.

Swap Code	Description
NONE	No change is made in the byte ordering (1234 = 1234)
SWAP WORDS	The words are swapped (1234=3412)
SWAP WORDS & BYTES	The words are swapped, then the bytes in each word are swapped (1234=4321)
SWAP BYTES	The bytes in each word are swapped (1234=2143)

These swap operations affect 4-byte (or 2-word) groups of data. Therefore, data swapping using these *Swap Codes* should be done only when using an even number of words, such as when 32-bit integer or floating-point data is involved.

3.5.9 Node IP Address

xxx.xxx.xxx.xxx

The IP address of the device being addressed by the command.

3.5.10 Service Port

502 or other port numbers supported on a server

Use a value of **502** when addressing Modbus TCP/IP servers that are compatible with the Schneider Electric MBAP specifications (this will be most devices). If a server implementation supports another service port, enter the value here.

3.5.11 Slave Address

0 - Broadcast to all nodes

1 to 255

Use this parameter to specify the slave address of a remote Modbus Serial device through a Modbus Ethernet to Serial converter.

Note: Use the *Node IP Address* parameter (page 26) to address commands to a remote Modbus TCP/IP device.

Note: Most Modbus devices accept an address in the range of only 1 to 247, so check with the slave device manufacturer to see if a particular slave can use addresses 248 to 255.

If the value is set to zero, the command will be a broadcast message on the network. The Modbus protocol permits broadcast commands for write operations. Do not use node address 0 for read operations.

3.5.12 Modbus Function

1, 2, 3, 4, 5, 6, 15, or 16

This parameter specifies the Modbus Function Code to be executed by the command. These function codes are defined in the Modbus protocol. The following table lists the purpose of each function supported by the module. More information on the protocol is available from www.modbus.org.

Modbus Function Code	Description
1	Read Coil Status
2	Read Input Status
3	Read Holding Registers
4	Read Input Registers
5	Force (Write) Single Coil
6	Preset (Write) Single Register
15	Force Multiple Coils
16	Preset Multiple Registers

3.5.13 MB Address in Device

This parameter specifies the starting Modbus register or bit address in the Server to be used by the command. Refer to the documentation of each Modbus Server device for the register and bit address assignments valid for that device.

The Modbus Function Code determines whether the address will be a register-level or bit-level OFFSET address into a given data type range. The offset will be the target data address in the Server minus the base address for that data type. Base addresses for the different data types are:

- 00001 or 000001 (0x0001) for bit-level Coil data (Function Codes 1, 5, and 15).
- 10001 or 100001 (1x0001) for bit-level Input Status data (Function Code 2)
- 30001 or 300001 (3x0001) for Input Register data (Function Code 4)
- 40001 or 400001 (4x0001) for Holding Register data (Function Codes 3, 6, and 16).

Address calculation examples:

- For bit-level Coil commands (FC 1, 5, or 15) to read or write a Coil 0X address 00001, specify a value of 0 (00001 - 00001 = 0).
- For Coil address 00115, specify 114
(00115 - 00001 = 114)
- For register read or write commands (FC 3, 6, or 16) 4X range, for 40001, specify a value of 0
(40001 - 40001 = 0).
- For 01101, 11101, 31101 or 41101, specify a value of 1100.
(01101 - 00001 = 1100)
(11101 - 10001 = 1100)
(31101 - 30001 = 1100)
(41101 - 40001 = 1100)

Note: If the documentation for a particular Modbus Server device lists data addresses in hexadecimal (base16) notation, you will need to convert the hexadecimal value to a decimal value to enter in this parameter. In such cases, it is not usually necessary to subtract 1 from the converted decimal number, as this addressing scheme typically uses the exact offset address expressed as a hexadecimal number.

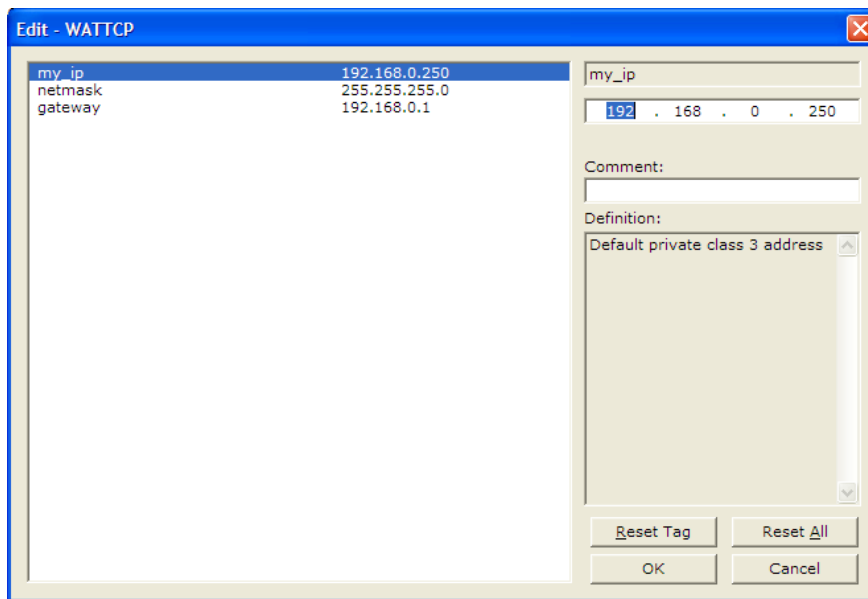
3.6 Ethernet Configuration

Use this procedure to configure the Ethernet settings for your module. You must assign an IP address, subnet mask and gateway address. After you complete this step, you can connect to the module with an Ethernet cable.

- 1 Determine the network settings for your module, with the help of your network administrator if necessary. You will need the following information:
 - IP address (fixed IP required) _____ . _____ . _____ . _____
 - Subnet mask _____ . _____ . _____ . _____
 - Gateway address _____ . _____ . _____ . _____

Note: The gateway address is optional, and is not required for networks that do not use a default gateway.

- 2 Double-click the **ETHERNET CONFIGURATION** icon. This action opens the *Edit* dialog box.



- 3 Edit the values for *my_ip*, *netmask* (subnet mask) and *gateway* (default gateway).
- 4 When you are finished editing, click **OK** to save your changes and return to the *ProSoft Configuration Builder* window.

4 Client Error/Status Data

In This Chapter

- ❖ MNET Error and Status Data..... 31
- ❖ MNET Client Error/Status Data 33
- ❖ Client Command Errors 37
- ❖ Ethernet LED Indicators 39

4.1 MNET Error and Status Data

The following topics list the register addresses that contain error and status data. Use the Database View option from the ProLinx Main Menu to view the contents of each register. The *ProLinx Reference Guide* provides the information on using this option.

4.1.1 MNET Server (Port 2000)

Status Register	Description
6200	Command Request Count
6201	Command Response Count
6202	Command Error Count
6203	Number of Request Packets
6204	Number of Response Packets
6205	Errors Sent
6206	Errors Received
6207	Configuration Error Word
6208	Current Error
6209	Last Error

4.1.2 MNET Server (Port 502)

Status Register	Description
6210	Command Request Count
6211	Command Response Count
6212	Command Error Count
6213	Number of Request Packets
6214	Number of Response Packets
6215	Errors Sent
6216	Errors Received
6217	Configuration Error Word
6218	Current Error
6219	Last Error

4.2 MNET Client Error/Status Data

The Client connection Error and Status Data areas are discussed in this section. The error/status data for each Client command is stored virtual data addresses 15500 to 15525 for Client 0, 15526 to 15551 for Client 1, etc. These error/status codes can be moved into the user data area by mapping them in the [DATA MAP] section of ProSoft Configuration Builder. The detailed contents of this virtual data area are shown in the next section.

The data area is initialized with zeros whenever the gateway is initialized. This occurs during a cold-start (power-on), reset (reset push-button pressed) or a warm-boot operation (commanded from the Debug menu or when loading a new configuration).

4.2.1 MNET Client [0]

Status Register	Description
15500	Command Request Count
15501	Command Response Count
15502	Command Error Count
15503	Number of Request Packets
15504	Number of Response Packets
15505	Errors Sent
15506	Errors Received
15507	Configuration Error Word
15508	Current Error
15509	Last Error
15510 to 15525	Command List Errors (16 per client)

4.2.2 MNET Client [1]

Status Register	Description
15526	Command Request Count
15527	Command Response Count
15528	Command Error Count
15529	Number of Request Packets
15530	Number of Response Packets
15531	Errors Sent
15532	Errors Received
15533	Configuration Error Word
15534	Current Error
15535	Last Error
15536 to 15551	Command List Errors (16 per client)

4.2.3 MNET Client [2]

Status Register	Description
15552	Command Request Count
15553	Command Response Count
15554	Command Error Count
15555	Number of Request Packets
15556	Number of Response Packets
15557	Errors Sent
15558	Errors Received
15559	Configuration Error Word
15560	Current Error
15561	Last Error
15562 to 15577	Command List Errors (16 per client)

4.2.4 MNET Client [3]

Status Register	Description
15578	Command Request Count
15579	Command Response Count
15580	Command Error Count
15581	Number of Request Packets
15582	Number of Response Packets
15583	Errors Sent
15584	Errors Received
15585	Configuration Error Word
15586	Current Error
15587	Last Error
15588 to 15603	Command List Errors (16 per client)

4.2.5 MNET Client [4]

Status Register	Description
15604	Command Request Count
15605	Command Response Count
15606	Command Error Count
15607	Number of Request Packets
15608	Number of Response Packets
15609	Errors Sent
15610	Errors Received
15611	Configuration Error Word
15612	Current Error
15613	Last Error
15614 15629	Command List Errors (16 per client)

4.2.6 MNET Client [5]

Status Register	Description
15630	Command Request Count
15631	Command Response Count
15632	Command Error Count
15633	Number of Request Packets
15634	Number of Response Packets
15635	Errors Sent
15636	Errors Received
15637	Configuration Error Word
15638	Current Error
15639	Last Error
15640 to 15655	Command List Errors (16 per client)

4.2.7 MNET Client [6]

Status Register	Description
15656	Command Request Count
15657	Command Response Count
15658	Command Error Count
15659	Number of Request Packets
15660	Number of Response Packets
15661	Errors Sent
15662	Errors Received
15663	Configuration Error Word
15664	Current Error
15665	Last Error
15666 to 15681	Command List Errors (16 per client)

4.2.8 MNET Client [7]

Status Register	Description
15682	Command Request Count
15683	Command Response Count
15684	Command Error Count
15685	Number of Request Packets
15686	Number of Response Packets
15687	Errors Sent
15688	Errors Received
15689	Configuration Error Word
15690	Current Error
15691	Last Error
15692 to 15707	Command List Errors (16 per client)

4.2.9 MNET Client [8]

Status Register	Description
15708	Command Request Count
15709	Command Response Count
15710	Command Error Count
15711	Number of Request Packets
15712	Number of Response Packets
15713	Errors Sent
15714	Errors Received
15715	Configuration Error Word
15716	Current Error
15717	Last Error
15718 to 15733	Command List Errors (16 per client)

4.2.10 MNET Client [9]

Status Register	Description
15734	Command Request Count
15735	Command Response Count
15736	Command Error Count
15737	Number of Request Packets
15738	Number of Response Packets
15739	Errors Sent
15740	Errors Received
15741	Configuration Error Word
15742	Current Error
15743	Last Error
15744 to 15759	Command List Errors (16 per client)

4.3 Client Command Errors

There are several different ways to view Client Command Errors.

- In the *MNETC.STATUS.CmdErrorList* controller tag array
- On the Client status data screens in the *ProSoft Configuration Builder Diagnostics*
- At a module database location specified by the configuration's *MNET Client x Command Error Pointer*, if the *Command Error Pointer* is enabled. This means that the first register refers to command 1 and so on.

Word Offset	Description
0	Command 0 Error
1	Command 1 Error
2	Command 2 Error
3	Command 3 Error
...
...	...
15	Command 15 Error
16	Command 16 Error

For every command that has an error, the module automatically sets the poll delay parameter to 30 seconds. This instructs the module to wait 30 seconds until it attempts to issue the command again.

As the commands in the Client Command Last are polled and executed, an error value is maintained in the module for each command. This error list can be transferred to the processor.

4.3.1 Standard Modbus Exception Code Errors

Code	Description
1	Illegal function
2	Illegal data address
3	Illegal data value
4	Failure in associated device
5	Acknowledge
6	Busy; message was rejected

4.3.2 Module Communication Error Codes

Code	Description
-2	Timeout while transmitting message
-11	Timeout waiting for response after request (same as -36)
253	Incorrect slave/server address in response
254	Incorrect function code in response
255	Invalid CRC/LRC value in response

4.3.3 MNET Client Specific Errors

Code	Description
-33	Failed to connect to server specified in command
-35	Invalid length of response message
-36	MNET command response timeout (same as -11)
-37	TCP/IP connection ended before session finished

4.3.4 Command List Entry Errors

Code	Description
-40	Too few parameters
-41	Invalid enable code
-42	Internal address > maximum address
-43	Invalid node address (<0 or >255)
-44	Count parameter set to 0
-45	Invalid function code
-46	Invalid swap code
-47	ARP could not resolve MAC from IP (bad IP address, not part of a network, invalid parameter to ARP routine).
-48	Error during ARP operation: the response to the ARP request did not arrive to the module after a user-adjustable ARP Timeout.

Note: When the Client gets error -47 or -48, it uses the adjustable ARP Timeout parameter in the configuration file to set an amount of time to wait before trying again to connect to this non-existent server. This feature allows the Client to continue sending commands and polling other existing servers, while waiting for the non-existent server to appear on the network.

4.4 Ethernet LED Indicators

LED	State	Description
Data	OFF	No activity on the Ethernet port.
	GREEN Flash	The Ethernet port is actively transmitting or receiving data.
Link	OFF	No physical network connection is detected. No Ethernet communication is possible. Check wiring and cables.
	GREEN Solid	Physical network connection detected. This LED must be ON solid for Ethernet communication to be possible.

5 Modbus Protocol Specification

In This Chapter

❖ Read Coil Status (Function Code 01)	42
❖ Read Input Status (Function Code 02)	43
❖ Read Holding Registers (Function Code 03)	44
❖ Read Input Registers (Function Code 04)	45
❖ Force Single Coil (Function Code 05)	46
❖ Preset Single Register (Function Code 06)	47
❖ Diagnostics (Function Code 08)	48
❖ Force Multiple Coils (Function Code 15)	56
❖ Preset Multiple Registers (Function Code 16)	57

5.1 Read Coil Status (Function Code 01)

Query

This function allows the user to obtain the ON/OFF status of logic coils used to control discrete outputs from the addressed Server only. Broadcast mode is not supported with this function code. In addition to the Server address and function fields, the message requires that the information field contain the initial coil address to be read (Starting Address) and the number of locations that will be interrogated to obtain status data.

The addressing allows up to 2000 coils to be obtained at each request; however, the specific Server device may have restrictions that lower the maximum quantity. The coils are numbered from zero; (coil number 1 = zero, coil number 2 = one, coil number 3 = two, and so on).

The following table is a sample read output status request to read coils 0020 to 0056 from Server device number 11.

Adr	Func	Data Start Pt Hi	Data Start Pt Lo	Data # Of Pts Ho	Data # Of Pts Lo	Error Check Field
11	01	00	13	00	25	CRC

Response

An example response to Read Coil Status is as shown in Figure C2. The data is packed one bit for each coil. The response includes the Server address, function code, quantity of data characters, the data characters, and error checking. Data will be packed with one bit for each coil (1 = ON, 0 = OFF). The low order bit of the first character contains the addressed coil, and the remainder follow. For coil quantities that are not even multiples of eight, the last characters will be filled in with zeros at high order end. The quantity of data characters is always specified as quantity of RTU characters, that is, the number is the same whether RTU or ASCII is used.

Because the Server interface device is serviced at the end of a controller's scan, data will reflect coil status at the end of the scan. Some Servers will limit the quantity of coils provided each scan; thus, for large coil quantities, multiple PC transactions must be made using coil status from sequential scans.

Adr	Func	Byte Count	Data Coil Status 20 to 27	Data Coil Status 28 to 35	Data Coil Status 36 to 43	Data Coil Status 44 to 51	Data Coil Status 52 to 56	Error Check Field
11	01	05	CD	6B	B2	OE	1B	CRC

The status of coils 20 to 27 is shown as CD(HEX) = 1100 1101 (Binary). Reading left to right, this shows that coils 27, 26, 23, 22, and 20 are all on. The other coil data bytes are decoded similarly. Due to the quantity of coil statuses requested, the last data field, which is shown 1B (HEX) = 0001 1011 (Binary), contains the status of only 5 coils (52 to 56) instead of 8 coils. The 3 left most bits are provided as zeros to fill the 8-bit format.

5.2 Read Input Status (Function Code 02)

Query

This function allows the user to obtain the ON/OFF status of discrete inputs in the addressed Server PC Broadcast mode is not supported with this function code. In addition to the Server address and function fields, the message requires that the information field contain the initial input address to be read (Starting Address) and the number of locations that will be interrogated to obtain status data.

The addressing allows up to 2000 inputs to be obtained at each request; however, the specific Server device may have restrictions that lower the maximum quantity. The inputs are numbered from zero; (input 10001 = zero, input 10002 = one, input 10003 = two, and so on, for a 584).

The following table is a sample read input status request to read inputs 10197 to 10218 from Server number 11.

Adr	Func	Data Start Pt Hi	Data Start Pt Lo	Data #of Pts Hi	Data #of Pts Lo	Error Check Field
11	02	00	C4	00	16	CRC

Response

An example response to Read Input Status is as shown in Figure C4. The data is packed one bit for each input. The response includes the Server address, function code, quantity of data characters, the data characters, and error checking. Data will be packed with one bit for each input (1=ON, 0=OFF). The lower order bit of the first character contains the addressed input, and the remainder follow. For input quantities that are not even multiples of eight, the last characters will be filled in with zeros at high order end. The quantity of data characters is always specified as a quantity of RTU characters, that is, the number is the same whether RTU or ASCII is used.

Because the Server interface device is serviced at the end of a controller's scan, data will reflect input status at the end of the scan. Some Servers will limit the quantity of inputs provided each scan; thus, for large coil quantities, multiple PC transactions must be made using coil status for sequential scans.

Adr	Func	Byte Count	Data Discrete Input 10197 to 10204	Data Discrete Input 10205 to 10212	Data Discrete Input 10213 to 10218	Error Check Field
11	02	03	AC	DB	35	CRC

The status of inputs 10197 to 10204 is shown as AC (HEX) = 10101 1100 (binary). Reading left to right, this show that inputs 10204, 10202, and 10199 are all on. The other input data bytes are decoded similar.

Due to the quantity of input statuses requested, the last data field which is shown as 35 HEX = 0011 0101 (binary) contains the status of only 6 inputs (10213 to 10218) instead of 8 inputs. The two left-most bits are provided as zeros to fill the 8-bit format.

5.3 Read Holding Registers (Function Code 03)

Query

Read Holding Registers (03) allows the user to obtain the binary contents of holding registers 4xxxx in the addressed Server. The registers can store the numerical values of associated timers and counters which can be driven to external devices. The addressing allows up to 125 registers to be obtained at each request; however, the specific Server device may have a restriction that is lower than this maximum quantity. The registers are numbered from zero (40001 = zero, 40002 = one, and so on). The broadcast mode is not allowed.

The example below reads registers 40108 through 40110 from Server 584 number 11.

Adr	Func	Data Start Reg Hi	Data Start Reg Lo	Data #of Regs Hi	Data #of Regs Lo	Error Check Field
11	03	00	6B	00	03	CRC

Response

The addressed Server responds with its address and the function code, followed by the information field. The information field contains 1 byte describing the quantity of data bytes to be returned. The contents of the registers requested (DATA) are two bytes each, with the binary content right justified within each pair of characters. The first byte includes the high order bits and the second, the low order bits.

Because the Server interface device is normally serviced at the end of the controller's scan, the data will reflect the register content at the end of the scan. Some Servers will limit the quantity of register content provided each scan; thus for large register quantities, multiple transmissions will be made using register content from sequential scans.

In the example below, the registers 40108 to 40110 have the decimal contents 555, 0, and 100 respectively.

Adr	Func	ByteCnt	Hi Data	Lo Data	Hi Data	Lo Data	Hi Data	Lo Data	Error Check Field
11	03	06	02	2B	00	00	00	64	CRC

5.4 Read Input Registers (Function Code 04)

Query

Function code 04 obtains the contents of the controller's input registers at addresses 3xxxx. These locations receive their values from devices connected to the I/O structure and can only be referenced, not altered from within the controller. The addressing allows up to 125 registers to be obtained at each request; however, the specific Server device may have restrictions that lower this maximum quantity. The registers are numbered for zero (30001 = zero, 30002 = one, and so on). Broadcast mode is not allowed.

The example below requests the contents of register 3009 in Server number 11.

Adr	Func	Data Start Reg Hi	Data Start Reg Lo	Data #of Regs Hi	Data #of Regs Lo	Error Check Field
11	04	00	08	00	01	CRC

Response

The addressed Server responds with its address and the function code followed by the information field. The information field contains 1 byte describing the quantity of data bytes to be returned. The contents of the registers requested (DATA) are 2 bytes each, with the binary content right justified within each pair of characters. The first byte includes the high order bits and the second, the low order bits.

Because the Server interface is normally serviced at the end of the controller's scan, the data will reflect the register content at the end of the scan. Each PC will limit the quantity of register contents provided each scan; thus for large register quantities, multiple PC scans will be required, and the data provided will be form sequential scans.

In the example below the register 3009 contains the decimal value 0.

Adr	Func	Byte Count	Data Input Reg Hi	Data Input Reg Lo	Error Check Field
11	04	02	00	00	E9

5.5 Force Single Coil (Function Code 05)

Query

This message forces a single coil either ON or OFF. Any coil that exists within the controller can be forced to either state (ON or OFF). However, because the controller is actively scanning, unless the coil is disabled, the controller can also alter the state of the coil. Coils are numbered from zero (coil 0001 = zero, coil 0002 = one, and so on). The data value 65,280 (FF00 HEX) will set the coil ON and the value zero will turn it OFF; all other values are illegal and will not affect that coil.

The use of Server address 00 (Broadcast Mode) will force all attached Servers to modify the desired coil.

Note: Functions 5, 6, 15, and 16 are the only messages that will be recognized as valid for broadcast.

The example below is a request to Server number 11 to turn ON coil 0173.

Adr	Func	Data Coil # Hi	Data Coil # Lo	Data On/off Ind	Data	Error Check Field
11	05	00	AC	FF	00	CRC

Response

The normal response to the Command Request is to re-transmit the message as received after the coil state has been altered.

Adr	Func	Data Coil # Hi	Data Coil # Lo	Data On/ Off	Data	Error Check Field
11	05	00	AC	FF	00	CRC

The forcing of a coil via MODBUS function 5 will be accomplished regardless of whether the addressed coil is disabled or not (*In ProSoft products, the coil is only affected if the necessary ladder logic is implemented*).

Note: The Modbus protocol does not include standard functions for testing or changing the DISABLE state of discrete inputs or outputs. Where applicable, this may be accomplished via device specific Program commands (*In ProSoft products, this is only accomplished through ladder logic programming*).

Coils that are reprogrammed in the controller logic program are not automatically cleared upon power up. Thus, if such a coil is set ON by function Code 5 and (even months later), an output is connected to that coil, the output will be "hot".

5.6 Preset Single Register (Function Code 06)

Query

Function (06) allows the user to modify the contents of a holding register. Any holding register that exists within the controller can have its contents changed by this message. However, because the controller is actively scanning, it also can alter the content of any holding register at any time. The values are provided in binary up to the maximum capacity of the controller unused high order bits must be set to zero. When used with Server address zero (Broadcast mode) all Server controllers will load the specified register with the contents specified.

Note Functions 5, 6, 15, and 16 are the only messages that will be recognized as valid for broadcast.

Adr	Func	Data Start Reg Hi	Data Start Reg Lo	Data #of Regs Hi	Data #of Regs Lo	Error Check Field
11	06	00	01	00	03	CRC

Response

The response to a preset single register request is to re-transmit the query message after the register has been altered.

Adr	Func	Data Reg Hi	Data Reg Lo	Data Input Reg Hi	Data Input Reg Lo	Error Check Field
11	06	00	01	00	03	CRC

5.7 Diagnostics (Function Code 08)

MODBUS function code 08 provides a series of tests for checking the communication system between a Client device and a server, or for checking various internal error conditions within a server.

The function uses a two-byte sub-function code field in the query to define the type of test to be performed. The server echoes both the function code and sub-function code in a normal response. Some of the diagnostics cause data to be returned from the remote device in the data field of a normal response.

In general, issuing a diagnostic function to a remote device does not affect the running of the user program in the remote device. Device memory bit and register data addresses are not accessed by the diagnostics. However, certain functions can optionally reset error counters in some remote devices.

A server device can, however, be forced into 'Listen Only Mode' in which it will monitor the messages on the communications system but not respond to them. This can affect the outcome of your application program if it depends upon any further exchange of data with the remote device. Generally, the mode is forced to remove a malfunctioning remote device from the communications system.

5.7.1 Sub-function Codes Supported

Only Sub-function 00 is supported by the MNETC gateway.

00 Return Query Data

The data passed in the request data field is to be returned (looped back) in the response. The entire response message should be identical to the request.

Sub-function	Data Field (Request)	Data Field (Response)
00 00	Any	Echo Request Data

01 Restart Communications Option

The remote device serial line port must be initialized and restarted, and all of its communications event counters are cleared. If the port is currently in Listen Only Mode, no response is returned. This function is the only one that brings the port out of Listen Only Mode. If the port is not currently in Listen Only Mode, a normal response is returned. This occurs before the restart is executed.

When the remote device receives the request, it attempts a restart and executes its power-up confidence tests. Successful completion of the tests will bring the port online.

A request data field contents of FF 00 hex causes the port's Communications Event Log to be cleared also. Contents of 00 00 leave the log as it was prior to the restart.

Sub-function	Data Field (Request)	Data Field (Response)
00 01	00 00	Echo Request Data
00 01	FF 00	Echo Request Data

02 Return Diagnostic Register

Sub-function	Data Field (Request)	Data Field (Response)
00 02	00 00	Diagnostic Register Contents

03 Change ASCII Input Delimiter

The character "CHAR" passed in the request data field becomes the end of message delimiter for future messages (replacing the default LF character). This function is useful in cases of a Line Feed is not required at the end of ASCII messages.

Sub-function	Data Field (Request)	Data Field (Response)
00 03	CHAR 00	Echo Request Data

04 Force Listen Only Mode

Forces the addressed remote device to its Listen Only Mode for MODBUS communications. This isolates it from the other devices on the network, allowing them to continue communicating without interruption from the addressed remote device. No response is returned.

When the remote device enters its Listen Only Mode, all active communication controls are turned off. The Ready watchdog timer is allowed to expire, locking the controls off. While the device is in this mode, any MODBUS messages addressed to it or broadcast are monitored, but no actions will be taken and no responses will be sent.

The only function that will be processed after the mode is entered will be the Restart Communications Option function (function code 8, sub-function 1).

Sub-function	Data Field (Request)	Data Field (Response)
00 04	00 00	No Response Returned

10 (0A Hex) Clear Counters and Diagnostic Register

The goal is to clear all counters and the diagnostic register. Counters are also cleared upon power-up.

Sub-function	Data Field (Request)	Data Field (Response)
00 0A	00 00	Echo Request Data

11 (0B Hex) Return Bus Message Count

The response data field returns the quantity of messages that the remote device has detected on the communications system since its last restart, clear counters operation, or power-up.

Sub-function	Data Field (Request)	Data Field (Response)
00 0B	00 00	Total Message Count

12 (0C Hex) Return Bus Communication Error Count

The response data field returns the quantity of CRC errors encountered by the remote device since its last restart, clear counters operation, or power-up.

Sub-function	Data Field (Request)	Data Field (Response)
00 0C	00 00	CRC Error Count

13 (0D Hex) Return Bus Exception Error Count

The response data field returns the quantity of MODBUS exception responses returned by the remote device since its last restart, clear counters operation, or power-up.

Exception responses are described and listed in MODBUS Exception Responses (page 53).

Sub-function	Data Field (Request)	Data Field (Response)
00 0D	00 00	Exception Error Count

14 (0E Hex) Return Slave Message Count

The response data field returns the quantity of messages addressed to the remote device, or broadcast, that the remote device has processed since its last restart, clear counters operation, or power-up.

Sub-function	Data Field (Request)	Data Field (Response)
00 0E	00 00	Slave Message Count

15 (0F Hex) Return Slave No Response Count

The response data field returns the quantity of messages addressed to the remote device for which it has returned no response (neither a normal response nor an exception response), since its last restart, clear counters operation, or power-up.

Sub-function	Data Field (Request)	Data Field (Response)
00 0F	00 00	Slave No Response Count

16 (10 Hex) Return Slave NAK Count

The response data field returns the quantity of messages addressed to the remote device for which it returned a Negative Acknowledge (NAK) exception response, since its last restart, clear counters operation, or power-up. Exception responses are described and listed in section 7 .

Sub-function	Data Field (Request)	Data Field (Response)
00 10	00 00	Slave NAK Count

17 (11 Hex) Return Slave Busy Count

The response data field returns the quantity of messages addressed to the remote device for which it returned a Slave Device Busy exception response, since its last restart, clear counters operation, or power-up.

Sub-function	Data Field (Request)	Data Field (Response)
00 11	00 00	Slave Device Busy Count

18 (12 Hex) Return Bus Character Overrun Count

The response data field returns the quantity of messages addressed to the remote device that it could not handle due to a character overrun condition, since its last restart, clear counters operation, or power-up. A character overrun is caused by data characters arriving at the port faster than they can be stored, or by the loss of a character due to a hardware malfunction.

Sub-function	Data Field (Request)	Data Field (Response)
00 12	00 00	Slave Character Overrun Count

20 (14 Hex) Clear Overrun Counter and Flag

Clears the overrun error counter and reset the error flag.

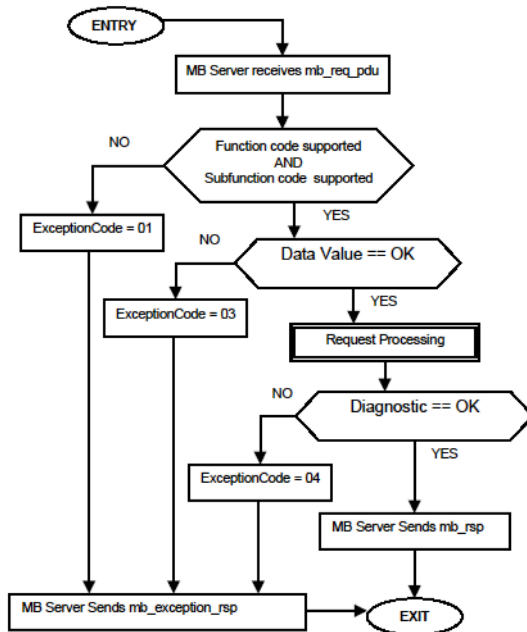
Sub-function	Data Field (Request)	Data Field (Response)
00 14	00 00	Echo Request Data

Example and State Diagram

Here is an example of a request to remote device to Return Query Data. This uses a sub-function code of zero (00 00 hex in the two-byte field). The data to be returned is sent in the two-byte data field (A5 37 hex).

Request		Response	
Field Name	(Hex)	Field Name	(Hex)
Function	08	Function	08
Sub-function Hi	00	Sub-function Hi	00
Sub-function Lo	00	Sub-function Lo	00
Data Hi	A5	Data Hi	A5
Data Lo	37	Data Lo	27

The data fields in responses to other kinds of queries could contain error counts or other data requested by the sub-function code.



5.7.2 Modbus Exception Responses

When a Modbus Client sends a request to a Server device, it expects a normal response. One of four possible events can occur from the Client's query:

- If the server device receives the request without a communication error, and can handle the query normally, it returns a normal response.
- If the server does not receive the request due to a communication error, no response is returned. The Client program will eventually process a timeout condition for the request.
- If the server receives the request, but detects a communication error (parity, LRC, CRC, ...), no response is returned. The Client program will eventually process a timeout condition for the request.
- If the server receives the request without a communication error, but cannot handle it (for example, if the request is to read a non-existent output or register), the server will return an exception response informing the Client of the nature of the error.

The exception response message has two fields that differentiate it from a normal response:

Function Code Field: In a normal response, the server echoes the function code of the original request in the function code field of the response. All function codes have a most-significant bit (MSB) of 0 (their values are all below 80 hexadecimal). In an exception response, the server sets the MSB of the function code to 1. This makes the function code value in an exception response exactly 80 hexadecimal higher than the value would be for a normal response.

With the function code's MSB set, the Client's application program can recognize the exception response and can examine the data field for the exception code.

Data Field: In a normal response, the server may return data or statistics in the data field (any information that was requested in the request). In an exception response, the server returns an exception code in the data field. This defines the server condition that caused the exception.

The following table shows an example of a Client request and server exception response.

Request		Response	
Field Name	(Hex)	Field Name	(Hex)
Function	01	Function	81
Starting Address Hi	04	Exception Code	02
Starting Address Lo	A1		
Quantity of Outputs Hi	00		
Quantity of Outputs Lo	01		

In this example, the Client addresses a request to server device. The function code (01) is for a Read Output Status operation. It requests the status of the output at address 1245 (04A1 hex). Note that only that one output is to be read, as specified by the number of outputs field (0001).

If the output address is non-existent in the server device, the server will return the exception response with the exception code shown (02). This specifies an illegal data address for the Server.

Modbus Exception Codes

Code	Name	Meaning
01	Illegal Function	The function code received in the query is not an allowable action for the Server. This may be because the function code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the Server is in the wrong state to process a request of this type, for example because it is unconfigured and is being asked to return register values.
02	Illegal Data Address	The data address received in the query is not an allowable address for the Server. More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, a request with offset 96 and length 4 would succeed; a request with offset 96 and length 5 will generate exception 02.
03	Illegal Data Value	A value contained in the query data field is not an allowable value for Server. This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does not mean that a data item submitted for storage in a register has a value outside the expectation of the application program, because the Modbus protocol is unaware of the significance of any particular value of any particular register.
04	Slave Device Failure	An unrecoverable error occurred while the Server was attempting to perform the requested action.
05	Acknowledge	Specialized use in conjunction with programming commands. The Server has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the Client. The Client can next issue a poll program complete message to determine if processing is completed.
06	Slave Device Busy	Specialized use in conjunction with programming commands. The Server is engaged in processing a long-duration program command. The Client should retransmit the message later when the Server is free.
08	Memory Parity Error	Specialized use in conjunction with function codes 20 and 21 and reference type 6, to indicate that the extended file area failed to pass a consistency check. The Server attempted to read record file, but detected a parity error in the memory. The Client can retry the request, but service may be required on the Server device.
0a	Gateway Path Unavailable	Specialized use in conjunction with gateways, indicates that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request. Usually means that the gateway is misconfigured or overloaded.
0b	Gateway Target Device Failed To Respond	Specialized use in conjunction with gateways, indicates that no response was obtained from the target device. Usually means that the device is not present on the network.

5.8 Force Multiple Coils (Function Code 15)

Query

This message forces each coil in a consecutive block of coils to a desired ON or OFF state. Any coil that exists within the controller can be forced to either state (ON or OFF). However, because the controller is actively scanning, unless the coils are disabled, the controller can also alter the state of the coil. Coils are numbered from zero (coil 00001 = zero, coil 00002 = one, and so on). The desired status of each coil is packed in the data field, one bit for each coil (1= ON, 0= OFF). The use of Server address 0 (Broadcast Mode) will force all attached Servers to modify the desired coils.

Note: Functions 5, 6, 15, and 16 are the only messages (other than Loopback Diagnostic Test) that will be recognized as valid for broadcast.

The following example forces 10 coils starting at address 20 (13 HEX). The two data fields, CD =1100 and 00 = 0000 000, indicate that coils 27, 26, 23, 22, and 20 are to be forced on.

Adr	Func	Hi Add	Lo Add	Quantity	Byte Cnt	Data Coil Status 20 to 27	Data Coil Status 28 to 29	Error Check Field
11	0F	00	13	00	0A	02	CD	00 CRC

Response

The normal response will be an echo of the Server address, function code, starting address, and quantity of coils forced.

Adr	Func	Hi Addr	Lo Addr	Quantity	Error Check Field
11	0F	00	13	00	0A CRC

The writing of coils via Modbus function 15 will be accomplished regardless of whether the addressed coils are disabled or not.

Coils that are unprogrammed in the controller logic program are not automatically cleared upon power up. Thus, if such a coil is set ON by function code 15 and (even months later) an output is connected to that coil, the output will be hot.

5.9 Preset Multiple Registers (Function Code 16)

Query

Holding registers existing within the controller can have their contents changed by this message (a maximum of 60 registers). However, because the controller is actively scanning, it also can alter the content of any holding register at any time. The values are provided in binary up to the maximum capacity of the controller (16-bit for the 184/384 and 584); unused high order bits must be set to zero.

Note: Function codes 5, 6, 15, and 16 are the only messages that will be recognized as valid for broadcast.

Adr	Func	Hi Add	Lo Add	Quantity	Byte Cnt	Hi Data	Lo Data	Hi Data	Lo Data	Error Check Field	
11	10	00	87	00	02	04	00	0A	01	02	CRC

Response

The normal response to a function 16 query is to echo the address, function code, starting address and number of registers to be loaded.

Adr	Func	Hi Addr	Lo Addr	Quantity	Error Check Field	
11	10	00	87	00	02	56

6 Support, Service & Warranty

In This Chapter

- ❖ Contacting Technical Support 59
- ❖ Return Material Authorization (RMA) Policies and Conditions..... 61
- ❖ LIMITED WARRANTY..... 63

Contacting Technical Support

ProSoft Technology, Inc. (ProSoft) is committed to providing the most efficient and effective support possible. Before calling, please gather the following information to assist in expediting this process:

- 1 Product Version Number
- 2 System architecture
- 3 Network details

If the issue is hardware related, we will also need information regarding:

- 1 Module configuration and associated ladder files, if any
- 2 Module operation and any unusual behavior
- 3 Configuration/Debug status information
- 4 LED patterns
- 5 Details about the serial, Ethernet or fieldbus devices interfaced to the module, if any.

Note: For technical support calls within the United States, an after-hours answering system allows 24-hour/7-days-a-week pager access to one of our qualified Technical and/or Application Support Engineers. Detailed contact information for all our worldwide locations is available on the following page.

Internet	Web Site: www.prosoft-technology.com/support E-mail address: support@prosoft-technology.com
Asia Pacific (location in Malaysia)	Tel: +603.7724.2080, E-mail: asiapc@prosoft-technology.com Languages spoken include: Chinese, English
Asia Pacific (location in China)	Tel: +86.21.5187.7337 x888, E-mail: asiapc@prosoft-technology.com Languages spoken include: Chinese, English
Europe (location in Toulouse, France)	Tel: +33 (0) 5.34.36.87.20, E-mail: support.EMEA@prosoft-technology.com Languages spoken include: French, English
Europe (location in Dubai, UAE)	Tel: +971-4-214-6911, E-mail: mea@prosoft-technology.com Languages spoken include: English, Hindi
North America (location in California)	Tel: +1.661.716.5100, E-mail: support@prosoft-technology.com Languages spoken include: English, Spanish
Latin America (Oficina Regional)	Tel: +1-281-2989109, E-Mail: latinam@prosoft-technology.com Languages spoken include: Spanish, English
Latin America (location in Puebla, Mexico)	Tel: +52-222-3-99-6565, E-mail: soporte@prosoft-technology.com Languages spoken include: Spanish
Brasil (location in Sao Paulo)	Tel: +55-11-5083-3776, E-mail: brasil@prosoft-technology.com Languages spoken include: Portuguese, English

6.1 Return Material Authorization (RMA) Policies and Conditions

The following Return Material Authorization (RMA) Policies and Conditions (collectively, "RMA Policies") apply to any returned product. These RMA Policies are subject to change by ProSoft Technology, Inc., without notice. For warranty information, see Limited Warranty (page 63). In the event of any inconsistency between the RMA Policies and the Warranty, the Warranty shall govern.

6.1.1 Returning Any Product

- a) In order to return a Product for repair, exchange, or otherwise, the Customer must obtain a Return Material Authorization (RMA) number from ProSoft Technology and comply with ProSoft Technology shipping instructions.
- b) In the event that the Customer experiences a problem with the Product for any reason, Customer should contact ProSoft Technical Support at one of the telephone numbers listed above (page 59). A Technical Support Engineer will request that you perform several tests in an attempt to isolate the problem. If after completing these tests, the Product is found to be the source of the problem, we will issue an RMA.
- c) All returned Products must be shipped freight prepaid, in the original shipping container or equivalent, to the location specified by ProSoft Technology, and be accompanied by proof of purchase and receipt date. The RMA number is to be prominently marked on the outside of the shipping box. Customer agrees to insure the Product or assume the risk of loss or damage in transit. Products shipped to ProSoft Technology using a shipment method other than that specified by ProSoft Technology, or shipped without an RMA number will be returned to the Customer, freight collect. Contact ProSoft Technical Support for further information.
- d) A 10% restocking fee applies to all warranty credit returns, whereby a Customer has an application change, ordered too many, does not need, etc. Returns for credit require that all accessory parts included in the original box (i.e.; antennas, cables) be returned. Failure to return these items will result in a deduction from the total credit due for each missing item.

6.1.2 Returning Units Under Warranty

A Technical Support Engineer must approve the return of Product under ProSoft Technology's Warranty:

- a) A replacement module will be shipped and invoiced. A purchase order will be required.
- b) Credit for a product under warranty will be issued upon receipt of authorized product by ProSoft Technology at designated location referenced on the Return Material Authorization
 - i. If a defect is found and is determined to be customer generated, or if the defect is otherwise not covered by ProSoft Technology's warranty, there will be no credit given. Customer will be contacted and can request module be returned at their expense;
 - ii. If defect is customer generated and is repairable, customer can authorize ProSoft Technology to repair the unit by providing a purchase order for 30% of the current list price plus freight charges, duties and taxes as applicable.

6.1.3 Returning Units Out of Warranty

- a) Customer sends unit in for evaluation to location specified by ProSoft Technology, freight prepaid.
- b) If no defect is found, Customer will be charged the equivalent of \$100 USD, plus freight charges, duties and taxes as applicable. A new purchase order will be required.
- c) If unit is repaired, charge to Customer will be 30% of current list price (USD) plus freight charges, duties and taxes as applicable. A new purchase order will be required or authorization to use the purchase order submitted for evaluation fee.

The following is a list of non-repairable units:

ScanPort Adapters

- o 1500 - All
- o 1550 - Can be repaired only if defect is the power supply
- o 1560 - Can be repaired only if defect is the power supply

inRAX Modules

- o 3150 - All
- o 3170 - All
- o 3250
- o 3300
- o 3350
- o 3600 - All
- o 3700
- o 3750
- o 3800-MNET

ProLinx Standalone Gateways

- o 4xxx - All (No hardware available to do repairs)

6.2 LIMITED WARRANTY

This Limited Warranty ("Warranty") governs all sales of hardware, software, and other products (collectively, "Product") manufactured and/or offered for sale by ProSoft Technology, Incorporated (ProSoft), and all related services provided by ProSoft, including maintenance, repair, warranty exchange, and service programs (collectively, "Services"). By purchasing or using the Product or Services, the individual or entity purchasing or using the Product or Services ("Customer") agrees to all of the terms and provisions (collectively, the "Terms") of this Limited Warranty. All sales of software or other intellectual property are, in addition, subject to any license agreement accompanying such software or other intellectual property.

6.2.1 What Is Covered By This Warranty

- a) *Warranty On New Products:* ProSoft warrants, to the original purchaser, that the Product that is the subject of the sale will (1) conform to and perform in accordance with published specifications prepared, approved and issued by ProSoft, and (2) will be free from defects in material or workmanship; provided these warranties only cover Product that is sold as new. This Warranty expires three (3) years from the date of shipment for Product purchased **on or after** January 1st, 2008, or one (1) year from the date of shipment for Product purchased **before** January 1st, 2008 (the "Warranty Period"). If the Customer discovers within the Warranty Period a failure of the Product to conform to specifications, or a defect in material or workmanship of the Product, the Customer must promptly notify ProSoft by fax, email or telephone. In no event may that notification be received by ProSoft later than 39 months from date of original shipment. Within a reasonable time after notification, ProSoft will correct any failure of the Product to conform to specifications or any defect in material or workmanship of the Product, with either new or remanufactured replacement parts. ProSoft reserves the right, and at its sole discretion, may replace unrepairable units with new or remanufactured equipment. All replacement units will be covered under warranty for the 3 year period commencing from the date of original equipment purchase, not the date of shipment of the replacement unit. Such repair, including both parts and labor, will be performed at ProSoft's expense. All warranty service will be performed at service centers designated by ProSoft.
- b) *Warranty On Services:* Materials and labor performed by ProSoft to repair a verified malfunction or defect are warranted in the terms specified above for new Product, provided said warranty will be for the period remaining on the original new equipment warranty or, if the original warranty is no longer in effect, for a period of 90 days from the date of repair.

6.2.2 What Is Not Covered By This Warranty

- a) ProSoft makes no representation or warranty, expressed or implied, that the operation of software purchased from ProSoft will be uninterrupted or error free or that the functions contained in the software will meet or satisfy the purchaser's intended use or requirements; the Customer assumes complete responsibility for decisions made or actions taken based on information obtained using ProSoft software.
- b) This Warranty does not cover the failure of the Product to perform specified functions, or any other non-conformance, defects, losses or damages caused by or attributable to any of the following: (i) shipping; (ii) improper installation or other failure of Customer to adhere to ProSoft's specifications or instructions; (iii) unauthorized repair or maintenance; (iv) attachments, equipment, options, parts, software, or user-created programming (including, but not limited to, programs developed with any IEC 61131-3, "C" or any variant of "C" programming languages) not furnished by ProSoft; (v) use of the Product for purposes other than those for which it was designed; (vi) any other abuse, misapplication, neglect or misuse by the Customer; (vii) accident, improper testing or causes external to the Product such as, but not limited to, exposure to extremes of temperature or humidity, power failure or power surges; or (viii) disasters such as fire, flood, earthquake, wind and lightning.
- c) The information in this Agreement is subject to change without notice. ProSoft shall not be liable for technical or editorial errors or omissions made herein; nor for incidental or consequential damages resulting from the furnishing, performance or use of this material. The user guide included with your original product purchase from ProSoft contains information protected by copyright. No part of the guide may be duplicated or reproduced in any form without prior written consent from ProSoft.

6.2.3 Disclaimer Regarding High Risk Activities

Product manufactured or supplied by ProSoft is not fault tolerant and is not designed, manufactured or intended for use in hazardous environments requiring fail-safe performance including and without limitation: the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of the product could lead directly or indirectly to death, personal injury or severe physical or environmental damage (collectively, "high risk activities"). ProSoft specifically disclaims any express or implied warranty of fitness for high risk activities.

6.2.4 Intellectual Property Indemnity

Buyer shall indemnify and hold harmless ProSoft and its employees from and against all liabilities, losses, claims, costs and expenses (including attorney's fees and expenses) related to any claim, investigation, litigation or proceeding (whether or not ProSoft is a party) which arises or is alleged to arise from Buyer's acts or omissions under these Terms or in any way with respect to the Products. Without limiting the foregoing, Buyer (at its own expense) shall indemnify and hold harmless ProSoft and defend or settle any action brought against such Companies to the extent based on a claim that any Product made to Buyer specifications infringed intellectual property rights of another party. ProSoft makes no warranty that the product is or will be delivered free of any person's claiming of patent, trademark, or similar infringement. The Buyer assumes all risks (including the risk of suit) that the product or any use of the product will infringe existing or subsequently issued patents, trademarks, or copyrights.

- a) Any documentation included with Product purchased from ProSoft is protected by copyright and may not be duplicated or reproduced in any form without prior written consent from ProSoft.
- b) ProSoft's technical specifications and documentation that are included with the Product are subject to editing and modification without notice.
- c) Transfer of title shall not operate to convey to Customer any right to make, or have made, any Product supplied by ProSoft.
- d) Customer is granted no right or license to use any software or other intellectual property in any manner or for any purpose not expressly permitted by any license agreement accompanying such software or other intellectual property.
- e) Customer agrees that it shall not, and shall not authorize others to, copy software provided by ProSoft (except as expressly permitted in any license agreement accompanying such software); transfer software to a third party separately from the Product; modify, alter, translate, decode, decompile, disassemble, reverse-engineer or otherwise attempt to derive the source code of the software or create derivative works based on the software; export the software or underlying technology in contravention of applicable US and international export laws and regulations; or use the software other than as authorized in connection with use of Product.
- f) **Additional Restrictions Relating To Software And Other Intellectual Property**

In addition to compliance with the Terms of this Warranty, Customers purchasing software or other intellectual property shall comply with any license agreement accompanying such software or other intellectual property. Failure to do so may void this Warranty with respect to such software and/or other intellectual property.

6.2.5 Disclaimer of all Other Warranties

The Warranty set forth in What Is Covered By This Warranty (page 63) are in lieu of all other warranties, express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

6.2.6 Limitation of Remedies **

In no event will ProSoft or its Dealer be liable for any special, incidental or consequential damages based on breach of warranty, breach of contract, negligence, strict tort or any other legal theory. Damages that ProSoft or its Dealer will not be responsible for include, but are not limited to: Loss of profits; loss of savings or revenue; loss of use of the product or any associated equipment; loss of data; cost of capital; cost of any substitute equipment, facilities, or services; downtime; the claims of third parties including, customers of the Purchaser; and, injury to property.

** Some areas do not allow time limitations on an implied warranty, or allow the exclusion or limitation of incidental or consequential damages. In such areas, the above limitations may not apply. This Warranty gives you specific legal rights, and you may also have other rights which vary from place to place.

6.2.7 Time Limit for Bringing Suit

Any action for breach of warranty must be commenced within 39 months following shipment of the Product.

6.2.8 No Other Warranties

Unless modified in writing and signed by both parties, this Warranty is understood to be the complete and exclusive agreement between the parties, suspending all oral or written prior agreements and all other communications between the parties relating to the subject matter of this Warranty, including statements made by salesperson. No employee of ProSoft or any other party is authorized to make any warranty in addition to those made in this Warranty. The Customer is warned, therefore, to check this Warranty carefully to see that it correctly reflects those terms that are important to the Customer.

6.2.9 Allocation of Risks

This Warranty allocates the risk of product failure between ProSoft and the Customer. This allocation is recognized by both parties and is reflected in the price of the goods. The Customer acknowledges that it has read this Warranty, understands it, and is bound by its Terms.

6.2.10 Controlling Law and Severability

This Warranty shall be governed by and construed in accordance with the laws of the United States and the domestic laws of the State of California, without reference to its conflicts of law provisions. If for any reason a court of competent jurisdiction finds any provisions of this Warranty, or a portion thereof, to be unenforceable, that provision shall be enforced to the maximum extent permissible and the remainder of this Warranty shall remain in full force and effect. Any cause of action with respect to the Product or Services must be instituted in a court of competent jurisdiction in the State of California.

Index

I

[MNET CLIENT x] • 20

O

00 Return Query Data • 48
01 Restart Communications Option • 49
02 Return Diagnostic Register • 49
03 Change ASCII Input Delimiter • 49
04 Force Listen Only Mode • 49

1

10 (0A Hex) Clear Counters and Diagnostic Register • 50
11 (0B Hex) Return Bus Message Count • 50
12 (0C Hex) Return Bus Communication Error Count • 50
13 (0D Hex) Return Bus Exception Error Count • 50
14 (0E Hex) Return Slave Message Count • 50
15 (0F Hex) Return Slave No Response Count • 50
16 (10 Hex) Return Slave NAK Count • 51
17 (11 Hex) Return Slave Busy Count • 51
18 (12 Hex) Return Bus Character Overrun Count • 51

2

20 (14 Hex) Clear Overrun Counter and Flag • 51

A

Allocation of Risks • 66
ARP Timeout • 20

B

Bit Input Offset • 19

C

Client Command Errors • 37
Client Error/Status Data • 31
Command Entry Formats • 24
Command List Entry Errors • 38
Command List Overview • 22
Commands Supported by the Module • 23
Configure the Gateway • 17
Configuring Module Parameters • 17
Contacting Technical Support • 59, 61
Controlling Law and Severability • 66

D

Diagnostics (Function Code 08) • 48
Disclaimer of all Other Warranties • 65
Disclaimer Regarding High Risk Activities • 64

E

Enable • 25
Ethernet Configuration • 29
Ethernet LED Indicators • 39
Example and State Diagram • 52

F

Float Flag • 18, 21
Float Offset • 18, 21
Float Start • 18, 21
Force Multiple Coils (Function Code 15) • 56
Force Single Coil (Function Code 05) • 46
Functional Overview • 9

G

Gateway Internal Database • 11

H

Holding Register Offset • 19
How to Contact Us • 2

I

Important Installation Instructions • 3
Installing ProSoft Configuration Builder Software • 15
Intellectual Property Indemnity • 65
Internal Address • 25

L

Limitation of Remedies ** • 66
LIMITED WARRANTY • 61, 63

M

Markings • 4
MB Address in Device • 28
MBAP Port Override • 20
Minimum Command Delay • 21
MNET Client [0] • 33
MNET Client [1] • 33
MNET Client [2] • 34
MNET Client [3] • 34
MNET Client [4] • 34
MNET Client [5] • 35
MNET Client [6] • 35
MNET Client [7] • 35
MNET Client [8] • 36
MNET Client [9] • 36
MNET Client Error/Status Data • 33
MNET Client Specific Errors • 38
MNET Client x Commands • 22
MNET Error and Status Data • 31
MNET Server (Port 2000) • 31
MNET Server (Port 502) • 32
MNET Servers • 18
Modbus Exception Codes • 55
Modbus Exception Responses • 50, 53
Modbus Function • 27
Modbus Protocol Specification • 41

Modbus TCP/IP (MNET) Port • 10
Modbus TCP/IP Client Access to Database • 11
Modbus TCP/IP Client Protocol Configuration • 15
Modbus TCP/IP Specifications • 13
Module Communication Error Codes • 38

N

No Other Warranties • 66
Node IP Address • 26, 27

O

Output Offset • 19

P

Pinouts • 3
Poll Interval • 25
Preset Multiple Registers (Function Code 16) • 57
Preset Single Register (Function Code 06) • 47
Printing a Configuration File • 17
ProLinx Gateways with Ethernet Ports • 3
ProLinx® Products Warnings • 3
ProSoft Technology® Product Documentation • 2
Protocol Functional Specifications • 13

R

Read Coil Status (Function Code 01) • 42
Read Holding Registers (Function Code 03) • 44
Read Input Registers (Function Code 04) • 45
Read Input Status (Function Code 02) • 43
Reg Count • 26
Response Timeout • 21
Retry Count • 21
Return Material Authorization (RMA) Policies and
Conditions • 61
Returning Any Product • 61
Returning Units Out of Warranty • 62
Returning Units Under Warranty • 62

S

Service Port • 27
Slave Address • 27
Standard Modbus Exception Code Errors • 37
Sub-function Codes Supported • 48
Support, Service & Warranty • 59
Swap Code • 26

T

Time Limit for Bringing Suit • 66
To order a ProLinx Plus gateway with the -WEB option
• 3
To upgrade a previously purchased Series C model • 3

U

Using the Online Help • 16

W

What Is Covered By This Warranty • 63, 65

What Is Not Covered By This Warranty • 64
Word Input Offset • 19

Y

Your Feedback Please • 2