ProSoft
TECHNOLOGY

Where Automation Connects.

ProLinx®
DNPSNET

ProLinx Gateway

DNP 3.0 Server over Ethernet

May 12, 2014

PROTOCOL MANUAL

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 © 2014 ProSoft Technology, Inc., All rights reserved.

DNPSNET Protocol Manual

May 12, 2014

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 DVD 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.

Contents

Guide to the DNPSNET Protocol Manual   7

1   Start Here   9

2   Diagnostics and Troubleshooting   27

3   Reference   35

Guide to the DNPSNET Protocol Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 8)	This section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Diagnostic and Troubleshooting	→	Diagnostics and Troubleshooting (page 27)	This section describes Diagnostic and Troubleshooting procedures.
Reference Product Specifications	→	Reference (page 35) Product Specifications (page 36)	These sections contain general references associated with this product and its Specifications..
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 57) Index	This section contains Support, Service and Warranty information. Index of chapters.

1   Start Here

In This Chapter

For most applications, the installation and configuration steps described in this section will work without additional programming. ProSoft Technology strongly recommends that you complete the steps in this chapter before developing a custom application.

1.1   ProLinx Reference Guide

The *ProLinx Reference Guide* on the ProSoft Solutions DVD provides detailed information on the entire range of ProLinx gateways. If you have any questions that are not answered in the DNPSNET Protocol Manual, please refer to the *ProLinx Reference Guide.*

1.2 Installing ProSoft Configuration Builder Software

You must install the *ProSoft Configuration Builder (PCB)* software to configure the module. 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 DVD*, included in the package with your module.

To install ProSoft Configuration Builder from the Product DVD

1   Insert the *ProSoft Solutions Product DVD* into the DVD 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 module.
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 DVD. You may wish to copy the files from the *Utilities* folder on the DVD to a convenient location on your hard drive.

1.2.1 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.**

1.3   Using ProSoft Configuration Builder

ProSoft Configuration Builder (PCB) provides a convenient way to manage module configuration files customized to meet your application needs. *PCB* is not only a powerful solution for new configuration files, but also allows you to import information from previously installed (known working) configurations to new projects.

1.3.1 Setting Up the Project

To begin, start ProSoft Configuration Builder (PCB). If you have used other Windows configuration tools before, you will find the screen layout familiar. ProSoft Configuration Builder's (PCB's) window consists of a tree view on the left, an information pane, and a configuration pane on the right side of the window. When you first start PCB, the tree view consists of folders for Default Project and Default Location, with a Default Module in the Default Location folder. The following illustration shows the PCB window with a new project.
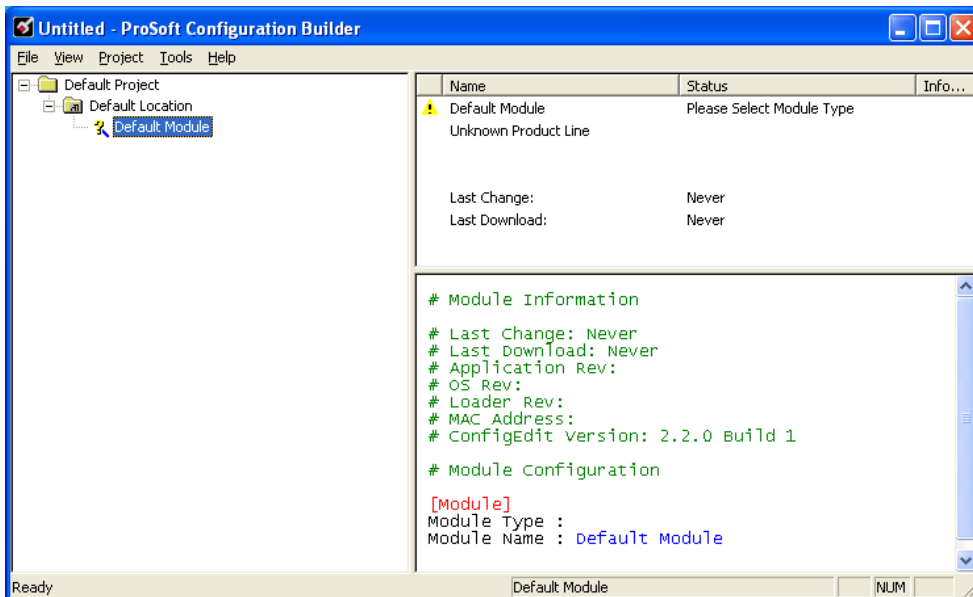
To add the gateway to the project

1 Use the mouse to select **DEFAULT MODULE** in the tree view, and then click the right mouse button to open a shortcut menu.

2 On the shortcut menu, choose **CHOOSE MODULE TYPE**. This action opens the *Choose Module Type* dialog box.
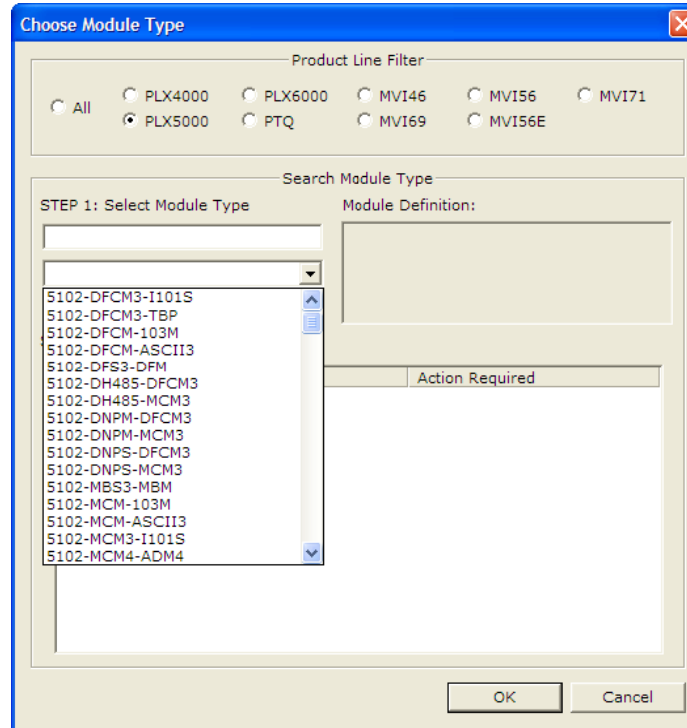
3 In the *Product Line Filter* area of the dialog box, select the appropriate product type radio button.

4 In the *STEP 1: Select Module Type* dropdown list, select the model number that matches your module, and then click **OK** to save your settings and return to the *PCB Main* window.

1.3.2 Renaming PCB Objects

Notice that the contents of the information pane and the configuration pane changed when you added the module to the project.
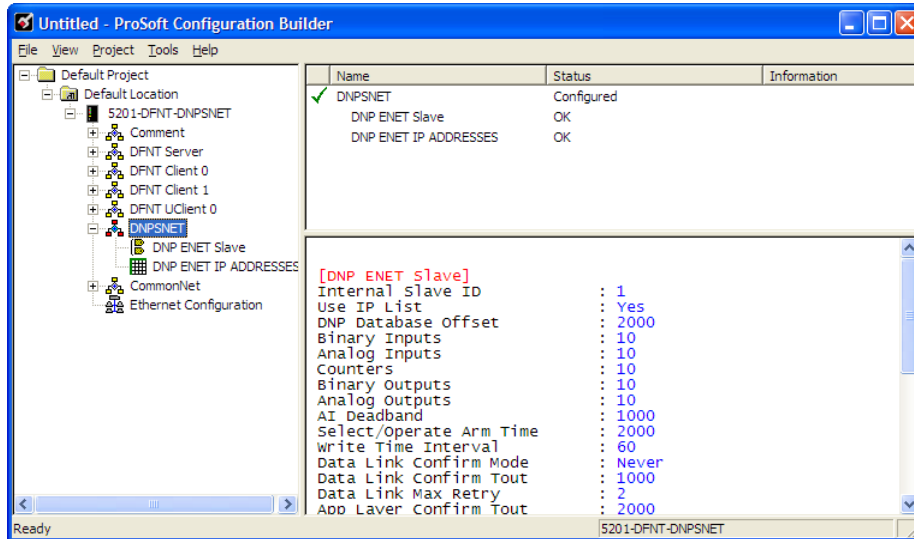
At this time, you may wish to rename the *Default Project* and *Default Location* folders in the tree view.

1 Select the object, and then click the right mouse button to open a shortcut menu. From the shortcut menu, choose **RENAME.**
2 Type the name to assign to the object.
3 Click *away* from the object to save the new name.

Configuring Module Parameters

1 Click the **[+]** sign next to the module icon to expand module information.
2 Click the **[+]** sign next to any ⛁ icon to view module 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.

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**.

1.4   DNPSNET Protocol Configuration

The following is excerpted from a configuration file showing typical examples of the DNPSNET Port of a CFG file for a DNPSNET port. Shipped with each unit (or available from the web) is a default configuration file that can easily form the basis for a working solution. This file can either be downloaded from the ProSoft web site at www.prosoft-technology.com, or transferred from the module.

Refer to the *ProLinx Reference Guide* for instructions on transferring the configuration file to or from the module.

1.4.1   [DNP ENET Slave]

This section provides information required to configure a slave application with the module. Most entries contained within this section are self explanatory with the possible exception of the Use IP List directive. This directive instructs the module to verify the address of the received message and ignore the message if it is not on our list of acceptable clients.

Internal Slave ID

0 to **65534**

This is the DNP address for the module. All messages with this address received from the master will be processed by the module.

Use IP List

YES or **NO**

This parameter specifies if the IP address of the host connected to the system will be validated. If the parameter is set to **NO**, any host may connect to the unit. If the parameter is set to **YES**, only hosts in the IP list will be permitted to connect to the module. All other IP addresses will be ignored by the module and the module will issue a RST to the TCP/IP connection.

DNP Database Definition Note: The databases are in the memory of the module in this sequence and are placed directly adjacent to each other. In other words when you change the size of a database you must adjust the transfer commands to accommodate the new location.

DNP Database Offset

0 to 3999

Offset in which to place DNP data.

Binary Inputs

0 to 500 words

This parameter specifies the number of digital input points to configure in the DNP slave device based on a word count. The valid range is 0 to 500 words.

Analog Inputs

0 to 500 points

This parameter sets the number of analog input points to configure in the DNP slave device. Each point will occupy a one-word area in the module memory.

Counters

0 to 250 points

This parameter sets the number of counter points to configure in the DNP slave device. Each point will occupy a two-word area in the module memory. This number corresponds to the number of frozen counters. The application maps the counters to the frozen counters directly. Valid values are 0 to 250 points.

Binary Outputs

0 to 125 words

Number of digital output points to configure in the DNP slave device based on a word count. Each word stores 16 points. Therefore, if the parameter is set to 2, 32 binary outputs will be defined for the application.

Analog Outputs

0 to 500 points

Number of analog output points to configure in the DNP slave device. Each point will occupy a one word area in the module memory.

AI Deadband

0 to **32767** data units

This value sets the global deadband for all analog input points. When the current value for an analog input point is not within the deadband limit set based on the last event for the point, an event will be generated.

Select/Operate Arm Time

1 to **65535** milliseconds

This parameter sets the time period after select command received in which operate command will be performed. After the select command is received, the operate command will only be honored if it arrives within this period of time. Valid arm timeout values are 1 to 65535 milliseconds. This example shows the value set to 2000 milliseconds.

Write Time Interval

0 to **1440** minutes

This parameter sets the time interval to set the need time IIN bit (0=never), which will cause the master to write the time. Stored in milliseconds in the module memory.

Data Link Confirm Mode

NEVER, **SOMETIMES**, or **ALWAYS**

An IED slave can request acknowledgment from the Master station when sending data. This parameter allows you to set the confirmation mode for your application. The module diagnostic menus use the following codes to indicate which mode has been configured: **0**=**NEVER**, **1**=**SOMETIMES**, **2**=**ALWAYS.**

Data Link Confirm Timeout

1 to **65535** milliseconds

Time period to wait for Master Data Link confirmation of last frame sent. This time is in milliseconds. This parameter is used only if the frame is sent with a confirmation request.

Data Link Max Retry

0 to **255** retries

Maximum number of retries at the Data Link level to obtain a confirmation. If this value is set to 0, retries are disabled at the data link level of the protocol. This parameter is only used if the frame is sent with confirmation requested.

App Layer Confirm Tout

1 to **65535** milliseconds

Event data contained in the last response may be sent again if not confirmed within the millisecond time period set. If application layer confirms are used with data link confirms, ensure that the application layer confirm timeout is set long enough.

Unsolicited Response

YES or **NO**

This parameter is set if the slave unit will send unsolicited response messages. If set to No, the slave will not send unsolicited responses. If set to Yes, the slave will send unsolicited responses.

Class 1 Unsol Resp Min

1 to **255** events

Minimum number of events in Class 1 required before an unsolicited response will be generated.

Class 2 Unsol Resp Min

1 to **255** events

Minimum number of events in Class 2 required before an unsolicited response will be generated.

Class 3 Unsol Resp Min

1 to **255** events

Minimum number of events in Class 3 required before an unsolicited response will be generated.

Unsol Resp Delay

0 to **65535** milliseconds

Maximum number of 1 millisecond intervals to wait after an event occurs before sending an unsolicited response message. If set to 0, only use minimum number of events.

Uresp Master Address

0 to **65535**

DNP destination address where unsolicited response messages are sent.

AI Events with Time

YES or **NO**

This parameter determines if the analog input events generated by the module will include the date and time of the event. If the parameter is set to **NO**, the default is set to no time data. If the parameter is set to **YES**, the default object will include the time of the event.

Time Sync Before Events

YES or **NO**

This parameter determines if events are to be generated by the module before the time synchronization from the master unit. If the parameter is set to **NO**, events will be generated irrespective of the module's time sync status. If the parameter is set to **YES**, events will be generated only if the module's time is synchronized.

Use Trip/Close Single Point

Yes or No

If you set this parameter to Yes, Trip/Close events will function like Pulse On operations. Only one bit will be reserved in the DNP BO database.

If you set this parameter to No, the dual-point relay control database (Trip/Close) is overlaid on the DNP Binary Output database of the module. Each DNP point index sent will have an offset of point index times 2 into the database. The first bit of the dual-point relay control database will correspond to the close relay and the second will correspond to the trip relay.

The bit definitions from control byte of CROB are as follows:

- 00 - Null (single bit control or select of Trip/Close
- 01 - Close relay
- 10 - Trip relay
- 11 - Invalid

If the operate command is used with the Null relay (00), the module will operate on the point as single point control. The following table describes the module's behavior:

Point Index in Command	Point in Database Controlled
0	Bit 0 in BO database
10	Bit 10 in BO database
15	Bit 15 in BO database

If the operate command is used with the close relay selected, the module will operate on the first bit of the two database bits associated with the point. The following table describes the module's behavior when the close relay is selected:

Point Index in Command	Point in Database Controlled
0	Bit 0 in BO database
1	Bit 2 in BO database
10	Bit 20 in BO database
15	Bit 30 in BO database

If the operate command is used with the trip relay selected, the module will operate on the second bit of the two database bits associated with the point. The following table describes the module's behavior when the trip relay is selected:

Point Index in Command	Point in Database Controlled
0	Bit 1 in BO database
1	Bit 3 in BO database
10	Bit 21 in BO database
15	Bit 31 in BO database

It is important to note that the trip and close relays are linked in the module. If a latch-on command is sent to the close relay its bit will be set and the associated trip relay bit will be cleared.

Because the single-point and dual-point control database share the same memory area, caution should be exercised to prevent control of one area by another. This can be accomplished by careful design of the system. The dual-point database could be isolated from the single-point database. For example, DNP point index 0 to 9 could be used for the dual-point database and correspond to bits 0 to 19. The single-point control points would then start at DNP point index 20 which corresponds to bit 20 of the database.

Using this technique, the DNPSNET module will not require any configuration for the new dual-point control, and the module will be backward compatible for current customer applications.

Trip/Close

The DNPSNET module supports Trip/Close functionality for Binary Output points.

This allows Trip/Close commands to be sent to the DNPSNET module, for dual point control. Each DNP Trip/Close command will occupy 2 bits within the module memory.

This does overlap the regular pulse on/off and latch on/off Binary Output database, therefore special consideration must be used to make sure that points are not used twice.

The following table describes the address mapping for the module using Latch and Pulse commands, and Trip/Close functionality.

DNP BO Database Point	BO Latch/Pulse Point	BO Trip/Close Point
0	BO 0	Close BO 0
1	BO 1	Trip BO 0
2	BO 2	Close BO 1
3	BO 3	Trip BO 1
4	BO 4	Close BO 2
5	BO 5	Trip BO 2
100	BO 100	Close BO 50
101	BO 101	Trip BO 50
1000	BO 1000	Close BO 500
1001	BO 1001	Trip BO 500
2046	BO 2046	Close BO 1023
2047	BO 2047	Trip BO 1023

As you can see from the above chart, trip/close requires 2 points within the module's DNP database. A trip is represented by the value of 2#10 for those 2 points, and a close is represented by the value of 2#01 for those same 2 points.

The module can only process 1024 trip/close points, as the database for the DNP BO is limited to 2048 bits total.

1.4.2 [DNP ENET IP ADDRESSES]

This section of the configuration file only applies if the directive labeled *Use IP List* is set to **YES** or **Y**. If *Use IP List* is enabled, the module will refuse to answer a request unless the IP address of the client is listed in this section. This section may contain no more than 10 addresses.
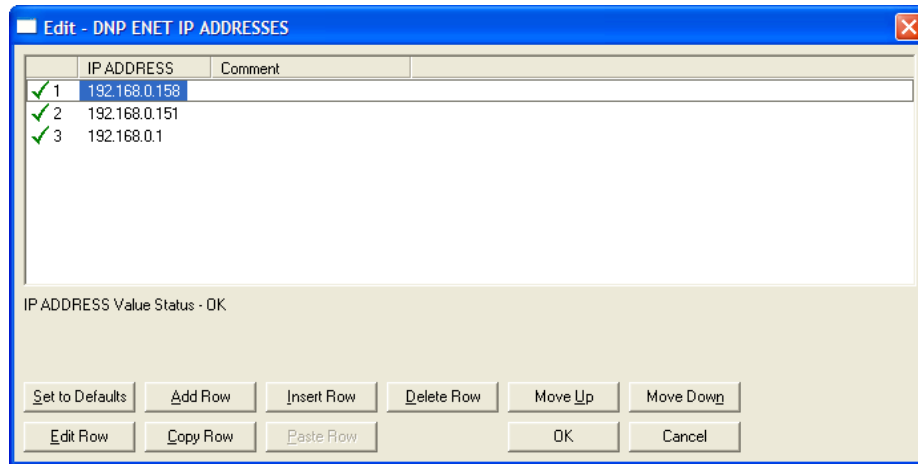
1.4.3 Common Configuration

Using the CommonNet Data Map

The *Data Map* section allows you to copy data between areas in the module's internal database.

You can copy a maximum of 100 registers per *Data Map* command, and you can configure a maximum of 200 separate copy commands.

You can copy data from the error or status tables in upper memory to internal database registers in the *User Data memory* area.

You can rearrange the byte and/or word order during the copy process. For example, by rearranging byte or word order, you can convert floating-point values to the correct format for a different protocol.

You can also use the *Data Map* to condense widely dispersed data into one contiguous data block, making it easier to access.
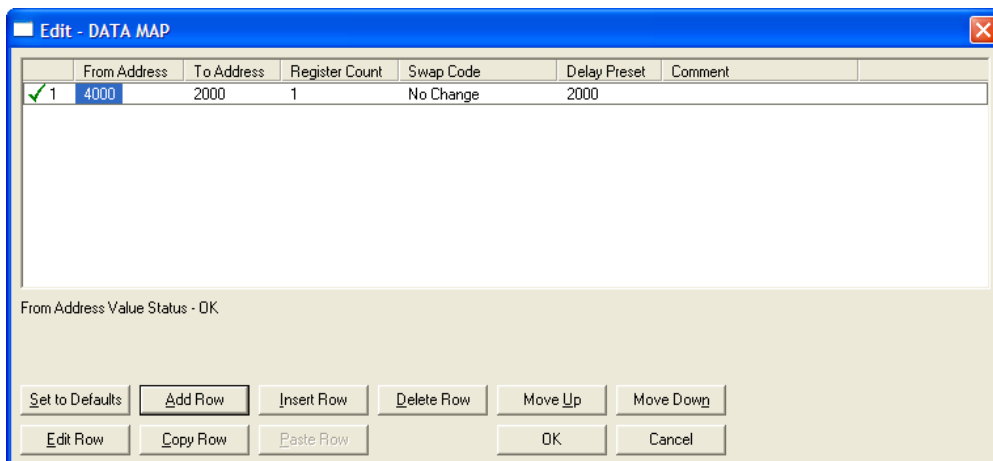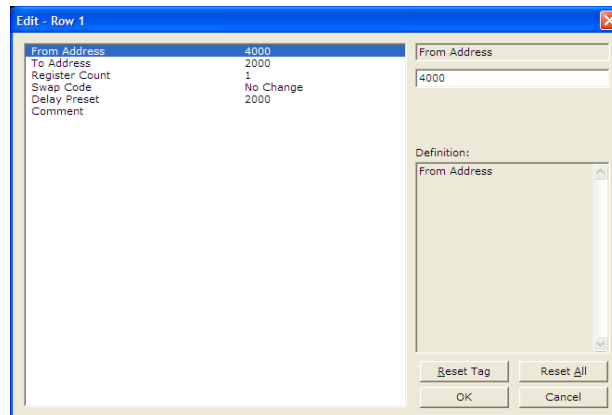
From Address

0 to highest *Status Data* address

This field specifies the beginning internal database register address for the copy operation. This address can be any valid address in the *User Data Area* or the *Status Data Area* of the gateway.

To Address

0 to **3999**

This parameter specifies the beginning destination register address for the copy operation. This address must always be within the *User Data registers* area. Take care to specify a destination address that will not overwrite data that has been stored in memory by one of the communication protocols running on the gateway.

Register Count

1 to **100**

This parameter specifies the number of registers to copy.

Swap Code

NO CHANGE, **WORD SWAP**, **WORD AND BYTE SWAP**, **BYTE SWAP**

You may need to swap the order of the bytes in the registers during the copy process in order to change the alignment of bytes between dissimilar protocols. This parameter is helpful when dealing with floating-point or other multi-register values, as there is no standard method of storage of these data types in slave devices.

The following table defines the values and their associated operations:
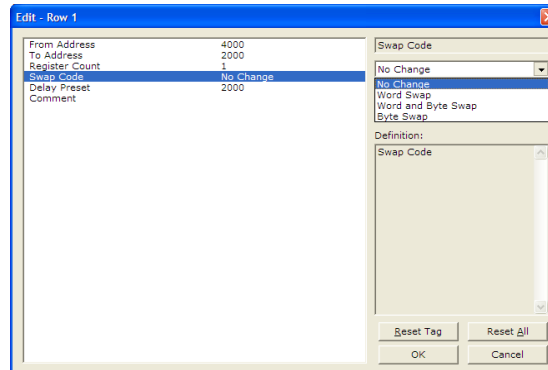
Swap Code	Description
No Swap	No change is made in the byte ordering (1234 = 1234)
Word Swap	The words are swapped (1234=3412)
Word and Byte Swap	The words are swapped, then the bytes in each word are swapped (1234=4321)
Bytes	The bytes in each word are swapped (1234=2143)

Delay Preset

This parameter sets an interval for each *Data Map* copy operation. The value you put for the *Delay Preset* is not a fixed amount of time. It is the number of firmware scans that must transpire between copy operations.

The firmware scan cycle can take a variable amount of time, depending on the level of activity of the protocol drivers running on the ProLinx gateway and the level of activity on the gateway's communication ports. Each firmware scan can take from 1 to several milliseconds to complete. Therefore, *Data Map* copy operations cannot be expected to happen at regular intervals.

If multiple copy operations (several rows in the *Data map* section) happen too frequently or all happen in the same update interval, they could delay the process scan of the gateway protocols, which could result in slow data updates or missed data on communication ports. To avoid these potential problems, you should set the *Delay Preset* to different values for each row in the *Data Map* section and set them to higher, rather than lower, numbers.

For example, *Delay Preset* values below 1000 could begin to cause a noticeable delay in data updates through the communication ports. And you should not set all *Delay Presets* to the same value. Instead, use different values for each row in the Data Map such as 1000, 1001, and 1002 or any other different *Delay Preset* values you like. This will prevent the copies from happening concurrently and prevent possible process scan delays.

1.4.4 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:

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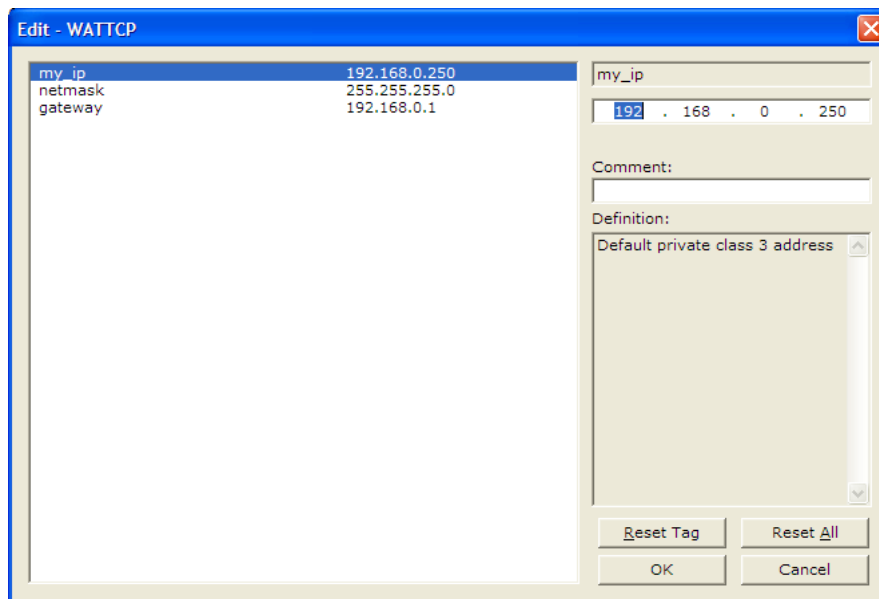
3   Edit the values for *my_ip*, *netmask* (subnet mask) and *gateway* (default gateway).
4   When you are finished editing, click **OK** to save your changes and return to the *ProSoft Configuration Builder* window.

1.4.5 Downloading the Project to the Module Using a Serial COM Port

For the module to use the settings you configured, you must download (copy) the updated *Project* file from your PC to the module.

To download the project file

1   In the tree view in *ProSoft Configuration Builder*, click once to select the module.
2   Right-click the module icon to open a shortcut menu. From the shortcut menu, choose **DOWNLOAD FROM PC TO DEVICE**. The program will scan your PC for a valid com port (this may take a few seconds). When *PCB* has found a valid COM port, the *Download* dialog box will open.
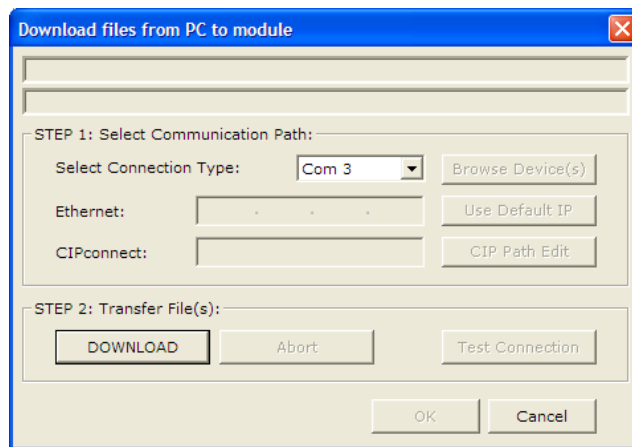
3   Choose the COM port to use from the dropdown list, and then click the **DOWNLOAD** button.

The module will perform a platform check to read and load its new settings. When the platform check is complete, the status bar in the *Download* dialog box will display the message *Module Running*.
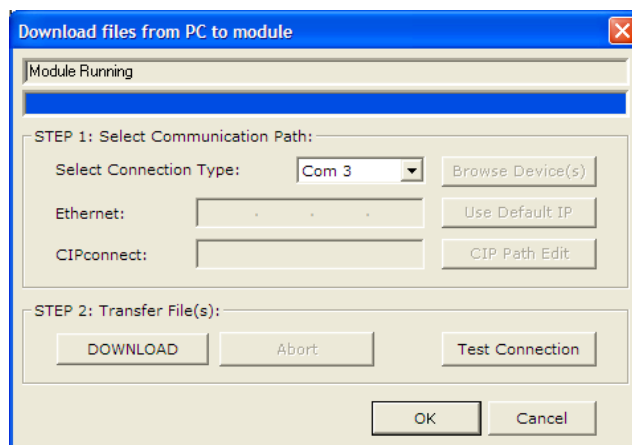
2   Diagnostics and Troubleshooting

In This Chapter

There are two ways to troubleshoot ProLinx gateways:

- Using the LEDs located on the front of the gateway
- Using the Debug port, which provides a view into the gateway's internal database.

2.1   Debug Port Requirements

In order to use the Debug capabilities of any ProLinx Module you will need the following:

- A PC running ProSoft Configuration Builder or HyperTerminal software
- A Null Modem cable
- A Mini-DIN to DB-9M connector
- Configuration and executable files described earlier

2.1.1   Using ProSoft Configuration Builder (PCB) for Diagnostics

The *Configuration and Debug* menu for this module is arranged as a tree structure, with the *Main* menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the *Main* menu.

Because this is a text-based menu system, you enter commands by typing the [command letter] from your computer keyboard in the *Diagnostic* window in *ProSoft Configuration Builder (PCB)*. The module does not respond to mouse movements or clicks. The command executes as soon as you press the **[COMMAND LETTER]** — you do not need to press **[ENTER].** When you type a **[COMMAND LETTER]**, a new screen will be displayed in your terminal application.

Required Hardware

You can connect directly from your PC's serial port to the serial port on the module to view configuration information, perform maintenance, and send or receive configuration files.

ProSoft Technology recommends the following minimum hardware to connect your PC to the module:

- 80486 based processor (Pentium preferred)
- 1 megabyte of memory
- At least one UART hardware-based serial communications port available. USB-based virtual UART systems (USB to serial port adapters) often do not function reliably, especially during binary file transfers, such as when uploading/downloading configuration files or module firmware upgrades.

Using the Diagnostic Window in ProSoft Configuration Builder

Tip: You can have a ProSoft Configuration Builder Diagnostics window open for more than one module at a time.

To connect to the module's Configuration/Debug serial port

1 Start *PCB*, and then select the module to test. Click the right mouse button to open a shortcut menu.
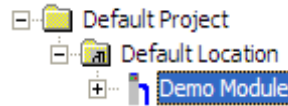
2 On the shortcut menu, choose **DIAGNOSTICS**.
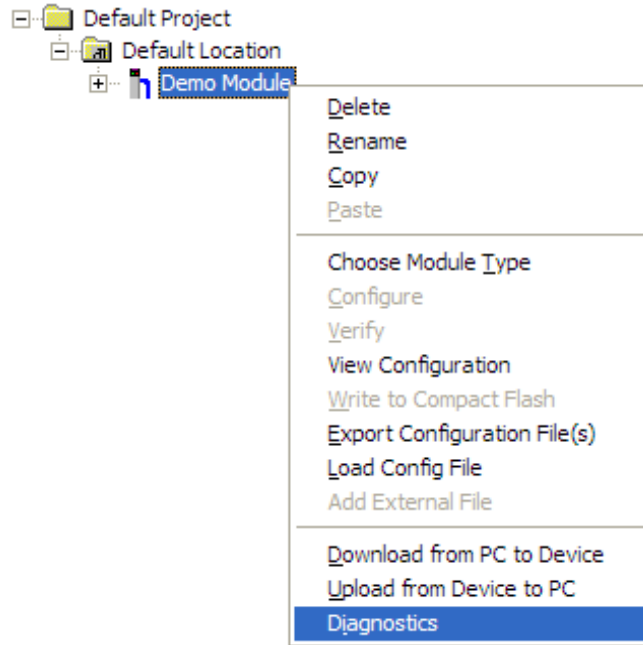
This action opens the *Diagnostics* dialog box.

3 Press **[?]** to open the *Main* menu.
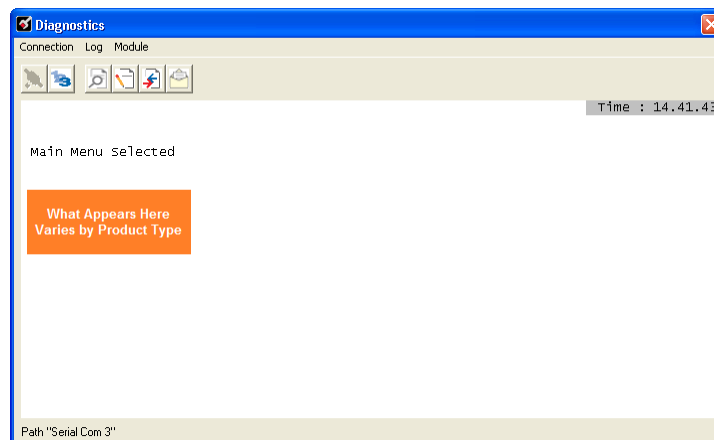
If there is no response from the module, follow these steps:

1 Click to configure the connection. On the *Connection Setup* dialog box, select a valid com port or other connection type supported by the module.
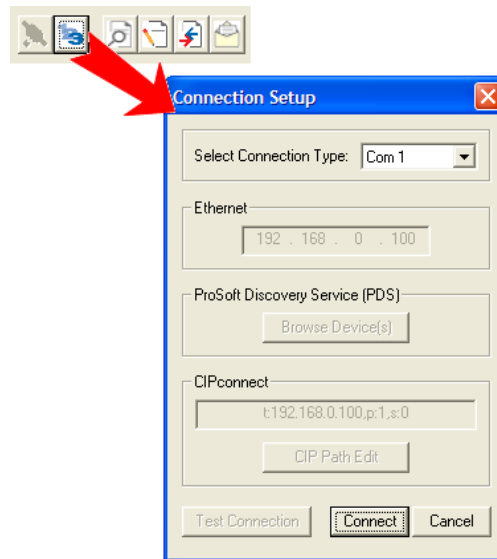
2 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.

3 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

If you are still not able to establish a connection, contact ProSoft Technology for assistance.

Navigation

All of the submenus for this module contain commands to redisplay the menu or return to the previous menu. You can always return from a submenu to the next higher menu by pressing **[M]** on your keyboard.

The organization of the menu structure is represented in simplified form in the following illustration:
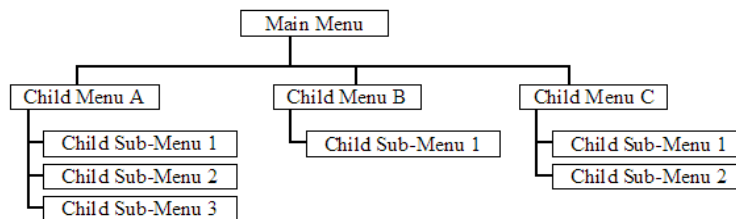
The remainder of this section shows the menus available for this module, and briefly discusses the commands available to you.

Keystrokes

The keyboard commands on these menus are usually not case sensitive. You can enter most commands in lowercase or uppercase letters.

The menus use a few special characters (**?**, **-**, **+**, **@**) that must be entered exactly as shown. Some of these characters will require you to use the **SHIFT**, **CTRL**, or **ALT** keys to enter them correctly. For example, on US English keyboards, enter the **?** command as **SHIFT** and **/**.

Also, take care to distinguish the different uses for uppercase letter "eye" (**I**), lowercase letter "el" (**L**), and the number one (**1**). Likewise, uppercase letter "oh" (**O**) and the number zero (**0**) are not interchangeable. Although these characters look alike on the screen, they perform different actions on the module and may not be used interchangeably.

2.2   LED Indicators

Troubleshooting the operation of the DNPSNET port can be performed using several methods.

The first and quickest is to scan the LEDs on the module to determine the existence and possibly the cause of a problem. This section provides insight into the operation of the Ethernet Port status LEDs. Information on the module's other LEDs can be found in the *ProLinx Reference Guide*.

2.2.1   Base Module LEDs

LED	State	Description
Power	Off	Power is not connected to the power terminals or source is insufficient to properly power the module (minimum required is 800mA at 24 Vdc)
	Green Solid	Power is connected to the power terminals.
Fault	Off	Normal operation.
	Red Solid	A critical error has occurred. Program executable has failed or has been user-terminated and is no longer running. Press Reset p/b or cycle power to clear error. If not, use the Debug procedures described later in this manual.
Cfg	Off	Normal operation.
	Amber Solid	The unit is in configuration mode. The configuration file is currently being downloaded or, after power-up, is being read, the unit is implementing the configuration values, and initializing the hardware. This will occur during power cycle, or after pressing the reset button. It also occurs after a cold/warm boot command is received.
Err	Off	Normal operation.
	Flashing	An error condition has been detected and is occurring on one of the application ports. Check configuration and troubleshoot for communication errors.
	Solid Red	This error flag is cleared at the start of each command attempt (Master/Client) or on each receipt of data (slave/adapter/server); so, if this condition exists, it indicates a large number of errors are occurring in the application (due to bad configuration) or on one or more ports (network communication failures).

2.2.2   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.

2.3   Error and Status Data

The program maintains an error/status data that can be mapped into the module's database using the data map feature available in all ProLinx modules. You can use the error/status table to determine the "health" of the module. This data can also be viewed using the Debug/Configuration port on the module.

2.3.1   Viewing Error and Status Data

The following section lists the register addresses that will contain error and status data. If these registers are mapped into the module's database (registers 0 to 3999), you can use the Database View option to view the contents of these registers using the Debug/ Configuration port. Refer to the *ProLinx Reference Guide* for detailed information on viewing error and status registers using the Debug/ Configuration port. If the data is not mapped to the database, the status data can still be viewed using the Debug/Configuration port. Select the DNPSNET command from the module's main memory and select the information to view from the sub-menu.

2.3.2 DNPS Port - Error and Status

The data area is initialized with zeros whenever the module is initialized. This occurs during a cold-start (power-on), reset (reset push-button pressed) or a warm-boot operation (commanded or loading of new configuration). These data registers are must be mapped into the valid range of registers (0 to 3999) for the module. Use the data mapping feature to accomplish this task. The following table lists the virtual registers assigned to the status and error information maintained by the driver.

Status Register	Description
10000	TCP Receive Frame Count
10001	TCP Transmit Frame Count
10002	UDP Receive Frame Count
10003	UDP Transmit Frame Count
10004	Total Frames Received
10005	Binary Input Event Count
10006	Analog Input Event Count
10007	Bad Function Code Error Count
10008	Object Unknown Error Count
10009	Out of Range Error Count
10010	Buffer Overflow Error Count
10011	Multifragment Receive Error Count
10012	Unsolicited Message Error Count
10013	Program Cycle Counter
10014	Transmit State
10015	TCP Driver State 0 - No session exists on socket 1 - Session established but no connections 2 - Session established and connected 1000 - The server has initiated a close operation on the connection 1001 - The server is waiting for the close on the connection to complete 1002 - The server is issuing an abort (reset) on the connection. The socket is forced closed. 1003 - The server is issuing an abort (reset) on the connection, due to control from the user.
10016	UDP Driver State
10017	Busy State
10018	Application Frame to Send
10019	Transmit Frame to Send
10020	TCP Message Length Remaining
10021	UDP Message Length Remaining
10022	Synchronization Error Count
10023	Overrun Error Count
10024	Length Error Count
10025	Bad CRC Error Count
10026	User Data Overflow Error Count
10027	Sequence Error Count
10028	Address Error Count
10029	No Valid Data

3 Reference

In This Chapter

3.1 Product Specifications

3.1.1 Server Functional Specifications

The DNPSNET module accepts DNP commands to control and monitor the data stored in the DNP databases. These data are passed between the module and the other protocol on the ProLinx module.

3.1.2 DNPSNET Server Specifications

Type	Specifications
General Parameters	
Internal Database	Binary Inputs: 8000 points (500 words)
	Analog Inputs: 500 points of analog input data
	Counters: 250 (500 words)
	Binary Outputs: 2000 points (125 words)
	Analog Outputs: 500 points of analog output
DNP Mode	DNP 3.0 Slave - Level 2
DNP Object Support	See Reference chapter for full Object Definition document
DNP Server	
Service Port	20000 (fixed)
Protocols	TCP (1 connection) and UDP
Node address	0 to 65534 (software selectable)
Status Data	Error codes, counters and port status

3.1.3 Hardware Specifications

| Specification	Description
Power Supply	24 Vdc nominal 18 Vdc to 32 Vdc allowed Positive, Negative, GND Terminals 2.5 mm screwdriver blade
Current Load	500 mA maximum @ 32 Vdc maximum
Operating Temperature	-4°F to 122°F (-20°C to 50°C)
Storage Temperature	-40°F to 185°F (-40°C to 85°C)
Relative Humidity	5% to 95% RH with no condensation
Dimensions (Height x Width x Depth)	Standard: 5.20 in x 2.07 in x 4.52 in (13.2 cm x 5.25 cm x 11.48 cm) Extended: 5.20 in x 2.73 in x 4.52 in (13.2 cm x 6.934 cm x 11.48 cm)
LED Indicators (On all modules)	Power and Hardware Fault Configuration and Application Communication Status Serial Configuration Port Activity and Error
Configuration Serial Port	DB-9M RS-232 only No hardware handshaking
Ethernet Port (Ethernet protocol modules only)	10 Base-T half-duplex RJ45 Connector Link LED and Activity LED indicators Electrical Isolation 1500 Vrms at 50 Hz to 60 Hz for 60 seconds, applied as specified in section 5.3.2 of IEC 60950: 1991 Ethernet Broadcast Storm Resiliency = less than or equal to 5000 [ARP] frames-per-second and less than or equal to 5 minutes duration
Application Serial Port(s) (Serial protocol modules only)	RS-232/422/485 RS-232 handshaking configurable RS-422/485 DB-9 to Screw Terminal Adapter **Note:** The number of serial application ports depends on the module type, and the combination of protocols.
Serial Port Isolation	2500 Vrms port signal isolation per UL 1577 3000 Vdc min. isolation port to ground and port to logic
Shipped with Each Unit	Mini-DIN to DB-9M serial cables 4-foot RS-232 configuration cable 2.5 mm screwdriver DVD (docs and configuration utility) RS-422/485 DB-9 to Screw Terminal Adapter for each serial application port (serial protocols only)

3.2 Cable Connections

The DNPSNET module has the following communication connections on the module:

- One Ethernet port (RJ45 connector)
- One RS-232 Configuration/Debug port (Mini-DIN connector)

3.2.1 RS-232 Configuration/Debug Port

This port is physically an eight-pin, Mini-DIN8F connection. A Mini-DIN8M to DB9M adapter cable is included with the module. This port permits a PC-based terminal emulation program to view configuration and status data in the module and to control the module. Here are the cable pinouts for RS-232 communication on this port.
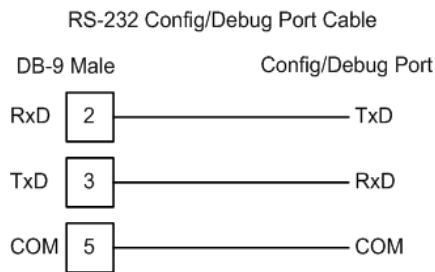
3.2.2 Ethernet Connection

The DNPSNET module has an RJ45 port located on the front of the module, labeled *Ethernet*, for use with the TCP/IP network. The module is connected to the Ethernet network using an Ethernet cable between the module's Ethernet port and an Ethernet switch or hub.

Note: Depending on hardware configuration, you may see more than one RJ45 port on the module. The Ethernet port is labeled *Ethernet*.

Warning: The DNPSNET module is NOT compatible with Power Over Ethernet (IEEE802.3af / IEEE802.3at) networks. Do NOT connect the module to Ethernet devices, hubs, switches or networks that supply AC or DC power over the Ethernet cable. Failure to observe this precaution may result in damage to hardware, or injury to personnel.

Important: The module requires a static (fixed) IP address that is not shared with any other device on the Ethernet network. Obtain a list of suitable IP addresses from your network administrator BEFORE configuring the Ethernet port on this module.

Ethernet Port Configuration - wattcp.cfg

The wattcp.cfg file must be set up properly in order to use a TCP/IP network connection. You can view the current network configuration in *ProSoft Configuration Builder (PCB)*, as shown:
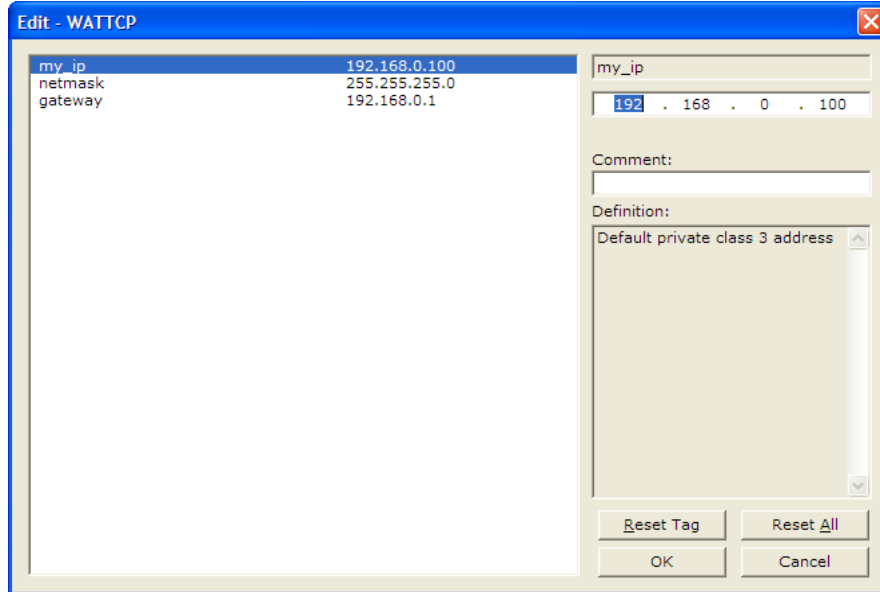
You may also view the network configuration using a PC serial port connection and an ASCII terminal program (like Windows HyperTerminal) by selecting **[@]** (Network Menu) and **[V]** (View) options when connected to the Debug port. For more information on serial port access, see the chapter on Diagnostics and Troubleshooting (page 27).

3.2.3 DB9 to Mini-DIN Adaptor (Cable 09)
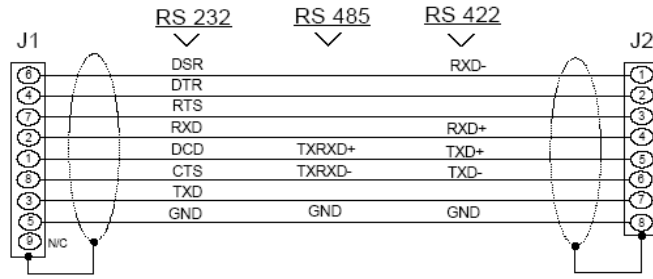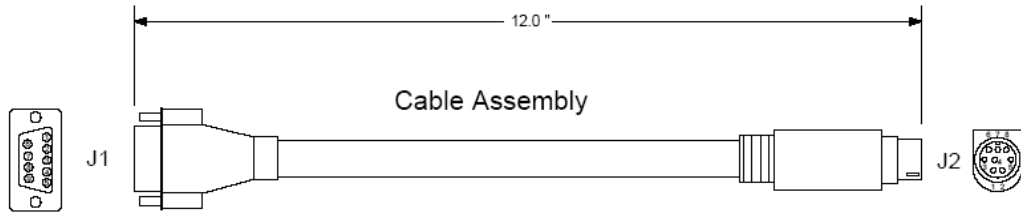
Wiring Diagram

3.3   Functional Overview

The DNPSNET protocol driver exists as a single service port (DNPSNET port 20000) implementation that supports a single TCP port connection and multiple UDP ports on a TCP/IP Ethernet network. The DNPSNET port operates as a server, supporting the DNP 3.0 protocol in a Level 2 implementation using the DNP User Group recommended extension for use on LAN/WAN. This is published in "Transporting DNP V3.00 over Local and Wide Area Networks", December 15, 1998 by the DNP Users Group and is available on the Internet at http://www.dnp.org.

The DNPSNET driver is implemented in ProLinx modules to interface DNP master units with a variety of communication protocols and interfaces. This driver supports DNP version 3.0, subset level 2. The Reference chapter of this documentation contains the Device Profile for the driver. The Reference chapter contains the subset definition for the driver. This document serves as the base for understanding the DNP server driver functionality and configuration. The discussion is general in nature deferring specifics to the individual product documents.

Before attempting to use this or any other DNPSNET protocol device, it is recommended that you have a copy of the DNP Basic 4 documentation and other information available through the DNP User Group. It is very important that these documents be understood for successful application of the protocol in a user's solution. If you are a member of the user group, you can download these documents from the http://www.dnp.org (http://www.dnp.org) Web site.

The DNPSNET driver will respond to requests from a remote DNP client (master) device. All data in the module's database configured as DNP data points is available to the remote master device for read and write requests. Additionally, the driver can send unsolicited event messages to a connected, TCP, client.

3.3.1   DNPSNET Ethernet Port

The ProLinx module supports the DNP 3.0 protocol as a server using the TCP and UDP protocols using service port 20000 (fixed).

The relationship between the port labeling on the front of the ProLinx module and the application is as follows:

Port Label	Function
Debug	Debug/Configuration
Ethernet	DNP Server Port

3.3.2 Module Internal Database

The internal database is central to the functionality of the module. This database is shared between all the ports on the module and is used as a conduit to pass information from one device on one network to one or more devices on another network. This permits data from devices on one communication port/network to be viewed and controlled by devices on another port/network.
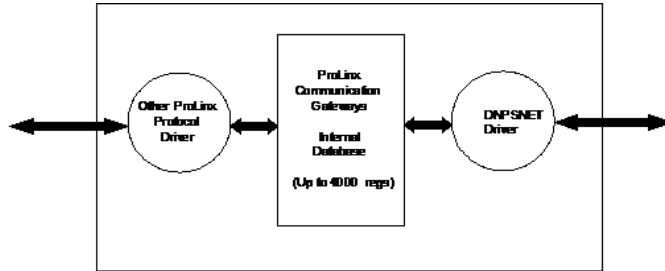
In addition to data from the server port, status and error information generated by the module can also be mapped into the internal database for use by the other protocol driver.

3.3.3 DNPSNET Database Layout

Central to the functionality of the DNPSNET driver is the database. This database is used as the interface between remote DNP devices and the other protocol implemented on a module. The content and structure of the user data area of the database is completely user defined. The following illustration shows the general format of the module's database:

Data Area		Data Size
DNP DATA	BINARY INPUTS	1 WORD PER 16 POINTS
	ANALOG INPUTS	1 WORD PER POINT
	COUNTER DATA	2 WORDS PER POINT
	BINARY OUTPUTS	1 WORD PER 16 POINTS
	ANALOG OUTPUTS	1 WORD PER POINT
	FROZEN COUNTER DATA	2 WORDS PER POINT
	BINARY INPUT EVENTS	11 BYTES PER POINT
	ANALOG INPUT EVENTS	13 BYTES PER POINT
USER DATA	REMAINING DATA AREA	

The first word of the module's database contains the first 16 points of binary input data (if defined). It is important to understand how the data is mapped to the database so that it can be accessed by the other protocol. Each DNP data type has a fixed size. This size is used in conjunction with the number of points configured for the type to determine the size and location in the database. Below is an example of a user database with a defined set of point counts:

Data Area		Registers	Cfg Values
DNP DATA	BINARY INPUTS	0 TO 4	5
	ANALOG INPUTS	5 TO 25	21
	COUNTER DATA	26 TO 35	5
	BINARY OUTPUTS	36 TO 37	2
	ANALOG OUTPUTS	38 TO 47	10
	FROZEN COUNTER DATA	48 TO 57	5
	BINARY INPUT EVENTS	58 TO 607	100
	ANALOG INPUT EVENTS	608 TO 1257	100
USER DATA	REMAINING DATA AREA	1258 TO 9999	

Note that the order of the data types is fixed by the driver (for some interfaces this order may be altered (that is, PROFIBUS Slave)). In order to access the binary input data, registers 0 to 4 are used. To read analog output data passed to the driver from a remote master, registers 38 to 47 are used. Register 38 contains the value for analog output point 0, and register 47 contains the value for analog output point 9.

Data in the frozen counter, binary input event and analog input event areas should not be altered by the other protocol in the module. The DNPSNET driver should only use this data area. The other protocol on the ProLinx module should place data in the binary input, analog input and counter data areas. The remote DNP master reads this data. The remote DNP master will write values to the binary and analog output data areas. This data should be used by the other protocol for control.

3.3.4 DNPSNET Driver Data Flow

The DNP Server Driver allows the module to respond to data read and write commands issued by a master on the DNP network. The following flow chart and associated table describe the flow of data into and out of the module.

Step	Description
1	The DNPSNET driver receives the configuration information from the Flash Disk in the module. This information configures and define the server characteristics.
2	A Host device issues read or write commands to the module's node address. The port driver qualifies the message before accepting it into the module.
3	After the module accepts the command, the data is immediately transferred to or from the internal database in the module. If the command is a read command (binary input, analog input, counter, event, and so on), the data is read out of the database and a response message is built. If the command is a write command (binary output or analog output), the data is written directly into the database and a response message is built.
4	After the data processing has been completed in Step 3, the response is issued to the originating master node.
5	Error/Status data are available in a Status Block that can be placed anywhere in the module's database. This area can be accessed by the other protocol on the module using the correct database offset.

Unsolicited generated by the module will only be sent to a client connected using the TCP protocol. A single TCP connection is supported by the module to prevent confusion as to which client to send the unsolicited messages. Multiple clients can communicate with the module using the UDP protocol but only one master will receive the unsolicited messages. If a client polls for event messages with the UDP protocol they will be sent and removed from the event buffer.

3.4   IIN Bit Definitions

The internal indication bits are stored in a word that follows the function code in all response messages. These bits report status and error information to the master DNP device. Below is a description of the word:

3.4.1   First Byte

Bit	Description
0	All stations message received. Set when a request is received with the destination address set to 0xffff. Cleared after next response. Used to let the master station know that the broadcast was received.
1	Class 1 data available. Set when class 1 data is ready to be sent from the slave to the master. Master should request class 1 data when this bit is set.
2	Class 2 data available. Set when class 2 data is ready to be sent from the slave to the master. Master should request class 2 data when this bit is set.
3	Class 3 data available. Set when class 3 data is ready to be sent from the slave to the master. Master should request class 3 data when this bit is set.
4	Time synchronization required from the master. The master should write the date and time when the bit is set. After receiving the write command, the bit will be cleared.
5	Slave digital outputs are in local control. This bit is not used in this application.
6	Device trouble. When this bit is set, there is a problem with the module.
7	Device restart. This bit is set when the slave either warm or cold boots. It is cleared after a master writes a 0 to the bit.

3.4.2   Second Byte

Bit	Description
0	Bad function code. The function code contained in the master request is not supported for the specified object/variation.
1	Requested object(s) unknown. Object requested by master is not supported by the application.
2	Parameters in the qualifier, range, or data fields are not valid or out of range for the slave.
3	Event buffer(s) or other application buffers have overflowed. This bit is also set if the slave receives a multi-frame message from the master.
4	Request understood but the requested operation is already executing. The slave will never set this bit.
5	Not Used
6	Reserved. Always 0.
7	Reserved. Always 0.

3.5   Device Profile

DNP V3.00
DEVICE PROFILE DOCUMENT

Vendor Name:	ProLinx Communication Gateways, Inc
Device Name:	DNPSNET Driver(VERSION 2.20)

Highest DNP Level Supported:	Device Function:
For Request: L2	Slave (TCP/IP Server (Data Provider))
For Responses: L2	

Notable objects, functions, and/or qualifiers supported in addition to the highest DNP level stated above (see attached table for complete list):

Supports both TCP and UDP protocols as specified in the recommendation document. Supports new function 24 and object 50 variation 3 for time synchronization. Supports list of valid IP addresses for clients to connect (may be disabled by user). Setting of IP list secure. Supports receipt of multiple messages in a single network packet.

The following features are configurable on the module: Time sync before events are generated and default analog input events, Obj32V4 or O32V2, select option.

Counter Freeze with reset will not zero values in the processor. Therefore, this function should not be utilized.

Module will not generate events until Restart IIN bit is cleared by DNP master.

Maximum Data Link Frame Size (octets):	Maximum Application Fragment Size (octets):
Transmitted: 292	Transmitted: 2048
Received: 292	Received: 2048

Maximum Data Link Re-tries:	Maximum Application Layer Re-tries:
Configurable	None

Requires Data Link Layer Confirmation:
Always set to Never as defined in recommendation

Requires Application Layer Confirmation:
When reporting Event Data as a slave unit

Time-outs while waiting for:
Data Link Confirm   : NA

DNP V3.00

	Complete Application Fragment	: Configurable at module start-up
	Application Confirm	: Configurable at module start-up (1 to 65535 milliseconds)
	Complete Application Response	: None

Sends/Executes Control Operations:		
	WRITE Binary Outputs	: Never
	SELECT/OPERATE	: Always
	DIRECT OPERATE	: Always
	DIRECT OPERATE-NO ACK	: Always

	Count > 1	: Always (1 to 65535)
	Pulse On	: Always
	Pulse Off	: Always
	Latch On	: Always
	Latch Off	: Always

	Queue	: Never
	Clear Queue	: Never

Reports Binary Input Change Events when no specific variation requested:	Reports time-tagged Binary Input Change Events when no specific variation requested:
Only time-tagged	Binary Input Change with Time

Sends Unsolicited Responses:	Sends Static Data in Unsolicited Responses:
This is configurable at module start-up. If the number of events for the Binary or Analog Input Events is greater than 0, unsolicited responses are supported. Use the Enable/Disable Unsolicited function code from the DNP master for control.	Never

Default Counter Object/Variation:	Counters Roll Over at:
Object : 20	32 Bits
Variation : 5	

Sends Multi-Fragment Responses: Yes

3.6 DNP Subset Definition

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
1	0	Binary Input: All Variations	1	06			1	Slave will return variation 1 data
	1	Binary Input	1	06	129, 130	00, 01	1	Slave will return this variation
	2	Binary Input with Status			129, 130	00, 01	8	Slave will return Unknown Object to this request
2	0	Binary Input Change: All Variations	1	06, 07, 08			56	Slave will return variation 2 data
	1	Binary Input Change Without Time	1	06, 07, 08	129, 130	17, 28	8	Slave will return this variation
	2	Binary Input Change With Time	1	06, 07, 08	129, 130	17, 28	56	Slave will return this variation
	3	Binary Input Change With Relative Time	1	06, 07, 08	129, 130	17, 28	24	Slave will parse this message and return no data
10	0	Binary Output: All Variations	1	06			8	Slave will return variation 2 data
	1	Binary Output					1	Slave will return Unknown Object to this request
	2	Binary Output Status	1	06	129, 130	00, 01	8	Slave will return this variation
12	0	Control Block: All Variations					88	Slave will use variation 1 control
	1	Control Relay Output Block	3, 4, 5, 6	17, 28	129	Echo of request	88	Slave will respond correctly to this variation
	2	Pattern Control Block					88	Slave will return Unknown Object to this request
	3	Pattern Mask					16	Slave will return Unknown Object to this request
20	0	Binary Counter: All Variations	1, 7, 8, 9, 10	06			32	Slave will return variation 5 data
	1	32-Bit Binary Counter			129, 130	00, 01	40	Slave will return Unknown Object to this request
	2	16-Bit Binary Counter			129, 130	00, 01	24	Slave will return Unknown Object to this request
	3	32-Bit Delta Counter			129, 130	00, 01	40	Slave will return Unknown Object to this request
	4	16-Bit Delta Counter			129, 130	00, 01	24	Slave will return Unknown Object to this request
	5	32-Bit Binary Counter Without Flag	1, 7, 8, 9, 10	06	129, 130	00, 01	32	Slave will return this variation

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	6	16-Bit Binary Counter Without Flag	1, 7, 8, 9, 10	06	129, 130	00, 01	16	Slave will return this variation (counter upper 16-bits removed)
	7	32-Bit Delta Counter Without Flag			129, 130	00, 01	32	Slave will return Unknown Object to this request
	8	16-Bit Delta Counter Without Flag			129, 130	00, 01	16	Slave will return Unknown Object to this request
21	0	Frozen Counter: All Variations	1	06			32	Slave will return variation 9 data
	1	32-Bit Frozen Counter			129, 130	00, 01	40	Slave will return Unknown Object to this request
	2	16-Bit Frozen Counter			129, 130	00, 01	24	Slave will return Unknown Object to this request
	3	32-Bit Frozen Delta Counter					40	Slave will return Unknown Object to this request
	4	16-Bit Frozen Delta Counter					24	Slave will return Unknown Object to this request
	5	32-Bit Frozen Counter With Time Of Freeze					88	Slave will return Unknown Object to this request
	6	16-Bit Frozen Counter With Time Of Freeze					72	Slave will return Unknown Object to this request
	7	32-Bit Frozen Delta Counter With Time Of Freeze					88	Slave will return Unknown Object to this request
	8	16-Bit Frozen Delta Counter With Time Of Freeze					72	Slave will return Unknown Object to this request
	9	32-Bit Frozen Counter Without Flag	1	06	129, 130	00, 01	32	Slave will return this variation
	10	16-Bit Frozen Counter Without Flag	1	06	129, 130	00, 01	16	Slave will return this variation (counter upper 16-bits removed)
	11	32-Bit Frozen Delta Counter Without Flag					32	Slave will return Unknown Object to this request
	12	16-Bit Frozen Delta Counter Without Flag					16	Slave will return Unknown Object to this request
22	0	Counter Change Event: All Variations	1	06, 07, 08				Slave will parse this request and return no data
	1	32-Bit Counter Change Event Without Time			129, 130	17, 28	40	Slave will return Unknown Object to this request

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	2	16-Bit Counter Change Event Without Time			129, 130	17, 28	24	Slave will return Unknown Object to this request
	3	32-Bit Delta Counter Change Event Without Time					40	Slave will return Unknown Object to this request
	4	16-Bit Delta Counter Change Event Without Time					24	Slave will return Unknown Object to this request
	5	32-Bit Counter Change Event With Time					88	Slave will return Unknown Object to this request
	6	16-Bit Counter Change Event With Time					72	Slave will return Unknown Object to this request
	7	32-Bit Delta Counter Change Event With Time					88	Slave will return Unknown Object to this request
	8	16-Bit Delta Counter Change Event With Time					72	Slave will return Unknown Object to this request
23	0	Frozen Counter Event: All Variations						Slave will return Unknown Object to this request
	1	32-Bit Frozen Counter Event Without Time					40	Slave will return Unknown Object to this request
	2	16-Bit Frozen Counter Event Without Time					24	Slave will return Unknown Object to this request
	3	32-Bit Frozen Delta Counter Event Without Time					40	Slave will return Unknown Object to this request
	4	16-Bit Frozen Delta Counter Event Without Time					24	Slave will return Unknown Object to this request
	5	32-Bit Frozen Counter Event With Time					88	Slave will return Unknown Object to this request
	6	16-Bit Frozen Counter Event With Time					72	Slave will return Unknown Object to this request
	7	32-Bit Frozen Delta Counter Event With Time					88	Slave will return Unknown Object to this request
	8	16-Bit Frozen Delta Counter Event With Time					72	Slave will return Unknown Object to this request

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
30	0	Analog Input: All Variations	1	06			16	Slave will respond with variation 4 data
	1	32-Bit Analog Input	1	06	129, 130	00, 01	40	Slave will return this variation (Note: Data will only be 16-bit)
	2	16-Bit Analog Input	1	06	129, 130	00, 01	24	Slave will return this variation
	3	32-Bit Analog Input Without Flag	1	06	129, 130	00, 01	32	Slave will return this variation (Note: Data will only be 16-bit)
	4	16-Bit Analog Input Without Flag	1	06	129, 130	00, 01	16	Slave will return this variation
	5	Short Floating Point Analog Input	1	06	129, 130	00, 01	40	Slave will return this variation
	6	Long Floating Point Analog Input	1	06	129, 130	00, 01	72	Slave will return this variation
31	0	Frozen Analog Input: All Variations						Slave will return Unknown Object to this request
	1	32-Bit Frozen Analog Input					40	Slave will return Unknown Object to this request
	2	16-Bit Frozen Analog Input					24	Slave will return Unknown Object to this request
	3	32-Bit Frozen Analog Input With Time To Freeze					88	Slave will return Unknown Object to this request
	4	16-Bit Frozen Analog Input With Time To Freeze					72	Slave will return Unknown Object to this request
	5	32-Bit Frozen Analog Input Without Flag					32	Slave will return Unknown Object to this request
	6	16-Bit Frozen Analog Input Without Flag					16	Slave will return Unknown Object to this request
	7	Short Floating Point Frozen Analog Input					40	Slave will return Unknown Object to this request
	8	Long Floating Point Frozen Analog Input					72	Slave will return Unknown Object to this request
32	0	Analog Change Event: All Variations	1	06, 07, 08			24	Slave will return variation 2 data
	1	32-Bit Analog Change Event Without Time	1	06, 07, 08	129, 130	17, 28	40	Slave will return this variation (Note: Data only 16-bit)
	2	16-Bit Analog Change Event Without Time	1	06, 07, 08	129, 130	17, 28	24	Slave will return this variation

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	3	32-Bit Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	88	Slave will return this variation (Note: Data only 16-bit)
	4	16-Bit Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	72	Slave will return this variation
	5	Short Floating Point Analog Change Event	1	06, 07, 08	129, 130	17, 28	40	Slave will return this variation
	6	Long Floating Point Analog Change Event	1	06, 07, 08	129, 130	17, 28	72	Slave will return this variation
	7	Short Floating Point Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	88	Slave will return this variation
	8	Long Floating Point Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	120	Slave will return this variation
33	0	Frozen Analog Event: All Variations						Slave will return Unknown Object to this request
	1	32-Bit Frozen Analog Event Without Time					40	Slave will return Unknown Object to this request
	2	16-Bit Frozen Analog Event Without Time					24	Slave will return Unknown Object to this request
	3	32-Bit Frozen Analog Event With Time					88	Slave will return Unknown Object to this request
	4	16-Bit Frozen Analog Event With Time					72	Slave will return Unknown Object to this request
	5	Short Floating Point Frozen Analog Event					40	Slave will return Unknown Object to this request
	6	Long Floating Point Frozen Analog Event					72	Slave will return Unknown Object to this request
	7	Short Floating Point Frozen Analog Event With Time					88	Slave will return Unknown Object to this request
	8	Long Floating Point Frozen Analog Event With Time					120	Slave will return Unknown Object to this request
40	0	Analog Output Status: All Variations	1	06			24	Slave will return variation 2 data

OBJECT			REQUEST		RESPONSE			
Obj	**Var**	**Description**	**Func Codes**	**Qual Codes (hex)**	**Func Codes**	**Qual Codes (hex)**	**Data Size (bits)**	**NOTES**
	1	32-Bit Analog Output Status	1	06	129,130	00,01	40	Slave will return this variation but data only 16-bit accuracy
	2	16-Bit Analog Output Status	1	06	129, 130	00, 01	24	Slave will return this variation
	3	Short Floating Point Analog Output Status	1	06	129, 130	00, 01	40	Slave will return this variation
	4	Long Floating Point Analog Output Status	1	06	129, 130	00, 01	72	Slave will return this variation
41	0	Analog Output Block: All Variations					24	Slave will respond to this request using variation 2 data
	1	32-Bit Analog Output Block	3, 4, 5, 6	17, 28	129,130	00,01	40	Slave will respond to this request but data only 16-bit
	2	16-Bit Analog Output Block	3, 4, 5, 6	17, 28	129	Echo of Request	24	Slave will respond to this request
	3	Short Floating Point Analog Output Block	3, 4, 5, 6	17, 28	129	Echo of Request	40	Not supported
	4	Long Floating Point Analog Output Block	3, 4, 5, 6	17, 28	129	Echo of Request	72	Not supported
50	0	Time and Date: All Variations	2	07, With Quant=1			48	Slave will use variation 1
	1	Time and Date	2	07, With Quant=1			48	Slave will respond to this variation
	2	Time and Date With Interval					80	Slave will return Unknown Object to this request
51	0	Time and Date CTO: All Variations						Slave will return Unknown Object to this request
	1	Time and Date CTO			129, 130	07, With Quant=1	48	Slave will return Unknown Object to this request
	2	Unsynchronized Time and Date CTO			129, 130	07, With Quant=1	48	Slave will return Unknown Object to this request
52	0	Time Delay: All Variations						
	1	Time Delay Coarse			129	07, With Quant=1	16	Slave will never return this variation
	2	Time Delay Fine			129	07, With Quant=1	16	Slave will return this variation to functions 0D, 0E, and 17
60	0	Not Defined						Not Defined in DNP
	1	Class 0 Data	1	06				Slave will respond to this variation with all static data

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	2	Class 1 Data	1	06, 07, 08				Slave will respond to this variation (No class 1 data defined in application)
	3	Class 2 Data	1	06, 07, 08				Slave will respond to this variation with all class 2 data (binary input events)
	4	Class 3 Data	1	06, 07, 08				Slave will respond to this variation with all class 3 data (analog input events)
70	0	Not Defined						Not Defined in DNP
	1	File Identifier						Slave will return Unknown Object to this request
80	0	Not Defined						Not Defined in DNP
	1	Internal Indications	2	00, Index=7			24	Slave will respond to this variation
81	0	Not Defined						Not Defined in DNP
	1	Storage Object						
82	0	Not Defined						Not Defined in DNP
	1	Device Profile						
83	0	Not Defined						Not Defined in DNP
	1	Private Registration Object						
	2	Private Registration Objection Descriptor						
90	0	Not Defined						Not Defined in DNP
	1	Application Identifier						
100	0							
	1	Short Floating Point					48	
	2	Long Floating Point					80	
	3	Extended Floating Point					88	
101	0							
	1	Small Packed Binary-Coded Decimal					16	
	2	Medium Packed Binary-Coded Decimal					32	
	3	Large Packed Binary-Coded Decimal					64	
110	0	Not Defined						Not Defined as the variation determines the string length

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	1 to 100	Octet String	1	00, 01, 06, 07, 08, 17, 28	129, 130	00, 01, 07, 08, 17, 28	8 * Var #	The module will return this variation for the points defined in the module. The variation determines the returned string length.
No Object			13					Slave supports the Cold Restart Function and will return Obj 52, Var 2, Qual 7, Cnt 1
			14					Slave supports the Warm Restart Function and will return Obj 52, Var 2, Qual 7, Cnt 1
			20					Slave supports the Enable Unsolicited Function
			21					Slave supports the Disable Unsolicited Function
			23					Slave supports the Delay Measurement & Time Synchronization Function and will return Obj 52, Var 2, Qual 7, Cnt 1

4 Support, Service & Warranty

In This Chapter

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

4.1 Warranty Information

For complete details regarding ProSoft Technology's TERMS & CONDITIONS OF SALE, WARRANTY, SUPPORT, SERVICE AND RETURN MATERIAL AUTHORIZATION INSTRUCTIONS please see the documents on the Product DVD or at www.prosoft-technology.com/warranty.

All documentation is subject to change without notice.

Index