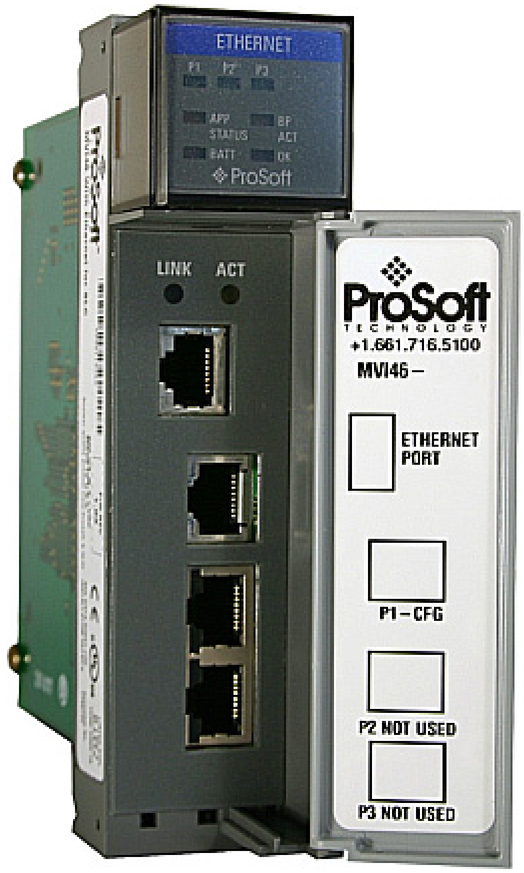
ProSoft TECHNOLOGY

Where Automation Connects.



inRAx®

MVI46-DNPSNET

SLC Platform

Distributed Network Protocol Interface Module

September 22, 2008

USER MANUAL

Please Read This Notice

Successful application of this module requires a reasonable working knowledge of the Rockwell Automation SLC hardware, the MVI46-DNPSNET Module and the application in which the combination is to be used. For this reason, it is important that those responsible for implementation satisfy themselves that the combination will meet the needs of the application without exposing personnel or equipment to unsafe or inappropriate working conditions.

This manual is provided to assist the user. Every attempt has been made to ensure that the information provided is accurate and a true reflection of the product's installation requirements. In order to ensure a complete understanding of the operation of the product, the user should read all applicable Rockwell Automation documentation on the operation of the Rockwell Automation hardware.

Under no conditions will ProSoft Technology be responsible or liable for indirect or consequential damages resulting from the use or application of the product.

Reproduction of the contents of this manual, in whole or in part, without written permission from ProSoft Technology is prohibited.

Information in this manual is subject to change without notice and does not represent a commitment on the part of ProSoft Technology Improvements and/or changes in this manual or the product may be made at any time. These changes will be made periodically to correct technical inaccuracies or typographical errors.

Battery Life Advisory

All modules in the MVI series use a rechargeable Lithium Vanadium Pentoxide battery to backup the 512K SRAM memory, real-time clock, and CMOS. The battery should last for the life of the module.
The module must be powered for approximately twenty hours before it becomes fully charged. After it is fully charged, the battery provides backup power for the CMOS setup and configuration data, the real-time clock, and the 512K SRAM memory for approximately 21 days.
Before you remove a module from its power source, ensure that the battery within the module is fully charged. A fully charged battery will hold the BIOS settings (after being removed from its power source) for a limited number of days. When the battery is fully discharged, the module will revert to the default BIOS settings.

Note: The battery is not user replaceable.

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 the product, documentation or support, please write or call us.

ProSoft Technology
1675 Chester Avenue, Fourth Floor
Bakersfield, CA 93301
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
http://www.prosoft-technology.com

MVI46-DNPSNET User Manual
September 22, 2008

ProSoft® 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 and are available at no charge from our web site: http://www.prosoft-technology.com
Printed documentation is available for purchase. Contact ProSoft Technology for pricing and availability.
Asia Pacific: +603.7724.2080
Europe, Middle East, Africa: +33.5.34.36.87.20
Latin America: +1.281.298.9109
North America: +1.661.716.5100

Contents

Guide to the MVI46-DNPSNET User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 9)	This Section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Verify Communication, Diagnostic and Troubleshooting	→	Verifying Communication (page 44) Diagnostics and Troubleshooting (page 33)	This section describes how to verify communications with the network. Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview Glossary	→	Reference (page 49) Functional Overview (page 51) Product Specifications (page 49)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 89)	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

Installing the MVI46-DNPSNET module requires a reasonable working knowledge of the Rockwell Automation hardware, the MVI46-DNPSNET Module and the application in which they will be used.

Caution: It is important that those responsible for implementation can complete the application without exposing personnel, or equipment, to unsafe or inappropriate working conditions. Safety, quality and experience are key factors in a successful installation.

1.1 System Requirements

The MVI46-DNPSNET module requires the following minimum hardware and software components:

- Rockwell Automation SLC 5/02 M0/M1 capable processors (or newer), with compatible power supply and one free slot in the rack, for the MVI46-DNPSNET module. The module requires 800mA of available power.
- Rockwell Automation RSLogix 500 programming software.
- Rockwell Automation RSLinx communication software
- Pentium® II 500 MHz minimum. Pentium III 733 MHz (or better) recommended
- Supported operating systems:
  - Microsoft® Windows 98
  - Windows NT® (version 4 with SP4 or higher)
  - Windows 2000
  - Windows XP
- 32 Mbytes of RAM minimum, 64 Mbytes of RAM recommended
- 50 Mbytes of free hard disk space (or more based on application requirements)

- 16-color VGA graphics adapter, 640 x 480 minimum resolution (256 Color 800 × 600 recommended)
- CD-ROM drive
- 3.5 inch floppy disk drive
- HyperTerminal or other terminal emulator program capable of file transfers using Zmodem protocol.

1.2 Package Contents

The following components are included with your MVI46-DNPSNET module, and are all required for installation and configuration.

Important: Before beginning the installation, please verify that all of the following items are present.

Qty.	Part Name	Part Number	Part Description
1	MVI46-DNPSNET Module	MVI46-DNPSNET	Distributed Network Protocol Interface Module
1	Cable	Cable #15, RS232 Null Modem	For RS232 Connection to the CFG Port
1	Cable	RJ45 to DB9 Male Adapter	For DB9 Connection to Module's Port
1	inRAx Solutions CD		Contains sample programs, utilities and documentation for the MVI46-DNPSNET module.

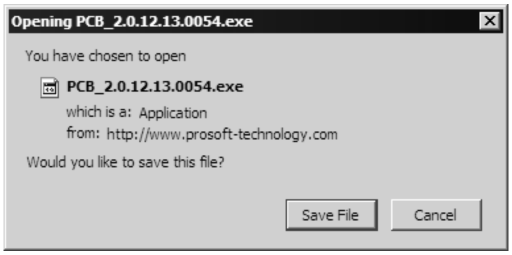
If any of these components are missing, please contact ProSoft Technology Support for replacement parts.

1.3 Install ProSoft Configuration Builder Software

You must install the ProSoft Configuration Builder (PCB) software in order to configure the MVI46-DNPSNET module. You can always get the newest version of ProSoft Configuration Builder from the ProSoft Technology web site.

To install ProSoft Configuration Builder from the ProSoft Web Site

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. The following illustrations show the file download prompt for two of the most common web browsers.



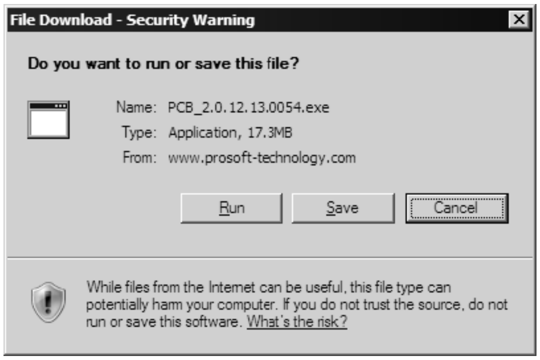
Opening PCB_2.0.12.13.0054.exe

You have chosen to open

PCB_2.0.12.13.0054.exe

which is a: Application
from: http://www.prosoft-technology.com

Would you like to save this file?



File Download - Security Warning

Do you want to run or save this file?

Name: PCB_2.0.12.13.0054.exe
Type: Application, 17.3MB
From: www.prosoft-technology.com

While files from the Internet can be useful, this file type can potentially harm your computer. If you do not trust the source, do not run or save this software. What's the risk?

4 Make a note of the location where you saved the file, for example "Desktop", or "My Documents", so you can start the installation program.
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 CD-ROM, included in the package with your MVI46-DNPSNET module.

To install ProSoft Configuration Builder from the CD-ROM

1 Insert the ProSoft Solutions CD-ROM into the CD drive of your PC. Wait for the startup screen to appear.
2 On the startup screen, click *Product Documentation*. This action opens an explorer window.

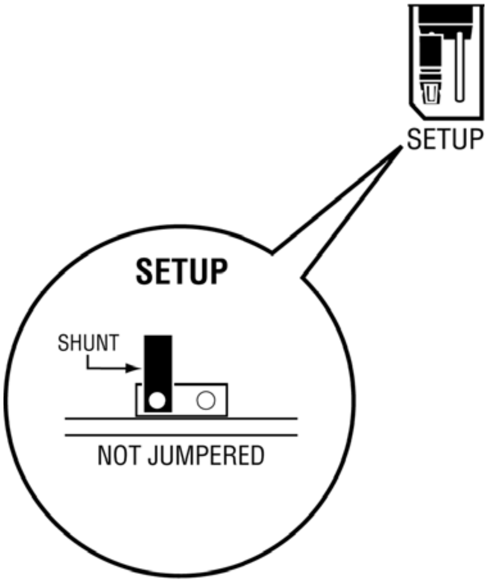
3 Click to open the *Utilities* folder. This folder contains all of the applications and files you will need to set up and configure your module.

4 Double-click the *ProSoft Configuration Builder Setup* program and follow the instructions on your screen to install the software on your PC.

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.

1.4 Setting Jumpers

Note: The Setup Jumper acts as "write protection" for the module's flash memory. In "write protected" mode, the Setup pins are not connected, and the module's firmware cannot be overwritten. Do not jumper the Setup pins together unless you are directed to do so by ProSoft Technical Support.



1.5 Install the Module in the Rack

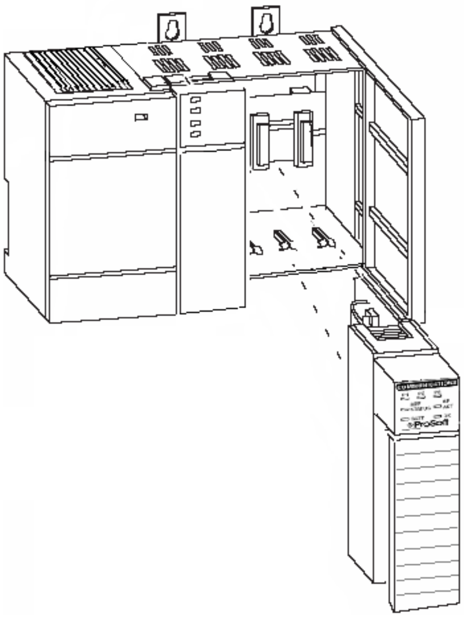
If you have not already installed and configured your SLC processor and power supply, please do so before installing the MVI46-DNPSNET module. Refer to your Rockwell Automation product documentation for installation instructions.

Warning: You must follow all safety instructions when installing this or any other electronic devices. Failure to follow safety procedures could result in damage to hardware or data, or even serious injury or death to personnel. Refer to the documentation for each device you plan to connect to verify that suitable safety procedures are in place before installing or servicing the device.

After you have checked the placement of the jumpers, insert MVI46-DNPSNET into the SLC™ chassis. Use the same technique recommended by Rockwell Automation to remove and install SLC™ modules.

Warning: This module is not hot-swappable! Always remove power from the rack before inserting or removing this module, or damage may result to the module, the processor, or other connected devices.

1 Turn power OFF.
2 Align the module with the top and bottom guides, and slide it into the rack until the module is firmly against the backplane connector.

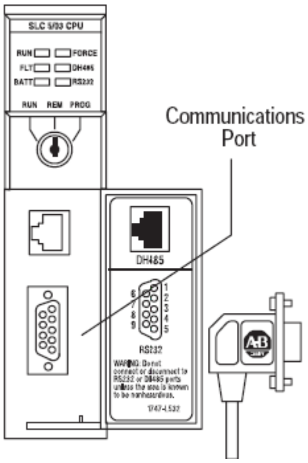


3 With a firm but steady push, snap the module into place.
4 Check that the holding clips on the top and bottom of the module are securely in the locking holes of the rack.
5 Make a note of the slot location. You will need to identify the slot in which the module is installed in order for the sample program to work correctly. Slot numbers are identified on the green circuit board (backplane) of the SLC rack.
6 Turn power ON.

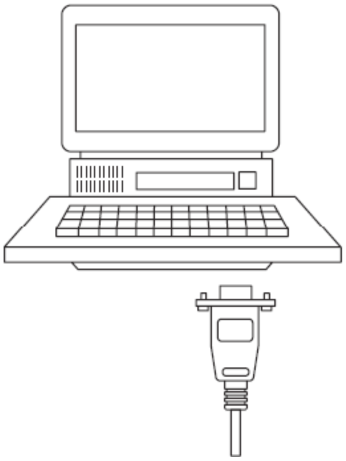
> **Note:** If you insert the module improperly, the system may stop working, or may behave unpredictably.

1.6 Connect your PC to the Processor

1 Connect the right-angle connector end of the cable to your controller at the communications port.


SLC 5/03 CPU
RUN
FORCE
FLT
DH485
BATT
RS232
RUN
REM
PROG
Communications Port
DH485
6
7
8
9
1
2
3
4
5
RS232
WARNING: Do not connect or disconnect to RS232 or DH485 ports unless the area is known to be nonhazardous.
1747-L532
A-B

2 Connect the straight connector end of the cable to the serial port on your computer.

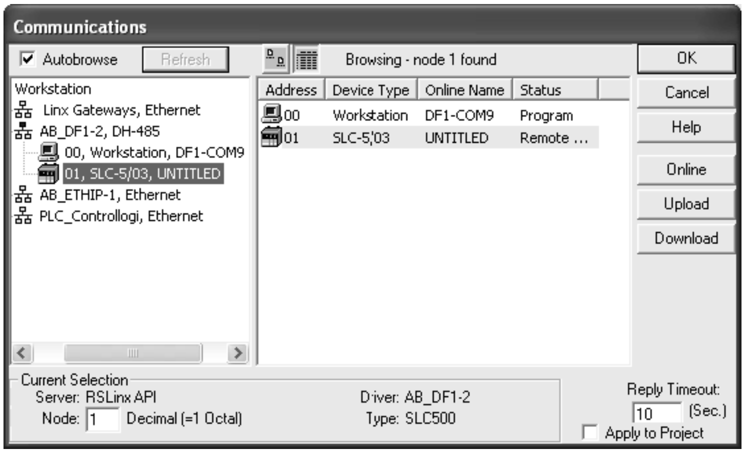


1.7 Download the Sample Program to the Processor

To download the sample program from RSLogix 500 to the SLC processor:

Note: The key switch on the front of the SLC processor must be in the REM position.

1 If you are not already online to the processor, open the Communications menu, and then choose Download. RSLogix will establish communication with the processor.



Browsing - node 1 found

Workstation
- Linx Gateways, Ethernet
- AB_DF1-2, DH-485
  - 00, Workstation, DF1-COM9
  - 01, SLC-5/03, UNTITLED
- AB_ETHIP-1, Ethernet
- PLC_Controllogi, Ethernet

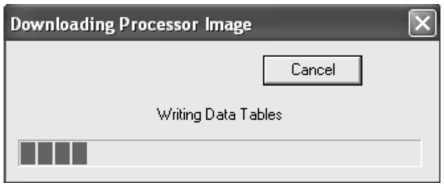
Address	Device Type	Online Name	Status
00	Workstation	DF1-COM9	Program
01	SLC-5/03	UNTITLED	Remote ...

Current Selection
Server: RSLinx API
Node: 1 Decimal (=1 Octal)
Driver: AB_DF1-2
Type: SLC500

Reply Timeout: 10 (Sec.)
☐ Apply to Project

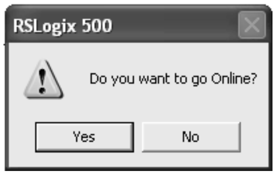
2 Click the Download button to transfer the sample program to the processor.

3 RSLogix will compile the program and transfer it to the processor. This process may take a few minutes.



Writing Data Tables

4 When the download is complete, RSLogix will open another confirmation dialog box. Click Yes to switch the processor from Program mode to Run mode.



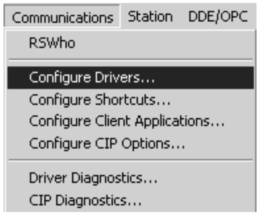
Do you want to go Online?

Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

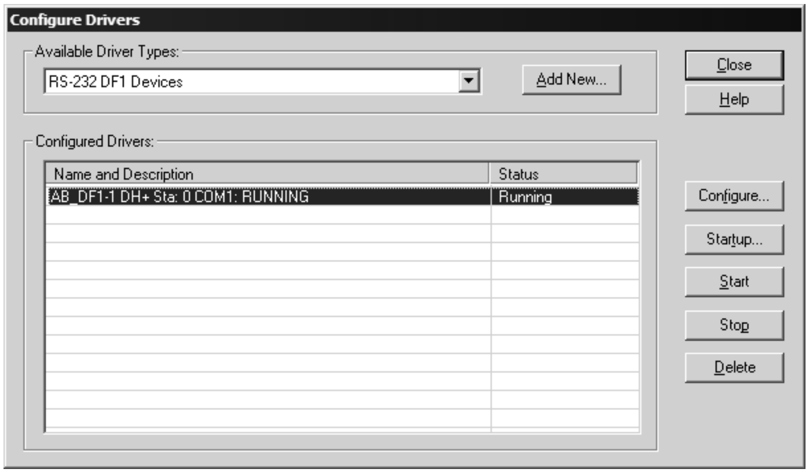
1.7.1 Configuring RSLinx

If RSLogix is unable to establish communication with the processor, follow these steps:

1 Open RSLinx.
2 Open the Communications menu, and choose Configure Drivers.



This action opens the Configure Drivers dialog box.

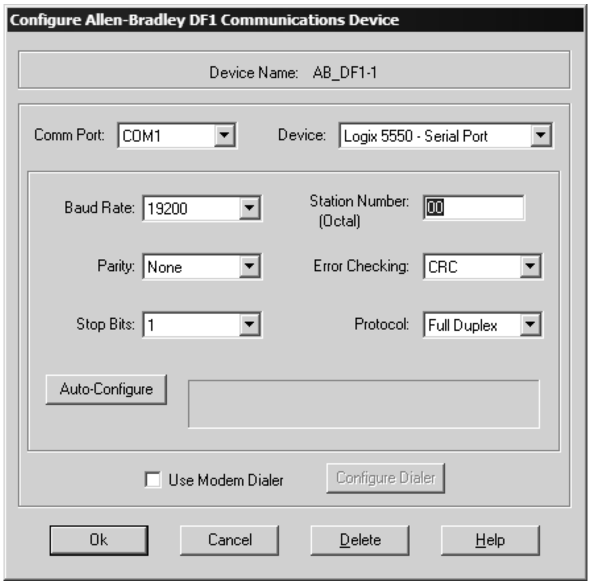


Available Driver Types: RS-232 DF1 Devices

Name and Description	Status
AB_DF1-1 DH+ Sta: 0 COM1: RUNNING	Running

Note: If the list of configured drivers is blank, you must first choose and configure a driver from the Available Driver Types list. The recommended driver type to choose for serial communication with the processor is "RS-232 DF1 Devices".

3 Click to select the driver, and then click Configure. This action opens the Configure Allen-Bradley DF1 Communications Device dialog box.



Device Name: AB_DF1-1

Comm Port: COM1

Device: Logix 5550 - Serial Port

Baud Rate: 19200

Station Number: (Octal) 00

Parity: None

Error Checking: CRC

Stop Bits: 1

Protocol: Full Duplex

4 Click the Auto-Configure button. RSLinx will attempt to configure your serial port to work with the selected driver.

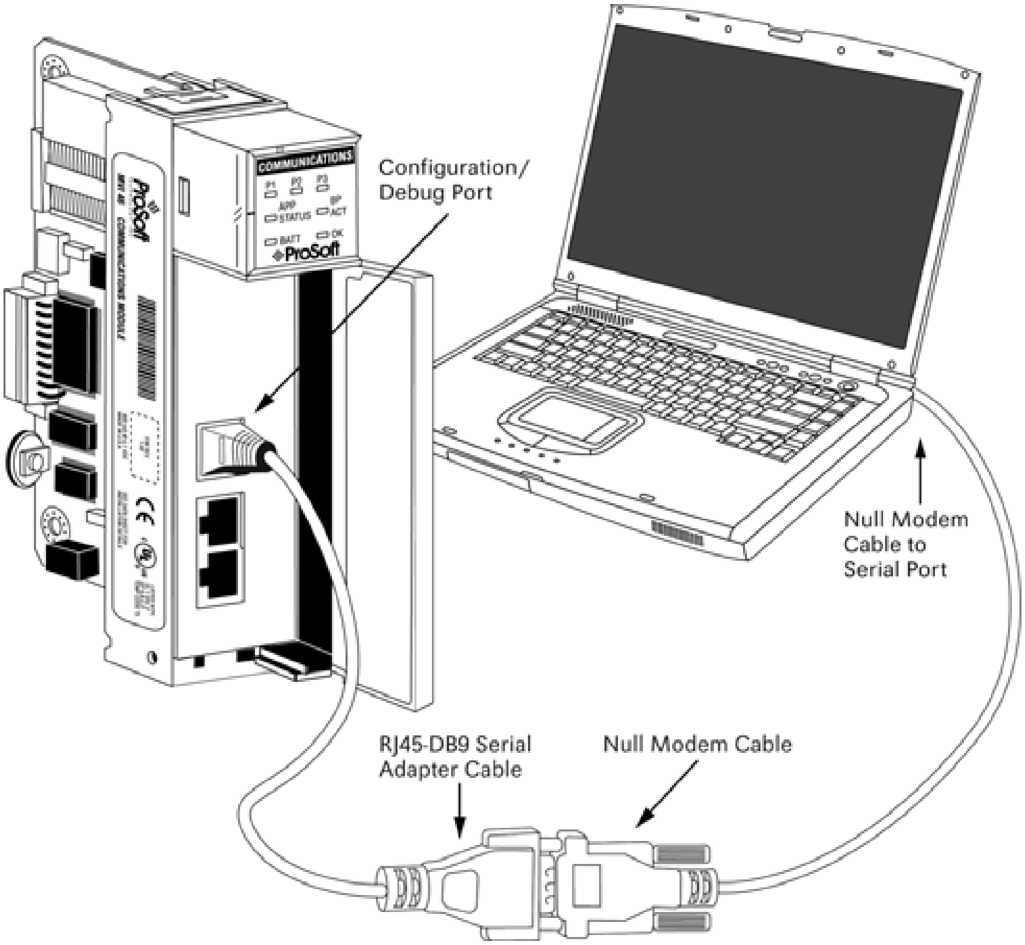
5 When you see the message "Auto Configuration Successful", click the OK button to dismiss the dialog box.

Note: If the auto-configuration procedure fails, verify that the cables are connected correctly between the processor and the serial port on your computer, and then try again. If you are still unable to auto-configure the port, refer to your RSLinx documentation for further troubleshooting steps.

1.8 Connect your PC to the Module

With the module securely mounted, connect your PC to the Configuration/Debug port using an RJ45-DB-9 Serial Adapter Cable and a Null Modem Cable.

1. Attach both cables as shown.
2. Insert the RJ45 cable connector into the Configuration/Debug port of the module.
3. Attach the other end to the serial port on your PC or laptop.



2 Installing and Configuring the Module

In This Chapter

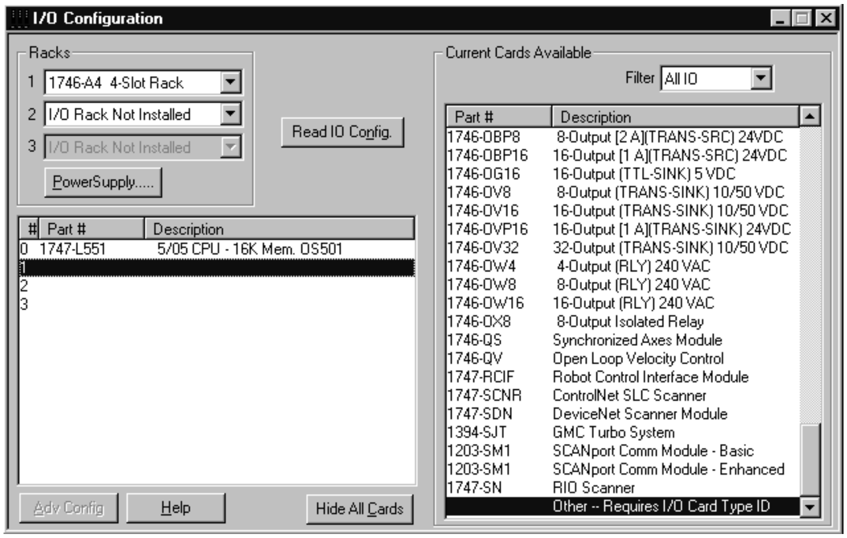
The configuration process consists of the following steps.

1 Download the sample program to the processor.

Note: For most applications, the sample program will work without modification. We strongly recommend setting up the module first with the sample program, before attempting to add the module to an existing application or create a custom application.

2 Modify the module's configuration files to meet the needs of your application, and copy the updated configuration to the module. Example configuration files are provided on the CD-ROM. Refer to Modifying the Configuration File (page 24) for more information on the configuration files.

3 First, define the module to the system. Select the I/O Configuration option from the program screen. This will display the following dialog box:



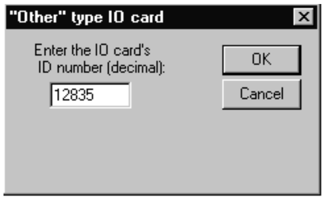
1 1746-A4 4-Slot Rack
2 I/O Rack Not Installed
3 I/O Rack Not Installed

Filter All IO

#	Part #	Description
0	1747-L551	5/05 CPU - 16K Mem. OS501
1		
2		
3		

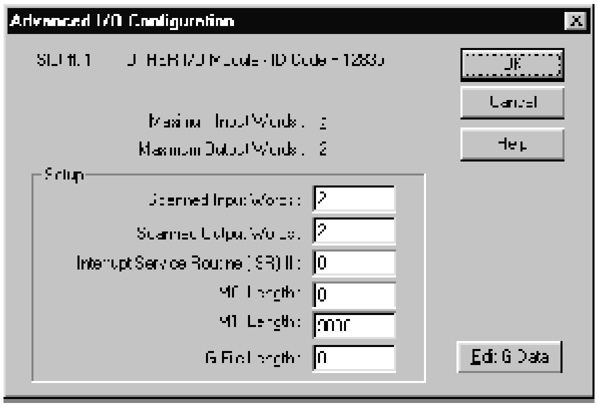
Part #	Description
1746-OBP8	8-Output [2 A](TRANS-SRC) 24VDC
1746-OBP16	16-Output [1 A](TRANS-SRC) 24VDC
1746-OG16	16-Output (TTL-SINK) 5 VDC
1746-OV8	8-Output (TRANS-SINK) 10/50 VDC
1746-OV16	16-Output (TRANS-SINK) 10/50 VDC
1746-OVP16	16-Output [1 A](TRANS-SINK) 24VDC
1746-OV32	32-Output (TRANS-SINK) 10/50 VDC
1746-OW4	4-Output (RLY) 240 VAC
1746-OW8	8-Output (RLY) 240 VAC
1746-OW16	16-Output (RLY) 240 VAC
1746-OX8	8-Output Isolated Relay
1746-QS	Synchronized Axes Module
1746-QV	Open Loop Velocity Control
1747-RCIF	Robot Control Interface Module
1747-SCNR	ControlNet SLC Scanner
1747-SDN	DeviceNet Scanner Module
1394-SJT	GMC Turbo System
1203-SM1	SCANport Comm Module - Basic
1203-SM1	SCANport Comm Module - Enhanced
1747-SN	RIO Scanner
	Other -- Requires I/O Card Type ID

4 Select the Other module from the list. This cause the program to display the following dialog box:


"Other" type IO card

Enter the IO card's ID number (decimal): 12835

5 Enter the module I/O card ID number as 12835, and then click OK. Double-click the mouse on the module just added to the rack. Fill in the dialog box presented as shown below.


Advanced I/O Configuration

SLOT 1 OTHER I/O Module - ID Code = 12835

Maximum Input Words : 2
Maximum Output Words : 2

Setup
Scanned Input Words : 2
Scanned Output Words : 2
Interrupt Service Routine (ISR) # : 0
M0 Length: 0
M1 Length: 3000
G File Length: 0

6 Click OK to apply these settings to the module. Then, close the I/O Configuration dialog box.

7 Next, define the user defined data areas to hold the status and read and write database areas. Edit the DNPSNET.CFG file now for the application to implement. Use any text editor to set the values in the file. You must retain the file name, DNPSNET.CFG.

8 The last step in the module setup is to add the ladder logic. If the example ladder logic is used, adjust the ladder to fit the application. When the ladder example is not used, copy the example ladder logic to your application and alter as necessary.

The module is now set up and ready to be used with your application. Insert the module in the rack, connect the module to the Ethernet network and attach the serial communication cable to the Debug/Configuration port. Download the DNPSNET.CFG file to the module. Download the new application to the controller and place the processor in run mode. If all the configuration parameters are set correctly and the module is attached to a network, the module's Application LED (APP LED) should remain off and the backplane activity LED (BP ACT) should blink very rapidly. Refer to the **Diagnostics and Trouble Shooting** section of this manual if you encounter errors. Attach a computer or terminal to Debug/Configuration port on the module and look at the status of the module using the Configuration/Debug Menu in the module. Refer to the **Diagnostics and Troubleshooting** section for a complete discussion of the use of this feature:

2.1 Module Configuration

In order for the MVI46-DNPSNET module to function, a minimum amount of configuration data must be transferred to the module. A text, configuration file named **DNPSNET.CFG** is shipped with the module. This file can serve as a starting point to develop a user application. Edit the file to configure the module for the application. Care must be taken in designing the system and building the file in order for the module to function as desired.

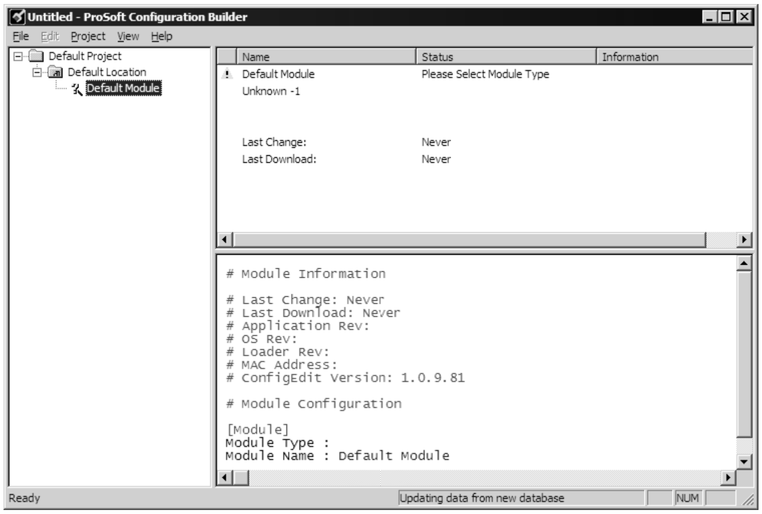
Refer to the **Module Set Up** section for a description of the configuration of the module. A terminal server program is used to upload and download the configuration file to the module. Refer to the **Diagnostics and Troubleshooting** section to perform these tasks. Additionally, the **WATTCP.CFG** file in the module must be configured for the specific network on with the module resides. Before connecting the module to the network, verify that the information in the **WATTCP.CFG** file is compatible with your network. Failure to properly construct the configuration or network file may cause problems when the module is connected to the network.

2.2 ProSoft Configuration Builder

ProSoft Configuration Builder (PCB) provides a quick and easy 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.

2.2.1 Set Up the Project

To begin, start ProSoft Configuration Builder. If you have used other Windows configuration tools before, you will find the screen layout familiar. ProSoft Configuration Builder'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 ProSoft Configuration Builder, 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 ProSoft Configuration Builder window with a new project.



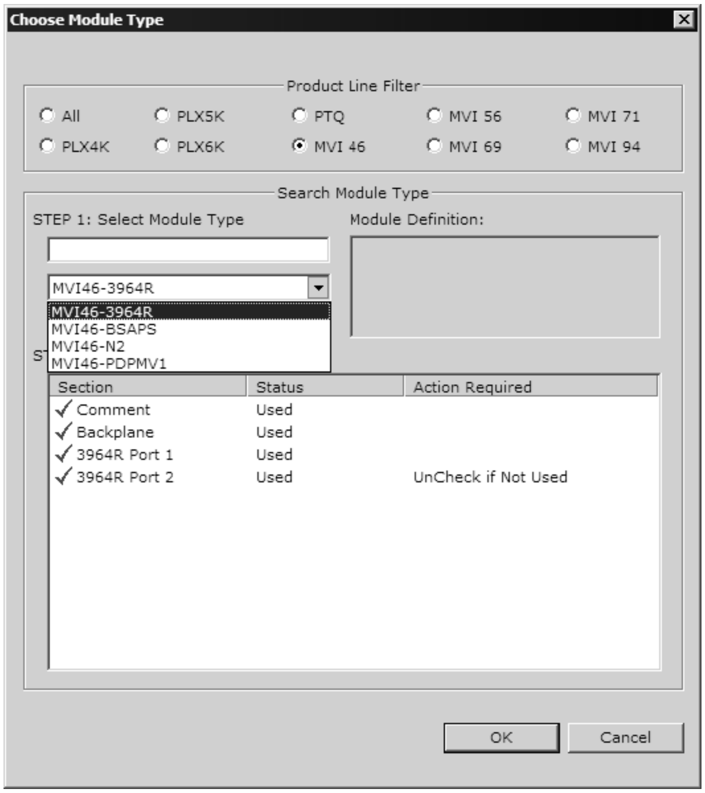
Name	Status	Information
Default Module	Please Select Module Type	
Unknown -1		
Last Change:	Never	
Last Download:	Never	

```
# Module Information

# Last Change: Never
# Last Download: Never
# Application Rev:
# OS Rev:
# Loader Rev:
# MAC Address:
# ConfigEdit Version: 1.0.9.81

# Module Configuration

[Module]
Module Type :
Module Name : Default Module
```

Your first task is to add the MVI46-DNPSNET module 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.



Product Line Filter: ○ All ○ PLX5K ○ PTQ ○ MVI 56 ○ MVI 71 ○ PLX4K ○ PLX6K ● MVI 46 ○ MVI 69 ○ MVI 94

STEP 1: Select Module Type: MVI46-3964R

MVI46-3964R
MVI46-BSAPS
MVI46-N2
MVI46-PDPMV1

Section	Status	Action Required
✓ Comment	Used	
✓ Backplane	Used	
✓ 3964R Port 1	Used	
✓ 3964R Port 2	Used	UnCheck if Not Used

3 In the Product Line Filter area of the dialog box, select MVI46. In the Select Module Type dropdown list, select MVI46-DNPSNET, and then click OK to save your settings and return to the ProSoft Configuration Builder window.

The next task is to set the module parameters.

Adding a Module

To add a module to your project:

1 Double-click the Default Module icon to open the Choose Module Type dialog box.
2 On the Choose Module Type dialog box, select the module type.

Or

1 Open the Project menu and choose Location.
2 On the Location menu, choose Add Module.

To add a module to a different location:

1 Right-click the Location folder and choose Add Module. A new module icon appears.

Or

1 Select the Location icon.
2 From the Project menu, select Location, then select Add Module.

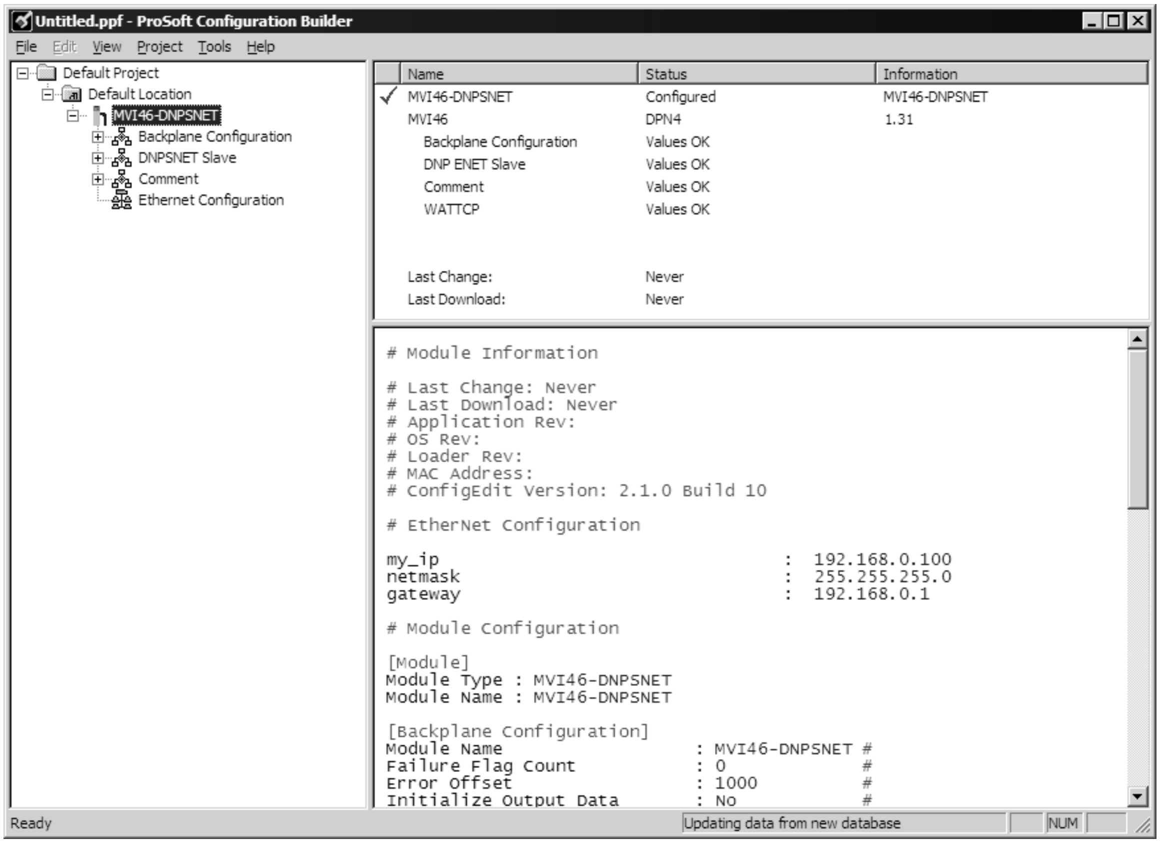
Adding a Project

To add a project to an existing project file:

1 Select the Default Project icon.
2 Choose Project from the Project menu, then choose Add Project. A new project folder appears.

2.2.2 Set Module Parameters

Notice that the contents of the information pane and the configuration pane changed when you added the MVI46-DNPSNET module to the project.



	Name	Status	Information
✓	MVI46-DNPSNET	Configured	MVI46-DNPSNET
	MVI46	DPN4	1.31
	Backplane Configuration	Values OK	
	DNP ENET Slave	Values OK	
	Comment	Values OK	
	WATTCP	Values OK	
	Last Change:	Never	
	Last Download:	Never	

```
# Module Information

# Last Change: Never
# Last Download: Never
# Application Rev:
# OS Rev:
# Loader Rev:
# MAC Address:
# ConfigEdit Version: 2.1.0 Build 10

# EtherNet Configuration

my_ip                                :  192.168.0.100
netmask                              :  255.255.255.0
gateway                              :  192.168.0.1

# Module Configuration

[Module]
Module Type : MVI46-DNPSNET
Module Name : MVI46-DNPSNET

[Backplane Configuration]
Module Name                  : MVI46-DNPSNET #
Failure Flag Count           : 0             #
Error Offset                 : 1000          #
Initialize Output Data       : No            #
```

Updating data from new database

At this time, you may wish to rename the "Default Project" and "Default Location" folders in the tree view.

To rename an object:

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.

Module Entries

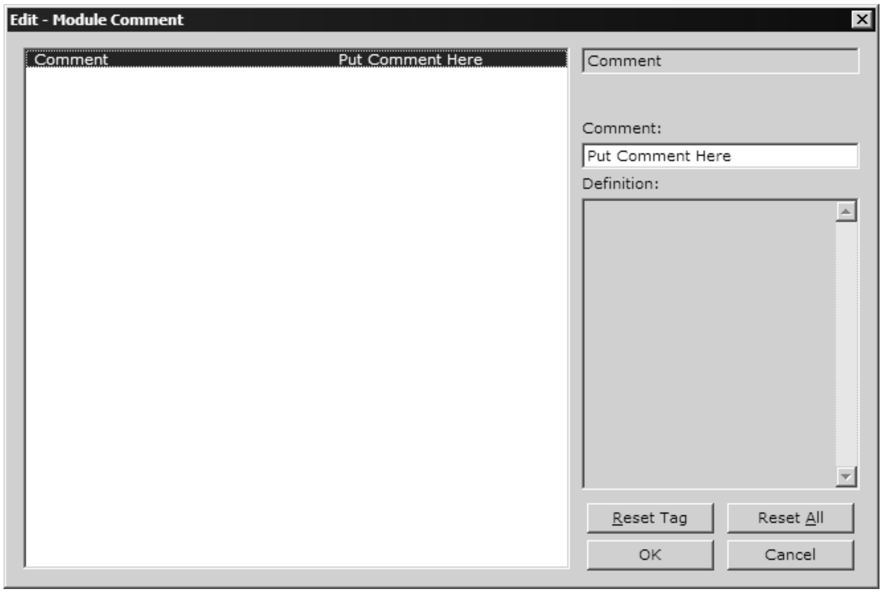
To configure module parameters

1 Click on the plus sign next to the icon ⊞ Comment to expand module information.
2 Double-click the Module Comment icon to open the Edit dialog box.
3 To edit a parameter, select the parameter in the left pane and make your changes in the right pane.
4 Click OK to save your changes.

Comment Entries

To add comments to your configuration file:

1 Click the plus sign to the left of the ⊞ Comment icon to expand the Module Comments.
2 Double-click the Module Comment icon. The Edit - Module Comment dialog appears.



Edit - Module Comment

Comment	Put Comment Here

Comment

Comment: Put Comment Here

Definition:

3 Enter your comment and click OK to save your changes.

Printing a Configuration File

To print 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 On the View Configuration window, open the File menu, and choose Print. This action opens the Print dialog box.
4 On the Print dialog box, choose the printer to use from the dropdown list, select printing options, and then click OK.

2.3 MVI46-DNPSNET Server Communication Module Configuration

[Section]/Item	Value	Range	Description
[Backplane Configuration]			Backplane transfer parameters
Module Name:		0 to 80 characters	This parameter assigns a name to the module that can be viewed using the configuration/debug port. Use this parameter to identify the module and the configuration file.
Password:		Up to 20 chars	This parameter sets the password for the module. If the parameter is not set, the module will not be password protected. The module interprets the password as a case-sensitive string.
Failure Flag Count:		0 to 65535	This parameter specifies the number of successive transfer errors that must occur before the communication ports are shut down. If the parameter is set to 0, the communication ports will continue to operate under all conditions. If the value is set larger than 0, the module will stop communications when the preset value is reached based on successive failures.
Error Offset:		0 to	This parameter specifies the register location in the module's database where module status data will be stored. If a value less than 0 is entered, the data will not be stored in the database. If the value specified is in the range of 0 to 8966, the data will be placed in the modules database.
Initialize Output Data:		Y or N	This parameter determines if the output data for the module should be initialized with values from the processor. If the value is set to N, the output data will be initialized to 0. If the value is set to Y, the data will be initialized with data from the processor.
M1 Write Size		0 to 9000	Size of the M1 file that is to be transferred.

[Section]/Item	Value	Range	Description
[DNP ENET Slave]			Server and protocol parameters
Internal Slave ID:		0 to 65534	This is the DNP address for the module. All messages with this address from the master will be processed by the module.
Use IP List:		Y or N	This parameter specifies if the IP address of the host connected to the system will be validated. If the parameter is set to N, any host may connect to the unit. If the parameter is set to Y, 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.

[Section]/Item	Value	Range	Description
Binary Inputs:		0 to 500 words	Number of digital input 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 inputs will be defined for the application.
Analog Inputs:		0 to 500 points	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	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.
Binary Outputs:		0 to 500 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	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.
Write Time Interval:		0 to 1440 minutes	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.
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:		Y or N	Set if the slave unit will send unsolicited response messages. If set to N, the slave will not send unsolicited responses. If set to Y, the slave will send unsolicited responses.

[Section]/Item	Value	Range	Description
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 65534	DNP destination address where unsolicited response messages are sent.
AI Events with time:		Y or N	This parameter sets if the analog input events generated by the module will include the date and time of the event. If the parameter is set to N, the default is set to no time data. If the parameter is set to Y, the default object will include the time of the event.
Time Sync Before Events:		Y or N	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 N, events will be generated irrespective of the module's time sync status. If the parameter is set to Y, events will be generated only if the module's time is synchronized.

[Section]/Item	Value	Range	Description
[DNP ENET IP Addresses]	[redacted]	[redacted]	List of valid IP addresses accepted by module
START			
# Insert the list of IP addresses for the host(s) to connect to this unit. Only used if Use IP List set to 1.			
END			

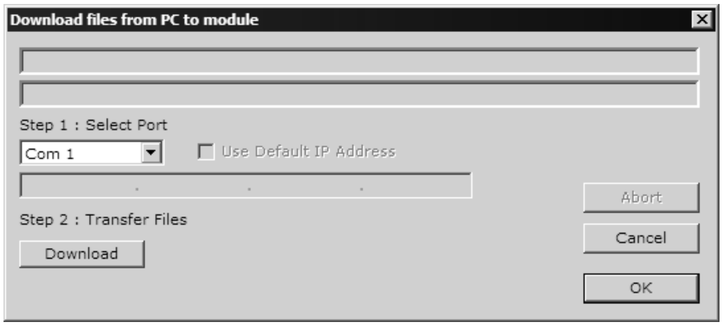
2.4 Download the Project to the Module

In order 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 MVI46-DNPSNET module.

2 Open the **Project menu,** and then choose **Module / Download**. 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 following dialog box will open.



Download files from PC to module

Step 1 : Select Port

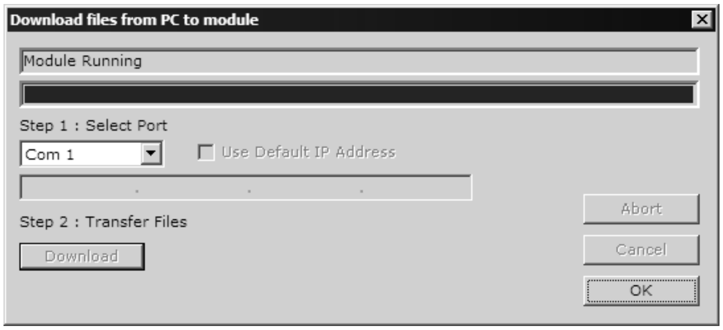
Com 1

☐ Use Default IP Address

Step 2 : Transfer Files

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 ProSoft Configuration Builder will be updated with the message *"Module Running*".



Download files from PC to module

Module Running

Step 1 : Select Port

Com 1

☐ Use Default IP Address

Step 2 : Transfer Files

3 Ladder Logic

In This Chapter

Ladder logic is required for application of the MVI46-DNPSNET module. Tasks that must be handled by the ladder logic are module data transfer, special block handling and status data receipt. Additionally, a power-up handler may be needed to handle the initialization of the module's data and to clear any processor fault conditions.

The sample ladder logic, on the ProSoft Solutions CD-ROM, is extensively commented, to provide information on the purpose and function of each rung. For most applications, the sample ladder will work without modification.

3.1 Module Data

All data related to the MVI46-DNPSNET module is stored in a user defined data files and the module's M1 file. Files should be defined for each data type to be used with the module. Additionally, a file should be defined to hold the module status data. The status data should be copied from the M1 file and placed in the assigned status file. Input (monitor) data should be copied from the user file to the M1 file and output (command) data should be copied from the user files to the M1 file. Refer to the Reference chapter of this document for a complete listing of the status data area for the module.

4 Diagnostics and Troubleshooting

In This Chapter

The module provides information on diagnostics and troubleshooting in the following forms:

- Status data values are transferred from the module to the processor.
- Data contained in the module can be viewed through the Configuration/Debug port attached to a terminal emulator.
- LED status indicators on the front of the module provide information on the module's status.

4.1 Reading Status Data from the Module

The MVI46-DNPSNET module provides the status data in each read block. This data can also be located in the module's database. For a complete listing of the status data object, refer to the **Module Set Up** section.

4.1.1 Required Hardware

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

ProSoft Technology recommends the following minimum hardware to connect your computer 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.
- A null modem serial cable.

4.1.2 The Configuration/Debug Menu

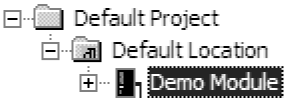
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

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.

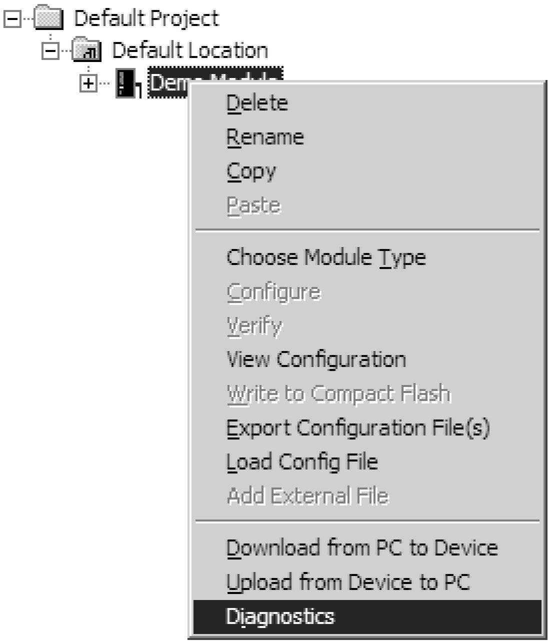
Using the Diagnostic Window in ProSoft Configuration Builder

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

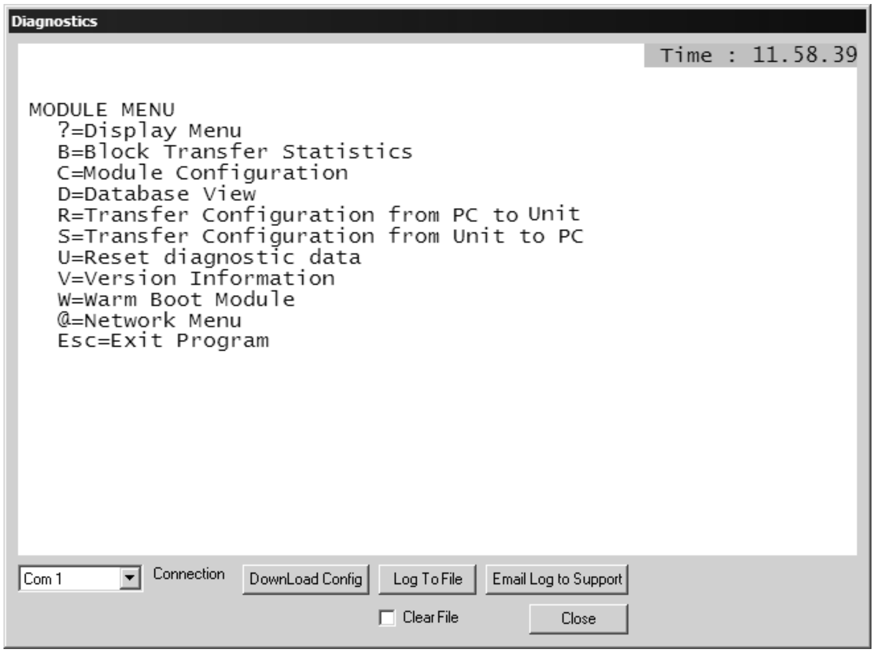
1 Start PCB program with the application file to be tested. Right click over the module icon.



2 On the shortcut menu, choose Diagnostics.



3 This action opens the Diagnostics dialog box. Press "?" to display the Main Menu.



```
                                                   Time : 11.58.39

MODULE MENU
  ?=Display Menu
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  R=Transfer Configuration from PC to Unit
  S=Transfer Configuration from Unit to PC
  U=Reset diagnostic data
  V=Version Information
  W=Warm Boot Module
  @=Network Menu
  Esc=Exit Program
```

Important: The illustrations of configuration/debug menus in this section are intended as a general guide, and may not exactly match the configuration/debug menus in your own module.

If there is no response from the module, follow these steps:

1 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.

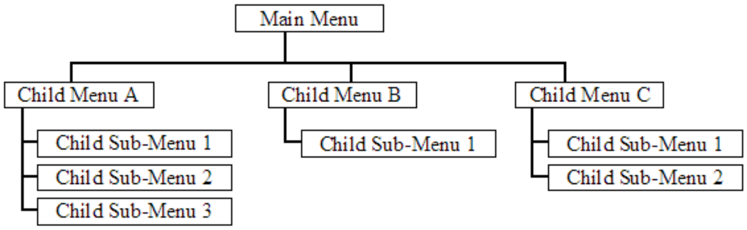
2 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

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

Navigation

All of the sub-menus for this module contain commands to redisplay the menu or return to the previous menu. You can always return from a sub-menu 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 you the menus available for this module, and briefly discusses the commands available to you.

Keystrokes

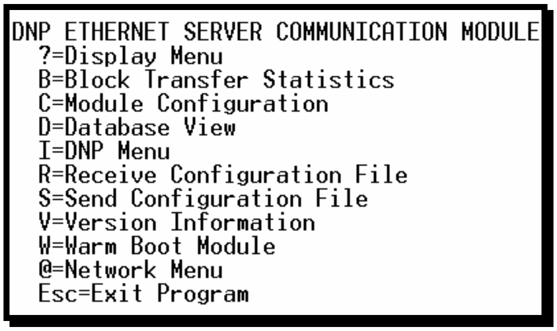
The keyboard commands on these menus are almost always non-case sensitive. You can enter most commands in lower case or capital letters.

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][/]**.

Also, take care to distinguish capital letter **[I]** from lower case letter **[l]** (L) and number **[1]**; likewise for capital letter **[O]** and number **[0]**. Although these characters look nearly the same on the screen, they perform different actions on the module.

4.1.3 Main Menu

When you first connect to the module from your computer, your terminal screen will be blank. To activate the main menu, press the [?] key on your computer's keyboard. If the module is connected properly, the following menu will appear on your terminal screen:



```
DNP ETHERNET SERVER COMMUNICATION MODULE
  ?=Display Menu
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  I=DNP Menu
  R=Receive Configuration File
  S=Send Configuration File
  V=Version Information
  W=Warm Boot Module
  @=Network Menu
  Esc=Exit Program
```

Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Only use these commands if you are specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

Viewing Block Transfer Statistics

Press **[B]** from the Main Menu to view the Block Transfer Statistics screen.

Use this command to display the configuration and statistics of the backplane data transfer operations between the module and the processor. The information on this screen can help determine if there are communication problems between the processor and the module.

Tip: To determine the number of blocks transferred each second, mark the numbers displayed at a specific time. Then some seconds later activate the command again. Subtract the previous numbers from the current numbers and divide by the quantity of seconds passed between the two readings.

Viewing Module Configuration

Press **[C]** to view the Module Configuration screen.

Use this command to display the current configuration and statistics for the module.

Opening the Database Menu

Press **[D]** to open the Database View menu. Use this menu command to view the current contents of the module's database.

Opening the DNP Menu

Press **[I]** from the Main Menu to open the DNP Menu. This menu allows you to view all data associated with the DNP Server driver. For more information about the commands on this menu, refer to DNP Menu (page 38).

Receiving the Configuration File

Press **[R]** to download (receive) the current configuration file from the module. For more information on receiving and sending configuration files, please see Uploading and Downloading the Configuration File.

Sending the Configuration File

Press **[S]** to upload (send) an updated configuration file to the module. For more information on receiving and sending configuration files, please see Uploading and Downloading the Configuration File.

Viewing Version Information

Press **[V]** to view Version information for the module.

Use this command to view the current version of the software for the module, as well as other important values. You may be asked to provide this information when calling for technical support on the product.

Values at the bottom of the display are important in determining module operation. The Program Scan Counter value is incremented each time a module's program cycle is complete.

Tip: Repeat this command at one-second intervals to determine the frequency of program execution.

Warm Booting the Module

Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Only use these commands if you are specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

Press **[W]** from the Main Menu to warm boot (restart) the module. This command will cause the program to exit and reload, refreshing configuration parameters that must be set on program initialization. Only use this command if you must force the module to re-boot.

Opening the Network Menu

Press **[@]** to open the network menu. The network menu allows you to send, receive and view the WATTCP.CFG file that contains the IP, gateway and other network specification information. You can find more information about the commands on this menu in the Network Menu (page 43) section.

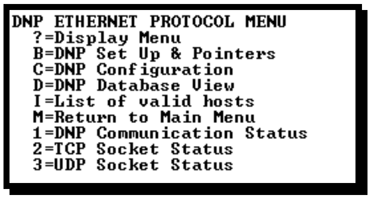
Exiting the Program

Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Only use these commands if you are specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

Press **[Esc]** to restart the module and force all drivers to be loaded. The module will use the configuration stored in the module's Flash ROM to configure the module.

4.1.4 DNP Menu

This opens the DNP menu. After the option is selected, press the '?' key to display the menu and the following is displayed:



```
DNP ETHERNET PROTOCOL MENU
  ?=Display Menu
  B=DNP Set Up & Pointers
  C=DNP Configuration
  D=DNP Database View
  I=List of valid hosts
  M=Return to Main Menu
  1=DNP Communication Status
  2=TCP Socket Status
  3=UDP Socket Status
```

Each option on the menu is discussed in the following topics.

Viewing DNP Set Up & Pointers

Press **[B]** to display the memory allocation and the database setup parameters.

Viewing DNP Configuration

Press **[C]** to displays the configuration information for the server. Use this command to confirm that the module is configured as desired. If any parameter is not set correctly, adjust the configuration file and download the altered file to the unit.

Opening the DNP Database View Menu

Press **[D]** to open the DNP Database View menu. Use this command to display the database associated with each data type.

Viewing a List of Valid Hosts

Press **[I]** to view the list of IP addresses from which the module will accept connections This list is only used if the module configuration parameter, Use IP List, is set to a value other than 0.

Returning to the Main Menu

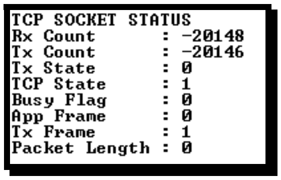
Press **[M]** to return to the Main Menu.

Viewing DNP Communication Status

Press **[1]** to view DNP Communication Status. Use this command to view the communication status data for the DNP driver.

Viewing TCP Socket Status

Press **[2]** to view the status of the TCP socket in the module. After selecting the option, the following is displayed:



```
TCP SOCKET STATUS
Rx Count      : -20148
Tx Count      : -20146
Tx State      : 0
TCP State     : 1
Busy Flag     : 0
App Frame     : 0
Tx Frame      : 1
Packet Length : 0
```

The parameters displayed have the following definitions:

Rx Count - Number of messages received on TCP socket

Tx Count - Number of messages transmitted on TCP socket

Tx State - 0=not transmitting, 1=transmitting

TCP State - Value used for TCP/IP socket state machine

Busy Flag - 0=not busy, 1=TCP has control of DNP server, 2=UDP has control of DNP server, 3=Unsolicited message being sent

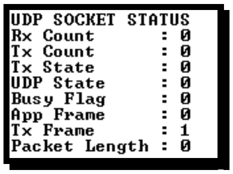
App Frame - 0=no application data frame data, 1=application data available

Tx Frame - 0=Data link level frame ready to send, 1=Data link level message not ready to send

Packet Length - Length of message left to process

Viewing UDP Socket Status

Press **[3]** to view the status of the UDP socket in the module. After selecting the option, the following is displayed:



```
UDP SOCKET STATUS
Rx Count      : 0
Tx Count      : 0
Tx State      : 0
UDP State     : 0
Busy Flag     : 0
App Frame     : 0
Tx Frame      : 1
Packet Length : 0
```

The parameters displayed have the following definitions:

Rx Count - Number of messages received on UDP socket

Tx Count - Number of messages transmitted on UDP socket

Tx State - 0=not transmitting, 1=transmitting

TCP State - Value used for UDP/IP socket state machine

Busy Flag - 0=not busy, 1=TCP has control of DNP server, 2=UDP has control of DNP server, 3=Unsolicited message being sent

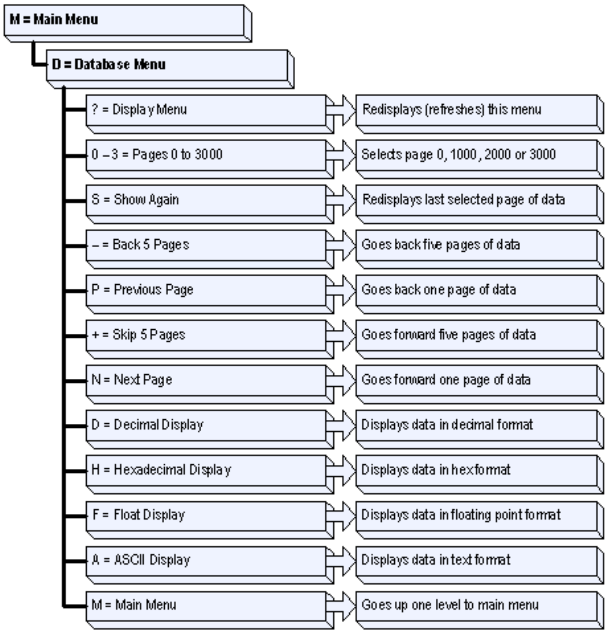
App Frame - 0=no application data frame data, 1=application data available

Tx Frame - 0=Data link level frame ready to send, 1=Data link level message not ready to send

Packet Length - Length of message left to process

4.1.5 Database View Menu

Press **[D]** from the Main Menu to open the Database View menu. Use this menu command to view the current contents of the module's database. Press **[?]** to view a list of commands available on this menu.


M = Main Menu
D = Database Menu
? = Display Menu
Redisplays (refreshes) this menu
0 – 3 = Pages 0 to 3000
Selects page 0, 1000, 2000 or 3000
S = Show Again
Redisplays last selected page of data
– = Back 5 Pages
Goes back five pages of data
P = Previous Page
Goes back one page of data
+ = Skip 5 Pages
Goes forward five pages of data
N = Next Page
Goes forward one page of data
D = Decimal Display
Displays data in decimal format
H = Hexadecimal Display
Displays data in hex format
F = Float Display
Displays data in floating point format
A = ASCII Display
Displays data in text format
M = Main Menu
Goes up one level to main menu

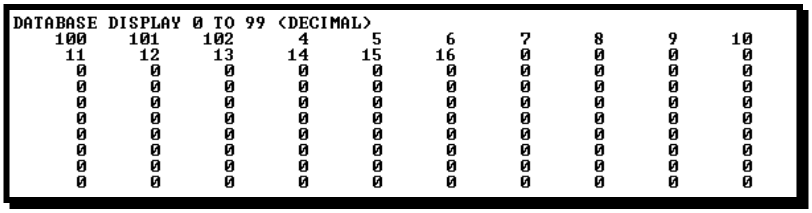
Viewing Register Pages

To view sets of register pages, use the keys described below:

Command	Description
[0]	Display registers 0 to 99
[1]	Display registers 1000 to 1099
[2]	Display registers 2000 to 2099

And so on. The total number of register pages available to view depends on your module's configuration.

Displaying the Current Page of Registers Again



```
DATABASE DISPLAY 0 TO 99 (DECIMAL)
     100     101     102       4       5       6       7       8       9      10
      11      12      13      14      15      16       0       0       0       0
       0       0       0       0       0       0       0       0       0       0
       0       0       0       0       0       0       0       0       0       0
       0       0       0       0       0       0       0       0       0       0
       0       0       0       0       0       0       0       0       0       0
       0       0       0       0       0       0       0       0       0       0
       0       0       0       0       0       0       0       0       0       0
       0       0       0       0       0       0       0       0       0       0
       0       0       0       0       0       0       0       0       0       0
```

This screen displays the current page of 100 registers in the database.

Moving Back Through 5 Pages of Registers

Press **[-]** from the Database View menu to skip back to the previous 500 registers of data.

Viewing the Previous 100 Registers of Data

Press **[P]** from the Database View menu to display the previous 100 registers of data.

Skipping 500 Registers of Data

Hold down **[Shift]** and press **[=]** to skip forward to the next 500 registers of data.

Viewing the Next 100 Registers of Data

Press **[N]** from the Database View menu to select and display the next 100 registers of data.

Viewing Data in Decimal Format

Press **[D]** to display the data on the current page in decimal format.

Viewing Data in Hexadecimal Format

Press **[H]** to display the data on the current page in hexadecimal format.

Viewing Data in Floating Point Format

Press **[F]** from the Database View menu. Use this command to display the data on the current page in floating point format. The program assumes that the values are aligned on even register boundaries. If floating-point values are not aligned as such, they are not displayed properly.

Viewing Data in ASCII (Text) Format

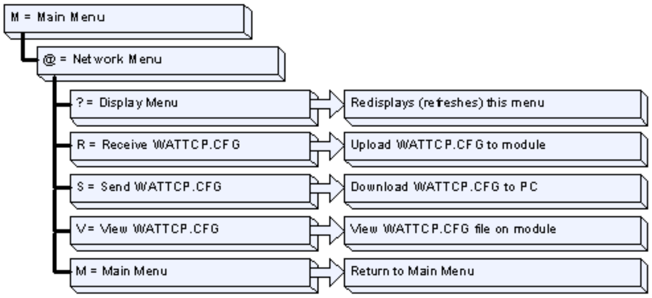
Press **[A]** to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Returning to the Main Menu

Press **[M]** to return to the Main Menu.

4.1.6 Network Menu

The network menu allows you to send, receive and view the WATTCP.CFG file that contains the IP and gateway addresses, and other network specification information.



Transferring WATTCP.CFG to the module

Press **[R]** to transfer a new WATTCP.CFG file from the PC to the module. Use this command to change the network configuration for the module (for example, the module's IP address).

Press **[Y]** to confirm the file transfer, and then follow the instructions on the terminal screen to complete the file transfer process.

Transferring WATTCP.CFG to the PC

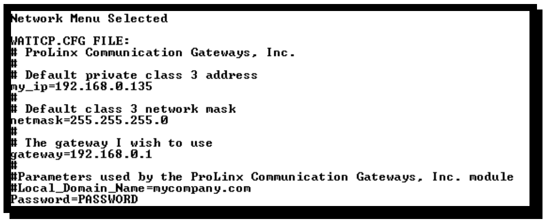
Press **[S]** to transfer the WATTCP.CFG file from the module to your PC.

Press **[Y]** to confirm the file transfer, and then follow the instructions on the terminal screen to complete the file transfer process.

After the file has been successfully transferred, you can open and edit the file to change the module's network configuration.

Viewing the WATTCP.CFG file on the module

Press **[V]** to view the module's WATTCP.CFG file. Use this command to confirm the module's current network settings.



```
Network Menu Selected

WATTCP.CFG FILE:
# ProLinx Communication Gateways, Inc.
#
# Default private class 3 address
my_ip=192.168.0.135
#
# Default class 3 network mask
netmask=255.255.255.0
#
# The gateway I wish to use
gateway=192.168.0.1
#
#Parameters used by the ProLinx Communication Gateways, Inc. module
#Local_Domain_Name=mycompany.com
Password=PASSWORD
```

Returning to the Main Menu

Press **[M]** to return to the Main Menu.

4.2 LED Status Indicators

The LEDs indicate the module's operating status as follows:

ProSoft Module	Color	Status	Indication
CFG	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
P1	Green	On	Port not used in application
		Off	Port not used in application
P2	Green	On	Port not used in application
		Off	Port not used in application
APP	Amber	Off	The MVI46-DNPSNET is working normally.
		On	The MVI46-DNPSNET module program has recognized a communication error on one of its DNP ports.
BP ACT	Amber	On	The LED is on when the module is performing a write operation on the backplane.
		Off	The LED is off when the module is performing a read operation on the backplane. Under normal operation, the LED should blink rapidly on and off.
OK	Red/ Green	Off	The card is not receiving any power and is not securely plugged into the rack.
		Green	The module is operating normally.
		Red	The program has detected an error or is being configured. If the LED remains red for over 10 seconds, the program has probably halted.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or battery is not present. Allow battery to charge by keeping module plugged into rack for 24 hours. If BAT LED still does not go off, contact ProSoft Technology, as this is not a user serviceable item.

If the APP, BP ACT and OK LEDs blink at a rate of every one-second, this indicates a serious problem with the module. Call ProSoft Technology support to arrange for repairs.

In addition to these LEDs, the module contains two LEDs under the module's door. The LED on the left (green) displays the link status. If the module is connected properly to a hub, this LED should be lit. The LED on the right (amber) is the data indication LED. Whenever the module is sending or receiving data on the Ethernet interface, the LED is illuminated.

4.2.1 Ethernet LED Indicators

LED	State	Description
Data	Off	No activity on the port.
	Green Flash	The port is either actively transmitting or receiving data.
Link	Off	No connection to hub or network is detected.
	Green Solid	Connected to hub or network correctly. This is the normal operating state.

4.2.2 Clearing a Fault Condition

Typically, if the OK LED on the front of the module turns red for more than ten seconds, a hardware problem has been detected in the module, or the program has exited.

To clear the condition, follow these steps:

1 Turn off power to the rack
2 Remove the card from the rack
3 Verify that all jumpers are set correctly
4 If the module requires a Compact Flash card, verify that the card is installed correctly
5 Re-insert the card in the rack and turn the power back on
6 Verify the configuration data being transferred to the module from the SLC processor.

If the module's OK LED does not turn green, verify that the module is inserted completely into the rack. If this does not cure the problem, contact ProSoft Technology Support.

4.2.3 Troubleshooting

Use the following troubleshooting steps if you encounter problems when the module is powered up. If these steps do not resolve your problem, please contact ProSoft Technology Technical Support.

Processor Errors

Problem Description	Steps to take
Processor Fault	Verify that the module is plugged into the slot that has been configured for the module. Verify that the slot in the rack configuration has been set up correctly in the ladder logic.
Processor I/O LED flashes	This indicates a problem with backplane communications. Verify that all modules in the rack are configured in the ladder logic.

Module Errors

Problem Description	Steps to take
BP ACT LED remains off or blinks slowly	This indicates that backplane transfer operations are failing. Connect to the module's Configuration/Debug port to check this. To establish backplane communications, verify the following items: ▪ The processor is in Run mode. ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write block data transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is configured in the processor.
OK LED remains red	The program has halted or a critical error has occurred. Connect to the Configuration/Debug port to see if the module is running. If the program has halted, turn off power to the rack, remove the card from the rack and re-insert the card in the rack, and then restore power to the rack.

4.2.4 Error Status Table

The program maintains an error/status table that is transferred to the processor in each read block. Ladder logic should be programmed to accept this block of data and place it in the module's controller tag. You can use the error/status data to determine the "health" of the module.

The data in the block is structured as shown in the following table.

Word	Variable Name	Description
0	Scan Counter	Program scan counter incremented each time the program loop is executed.
1 to 2	Product Name (ASCII)	These two words contain the product name of the module in ASCII format.
3 to 4	Revision (ASCII)	These two words contain the product revision level of the firmware in ASCII format.
5 to 6	Operating System Revision (ASCII)	These two words contain the module's internal operating system revision level in ASCII format.
7 to 8	Production Run Number (ASCII)	These two words contain the production 'batch' number for the particular chip in the module in ASCII format.
9	Read Block Count	Total number of blocks transferred from the module to the processor.
10	Write Block Count	Total number of blocks transferred from the processor to the module.
11	Parse Block Count	Total number of blocks parsed by the module that were received from the processor.
12	Block number error	Number of BTW requests that resulted in an incorrect BTW identification code.
13	DNP Slave Port total number of message frames received by slave	This value represents the total number of message frames that have matched this slaves address on this port. This count includes message frames which the slave may or may not be able to parse and respond.

Word	Variable Name	Description
14	DNP Slave Port total number of response message frames sent from slave	This value represents the number of good (non-error) responses that the slave has sent to the master on this port. The presumption is that if the slave is responding, the message was good. Note: This is a frame count.
15	DNP Slave Port total number of message frames seen by slave	This value represents the total number of message frames received by the slave, regardless of the slave address.
16	DNP Slave synchronization error count (Physical Layer Error)	This value counts the number of times a sync error occurs. The error occurs when extra bytes are received before the start bytes (0x05 and 0x64) are received.
17	DNP Slave overrun error count Layer Error) (Physical	This value counts the number of times the overrun error occurs. This error occurs when the mainline Data Link Layer routine cannot read the data received on the communication port before it is overwritten.
18	DNP Slave length error count (Physical Layer Error)	This value counts the number of times an invalid length byte is received. If the length of the message does not match the length value in the message, this error occurs.
19	DNP Slave bad CRC error (Data Link Layer Error)	This value counts the number of times a bad CRC value is received in a message.
20	DNP Slave user data overflow error (Transport Layer Error)	This value counts the number of times the application layer receives a message fragment buffer which is too small.
21	DNP Slave sequence error (Transport Layer Error)	This value counts the number of times the sequence numbers of multi-frame request fragments do not increment correctly.
22	DNP Slave address error (Transport Layer Error)	This value counts the number of times the source addresses contained in a multi-frame request fragments do not match.
23	DNP Slave Binary Input Event count	This value contains the total number of binary input events which have occurred.
24	DNP Slave Binary Input Event count in buffer	This value represents the number of binary input events which are waiting to be sent to the master.
25	DNP Slave Analog Input Event count	This value contains the total number of analog input events which have occurred.
26	DNP Slave Analog Input Event count in buffer	This value represents the number of analog input events which are waiting to be sent to the master.
27	DNP Slave bad function code error (Application Layer Error)	This value counts the number of times a bad function code for a selected object/variation is received by the slave device.
28	DNP Slave object unknown error (Application Layer Error)	This value counts the number of times a request for an unsupported object is received by the slave device.
29	DNP Slave out of range error (Application Layer Error)	This value counts the number of times a parameter in the qualifier, range or data field is not valid or out of range.
30	DNP Slave message overflow error (Application Layer Error)	This value counts the number of times an application response message from the slave is too long to transmit.

Word	Variable Name	Description
31	DNP Slave multi-frame message from DNP Master error (Application Layer Error)	This value counts the number of times the slave receives a multi-frame message from the master. The application does not support multi-frame master messages.
32 Free Memory MSB	Free Memory LSB	Free memory in module

5 Reference

In This Chapter

5.1 Product Specifications

The MVI46 DNP 3.0 Server over Ethernet Communications Module supports the implementation of the DNP 3.0 (Distributed Network Protocol) over Ethernet, allowing SLC processors to easily communicate with host systems supporting the protocol. The module supports DNP Subset Level 2 features and some Level 3 features.

5.1.1 Features and Benefits

The MVI46-DNPSNET (Distributed Network Protocol Module over Ethernet) allows SLC processors to easily communicate with other DNP protocol-compatible devices.

The module supports DNP subset level 2 features and some Level 3 features. The MVI46-DNPSNET module acts as an input/output module between the DNP Ethernet network and the SLC backplane. The data transfer from the SLC processor is asynchronous from the actions on the DNP network. Databases are defined in the module to house the data required by the protocol and transferred using M0/M1 files.

5.1.2 General Specifications

- Single Slot - 1746 backplane compatible (Local or extended I/O rack only. Remote rack not supported)
- The module is recognized as an Input/Output module and has access to processor memory for data transfer between processor and module using M0/M1 files

- Ladder Logic is used for data transfer between module and processor. Sample ladder file included
- Configuration data obtained from configuration text file downloaded to module. Sample configuration file included

5.1.3 Hardware Specifications

Specification	Description
Backplane Current Load	800 ma @ 5V (from backplane)
Operating Temperature	0 to 60°C (32 to 140°F)
Storage Temperature	-40 to 85°C (-40 to 185°F)
Relative Humidity	5% to 95% (non-condensing)
Shock	30g operational, 50g non-operational
Vibration	5 g from 10150 Hz
Processor	Compatible with Rockwell Automation SLC 5/02 M0/M1 capable processors or newer
LED indicators	Module status, Backplane transfer status, Application status, Serial activity (debug port), Ethernet link and activity, and error LED status
Debug/Configuration port (CFG)	
CFG Port (CFG)	RJ45 (DB-9M with supplied cable) RS-232 only No hardware handshaking
Configuration Connector	RJ45 RS-232 Connector (RJ45 to DB-9 cable shipped with unit)
Application Ports	
Ethernet Port (Ethernet Modules)	RJ45 Connector Link and activity LED indicators Electrical Isolation 1500 V rms at 50 Hz to 60 Hz for 60 s, 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

5.1.4 Functional Specifications

The MVI46-DNPSNET module accepts DNP commands to control and monitor the data stored in the DNP databases. This data is passed between the module and the SLC processor over the backplane for use in user applications.

- DNP databases to house data for the slave port supporting the following maximum input counts
  - Binary input: 8000 points (500 words)
  - Binary output: 8000 points (500 words)
  - Counter: 250 (500 words)
  - Analog input: 500
  - Analog output: 500
- User-definable module memory usage
- Data movement between module using M0/M1 data files

- Ethernet port supporting both TCP and UDP over Ethernet
- Supports DNP 3.0 in a level 2 implementation
- Supports sending of input event data from the ladder to the module
- Supports time synchronization from/to processor
- Configurable via text file
- Status and error information
- All data in the DNP slave is contained in user-defined files

5.2 Functional Overview

This section provides an overview of how the MVI46-DNPSNET module transfers data using the DNPSNET protocol. You should understand the important concepts in this chapter before you begin installing and configuring the module.

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.

5.2.1 General Concepts

The following discussion explains several concepts that are important for understanding the operation of the MVI46-DNPSNET module.

Module Power Up

On power up the module begins performing the following logical functions:

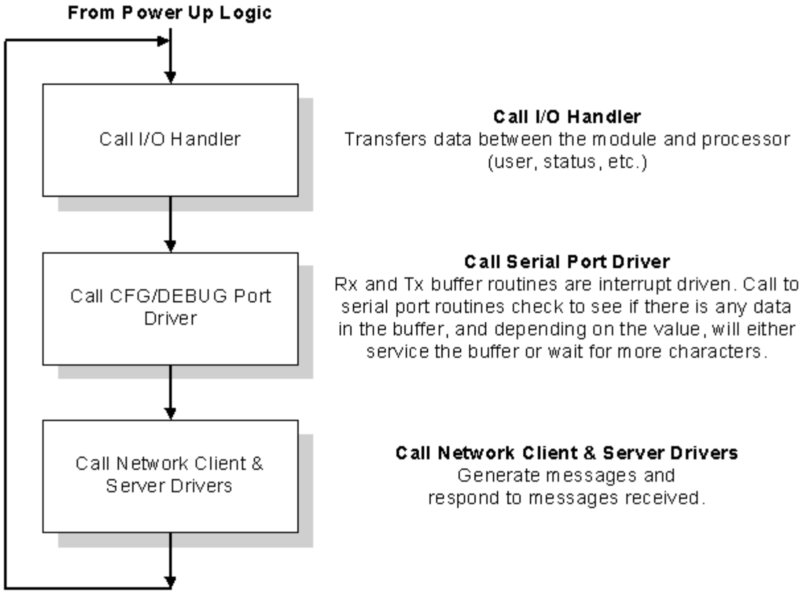
1 Initialize hardware components
- Install packet driver for Ethernet network interface and TCP/IP stack
- Initialize SLC backplane driver
- Test and clear all RAM
- Initialize the serial communication port

2 Read configuration file from Compact Flash Disk
3 Allocate and initialize module databases
4 Enable Slave Driver

After the module has received the configuration, the module will begin communicating with other nodes on the network, depending on the configuration.

Main Logic Loop

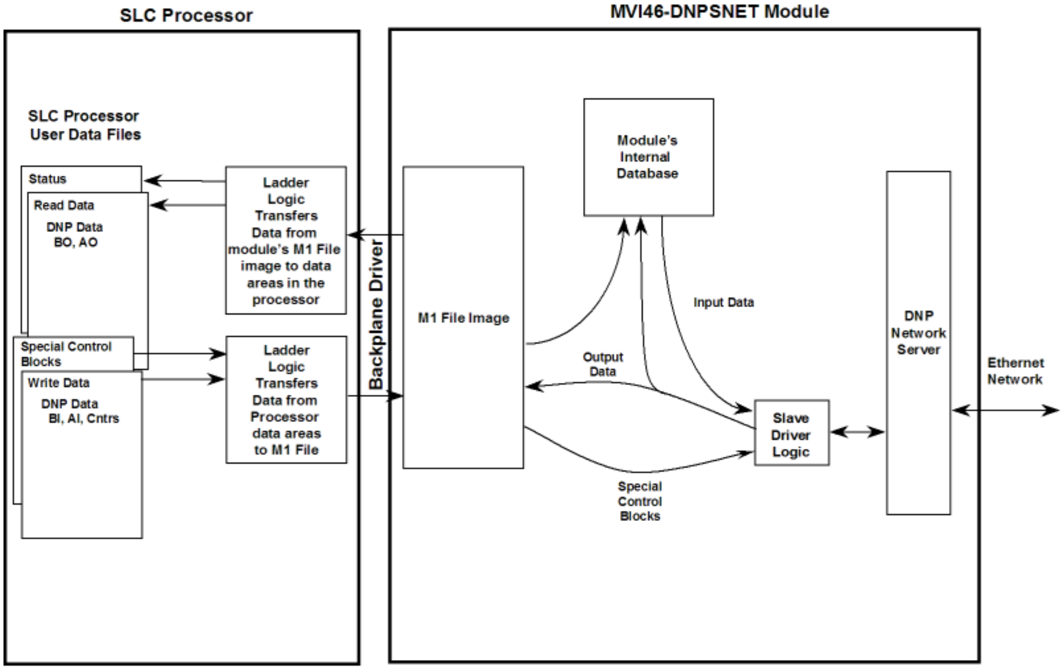
Upon completing the power up configuration process, the module enters an infinite loop that performs the functions shown in the following diagram.


From Power Up Logic
Call I/O Handler
Call CFG/DEBUG Port Driver
Call Network Client & Server Drivers
Call I/O Handler
Transfers data between the module and processor (user, status, etc.)
Call Serial Port Driver
Rx and Tx buffer routines are interrupt driven. Call to serial port routines check to see if there is any data in the buffer, and depending on the value, will either service the buffer or wait for more characters.
Call Network Client & Server Drivers
Generate messages and respond to messages received.

Backplane Data Transfer

The MVI46-DNPSNET module communicates directly over the SLC backplane. All data for the module is contained in the module's M1 file. Data is moved between the module and the SLC processor across the backplane using the module's M1 file. The SLC scan rate and the communication load on the module determine the update frequency of the M1 file. The COP instruction can be used to move data between user data files and the module's M1 file.

The following illustration shows the data transfer method used to move data between the SLC processor, the MVI46-DNPSNET module and the DNP Network.



All data transferred between the module and the processor over the backplane is through the M1 file. Ladder logic must be written in the SLC processor to interface the M1-file data with data defined in the user-defined data files in the SLC. All data used by the module is stored in its internal database.

All data used by the module is stored in its internal databases. These databases are defined as a virtual DNP data tables with addresses from 0 to the maximum number of points for each data type.

The following illustration shows the layout of the databases:

DATA AREA	
DNP DATA	BINARY INPUTS
	ANALOG INPUTS
	COUNTER DATA
	BINARY OUTPUTS
	ANALOG OUTPUTS
FROZEN DATA	FROZEN COUNTER DATA
LAST VALUE DATA	BINARY INPUTS
	ANALOG INPUTS
EVENT DATA	BINARY INPUT EVENTS
	ANALOG INPUT EVENTS

Data contained in this database is constantly updated with the M1 file data by the module and requires no SLC ladder logic to implement. A reserved word in the M1 and database, address 8900, is used as a control register for module control by the ladder logic and for control of the ladder logic by the module.

If the processor places a value of 9998 in this register, the module will perform a warm-boot operation. If the processor places a value of 9999 in this register, the module will perform a cold-boot operation. In this application module, both of these operations perform the same function. They exit the program and then restart the program. Many of the program parameters set in the user configuration must be set at program initialization and cannot be set while the program is running. Therefore, both functions operate the same.

The module can be configured to have the output data in the module set to that stored in the SLC processor during program initialization. This feature requires ladder logic support. When the module performs a restart operation, it will set the output area of the M1 file to the current values stored in the processor. The module requests this action by placing a value of 1000 in the control register. After the ladder logic has completely built the M1 file image, it will set the control register to a value of 1001. This informs the module that the initialization is complete and the program can continue with the startup procedure.

Command Control

Command control instructions control the module or request special data from the module. These blocks of data are placed in the command control portion of the module's M1-file. This data area starts at address 8900 and has a size of 64 words. The current version of the software supports several command control blocks each of which is discussed in the following topics:

Block 9958 - SLC Binary Input Event

If the SLC sends a block 9958, the module will place the binary input event data in the block into the event buffer and alter the data values for the points in the DNP binary input database. The format for the event message is shown below.

Word Offset in M1 File	Data Field(s)	Description
8900	Block ID	This field contains the value of 9958 identifying the event block to the module.
8901	Event Count	This field contains the number of events contained in the block. Valid values for this field are 1 to 12.
8902	Sequence Counter	This field holds the sequence counter for each 9958 block transfer. This synchronizes and confirms receipt of the block by the module.
8903	DNP Binary Input Data point	This is the data point in the DNP binary input database represented by the event.
8904	Month/Day/State	Formatted: bits 0 to 4 = Day, bits 8 to 11 = Month, bit 15 = digital state for point. All other bits are ignored.
8905	Hour/Minute	Formatted: bits 0 to 5 = Minutes, bits 8 to 12 = Hour. All other bits are ignored.
8906	Sec/Millisecond	Formatted: bits 0 to 9 = Milliseconds, bits 10 to 15 = Seconds.

Word Offset in M1 File	Data Field(s)	Description
8907	Year	This is the four digit year for the event.
8908 to 8912		Five words of data for Event #2.
8913 to 8917		Five words of data for Event #3.
8918 to 8922		Five words of data for Event #4.
8923 to 8927		Five words of data for Event #5.
8928 to 8932		Five words of data for Event #6.
8933 to 8937		Five words of data for Event #7.
8938 to 8942		Five words of data for Event #8.
8943 to 8947		Five words of data for Event #9.
8948 to 8952		Five words of data for Event #10.
8953 to 8957		Five words of data for Event #11.
8958 to 8962		Five words of data for Event #12.

Up to 12 events can be passed from the SLC to the module in each block. To insure that the command block reached the module and was processed, the module will place the following data in the M1 file.

Word Offset in M1 File	Data Field(s)	Description
8900	Complete	This field contains the value of 0 to indicate that the command has been completed by the module.
8901	Block ID	Identification code for block set to 9958.
8902	Event Count	This field contains the number of events processed by the module.
8903	Sequence Counter	This field contains the sequence counter of the last successful block 9958 received.

The sequence counter field in the returned block is set to the last successfully processed block 9958 from the SLC. Compare this value to that sent by the SLC. If the values match, the events can be removed from the SLC. If the values do not match, or the SLC does not receive a 9958 block, the SLC must re-send the block.

Block 9959 - SLC Analog Input Event

If the SLC sends a block 9959, the module will place the analog input event data in the block into the event buffer and alter the data values for the points in the DNP analog input database. The format for the event message is shown below.

Word Offset in M1 File	Data Field(s)	Description
8900	Block ID	This field contains the value of 9959 identifying the event block to the module.
8901	Event Count	This field contains the number of events contained in the block. Valid values for this field are 1 to 10.
8902	Sequence Counter	This field holds the sequence counter for each 9959 block transfer. This synchronizes and confirms receipt of the block by the module.
8903	DNP Analog Input Data point	This is the data point in the DNP analog input database represented by the event.

Word Offset in M1 File	Data Field(s)	Description
8904	Analog Input Value	This is the new analog input value represented in the event.
8905	Month/Day	Formatted: bits 0 to 4 = Day, bits 8 to 11 = Month. All other bits are ignored.
8906	Hour/Minute	Formatted: bits 0 to 5 = Minutes, bits 8 to 12 = Hour. All other bits are ignored.
8907	Sec/Millisecond	Formatted: bits 0 to 9 = Milliseconds, bits 10 to 15 = Seconds.
8908	Year	Four digit year value for event.
8909 to 8914		Six words of data for Event #2.
8915 to 8920		Six words of data for Event #3.
8921 to 8926		Six words of data for Event #4.
8927 to 8932		Six words of data for Event #5.
8933 to 8938		Six words of data for Event #6.
8939 to 8944		Six words of data for Event #7.
8945 to 8950		Six words of data for Event #8.
8951 to 8956		Six words of data for Event #9.
8957 to 8962		Six words of data for Event #10.

Up to 10 events can be passed from the SLC to the module in each block. To insure that the block reached the module and was processed, the module will place the following information in the M1 file.

Word Offset in M1 File	Data Field(s)	Description
8900	Complete	This field contains the value of 0 to indicate that the command has been completed by the module.
8901	Block ID	Identification code for block set to 9959.
8902	Event Count	This field contains the number of events processed by the module.
8903	Sequence Counter	

The sequence counter field in the returned block is set to the last successfully processed block 9959 from the SLC. Compare this value to that sent by the SLC. If the values match, the events can be removed from the SLC. If the values do not match, or the SLC does not receive a 9959 block, the SLC must re-send the block.

Block 9970 - Set SLC Time Using Module Time

This block transfers the module's time to the SLC processor. Ladder logic must be used to set the processor's clock using the data received. The format of the block sent from the SLC has the following format:

Word Offset in M1 File	Data Field(s)	Description
8900	Block ID	This field contains the value of 9970 identifying the block type to the module.

The module responds to the request with the following block of data:

Word Offset in M1 File	Data Field(s)	Description
8900	Complete	This field contains the value of 0 to indicate that the command has been completed by the module.
8901	Block ID	This field contains the block identification code of 9970 for the block.
8902	Year	This field contains the four-digit year to be used with the new time value.
8903	Month	This field contains the month value for the new time. Valid entry for this field is in the range of 1 to 12.
8904	Day	This field contains the day value for the new time. Valid entry for this field is in the range of 1 to 31.
8905	Hour	This field contains the hour value for the new time. Valid entry for this field is in the range of 0 to 23.
8906	Minute	This field contains the minute value for the new time. Valid entry for this field is in the range of 0 to 59.
8907	Seconds	This field contains the second value for the new time. Valid entry for this field is in the range of 0 to 59.
8908	Milliseconds	This field contains the millisecond value for the new time. Valid entry for this field is in the range of 0 to 999.
8909	Remote Time Synchronization	This field informs the PLC if the date and time passed has been synchronized with a remote DNP master device on the module's slave port.

Block 9971 - Set Module's Time Using SLC Time

This block sets the clock in the module to match the clock in the SLC processor. If the SLC sends a block 9971, the module will set its time using the data contained the block. The format of the block is shown below:

Word Offset in M1 File	Data Field(s)	Description
8900	Block ID	This field contains the block identification code of 9971 for the block.
8901	Year	This field contains the four-digit year to be used with the new time value.
8902	Month	This field contains the month value for the new time. Valid entry for this field is in the range of 1 to 12.
8903	Day	This field contains the day value for the new time. Valid entry for this field is in the range of 1 to 31.
8904	Hour	This field contains the hour value for the new time. Valid entry for this field is in the range of 0 to 23.

Word Offset in M1 File	Data Field(s)	Description
8905	Minute	This field contains the minute value for the new time. Valid entry for this field is in the range of 0 to 59.
8906	Seconds	This field contains the second value for the new time. Valid entry for this field is in the range of 0 to 59.
8907	Milliseconds	This field contains the millisecond value for the new time. Valid entry for this field is in the range of 0 to 999.

The module will respond to a valid 9971 block with a block of data with the following format:

Word Offset in M1 File	Data Field(s)	Description
8900	Complete	This field contains the value of 0 to indicate that the command has been completed by the module.
8901	Block Write ID	This field contains the block identification code of 9971 for the block.

Block 9998 - Warm Boot Module

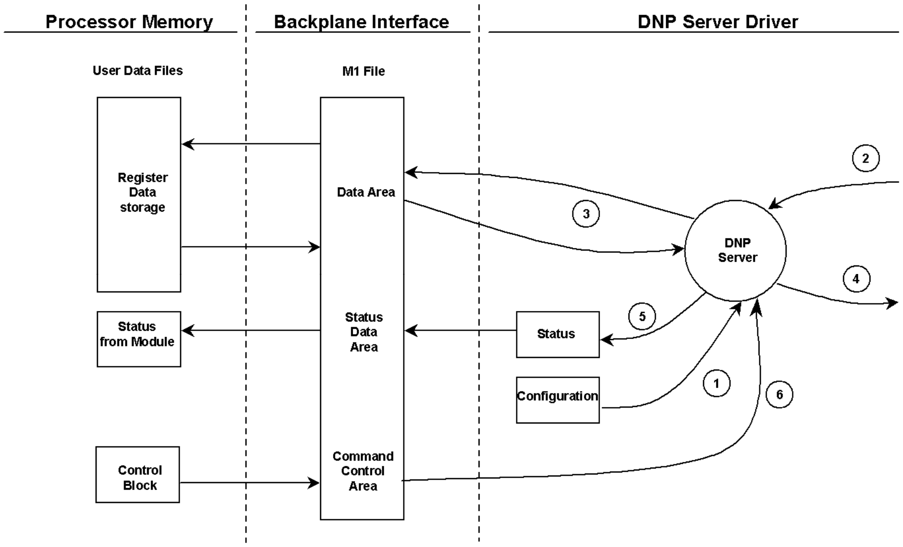
If the SLC processor sends a block number 9998, the module will perform a warm-boot operation.

Block 9999 - Cold Boot Module

If the SLC processor sends a block number 9999, the application performs the cold-boot operation. The module exits the program and performs a soft restart on the module.

Data Flow Between MVI46-DNPSNET Module and the SLC Processor

The following topics describe the flow of data between the two pieces of hardware (SLC processor and MVI46-DNPSNET module) and other nodes on the DNP network. The DNP Server Driver allows the MVI46-DNPSNET 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 configuration information for the module is retrieved from the DNPSNET.CFG file on the Compact Flash Disk. This information configures the module and define the Ethernet node characteristics.
2	A Host device (DNP Master unit) issues a read or write command to the module's node address. The 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 appropriate internal database in the module. If the command is a read command, the data is read out of the database and a response message is built. If the command is a write command, the data is written directly into the database and the M1 file, 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	Counters are available in the Status Block that permit the ladder logic program to determine the level of activity of the Slave Driver.
6	The module constantly monitors for command control blocks from the processor. If a valid block is received, the function is executed. Additionally, data is constantly being exchanged between the module and the processor.

Review the **Module Configuration** section for a complete list of the parameters that must be defined for a slave port.

5.3 Cable Connections

The MVI46-DNPSNET module has the following communication connections on the module:

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

5.3.1 Ethernet Connection

The MVI46-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 MVI46-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.

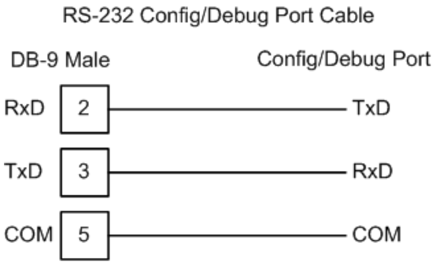
Ethernet Port Configuration - wattcp.cfg

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 using an ASCII terminal by selecting "@" (Network Menu) and "V" (View) options when connected to the Debug port.

```
# WATTCP.CFG FILE:
# ProSoft Technology.
  my_ip=192.168.0.100
# Default class 3 network mask
  netmask=255.255.255.0
# The gateway I wish to use
  gateway=192.168.0.1
```

5.3.2 RS-232 Configuration/Debug Port

This port is physically an RJ45 connection. An RJ45 to DB-9 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. The cable for communications on this port is shown in the following diagram:

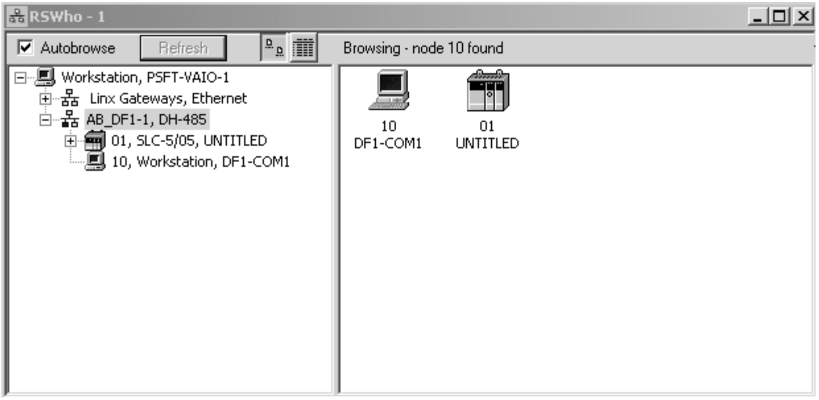


<u>Disabling the RSLinx Driver for the Com Port on the PC</u>

The communication port driver in RSLinx can occasionally prevent other applications from using the PC's COM port. If you are not able to connect to the module's configuration/debug port using ProSoft Configuration Builder (PCB), HyperTerminal or another terminal emulator, follow these steps to disable the RSLinx Driver.

1 Open RSLinx and go to Communications>RSWho

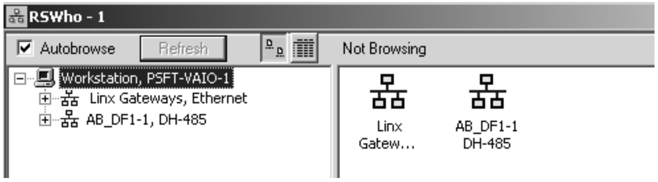
2 Make sure that you are not actively browsing using the driver that you wish to stop. The following shows an actively browsed network:



Browsing - node 10 found

```
⊟ Workstation, PSFT-VAIO-1
   ⊞ Linx Gateways, Ethernet
   ⊟ AB_DF1-1, DH-485
      ⊞ 01, SLC-5/05, UNTITLED
         10, Workstation, DF1-COM1
```

3 Notice how the DF1 driver is opened, and the driver is looking for a processor on node 1. If the network is being browsed, then you will not be able to stop this driver. To stop the driver your RSWho screen should look like this:



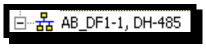
RSWho - 1

☑ Autobrowse

Not Browsing

⊟ Workstation, PSFT-VAIO-1
  ⊞ Linx Gateways, Ethernet
  ⊞ AB_DF1-1, DH-485

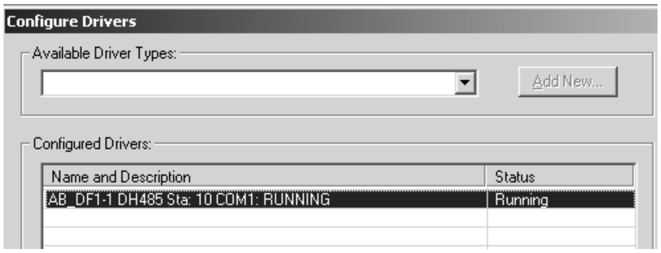
Branches are displayed or hidden by clicking on the ⊞ or the ⊟ icons.



⊟ AB_DF1-1, DH-485

4 When you have verified that the driver is not being browsed, go to

Communications>Configure Drivers
You may see something like this:



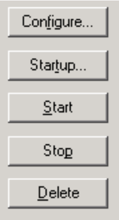
Configure Drivers

Available Driver Types:

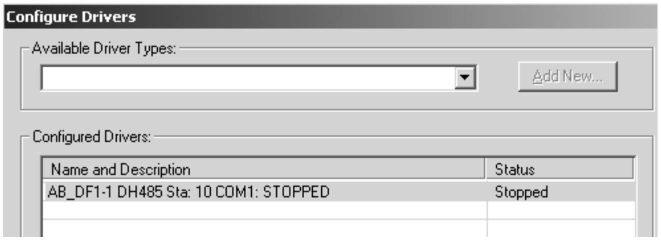
Configured Drivers:

Name and Description	Status
AB_DF1-1 DH485 Sta: 10 COM1: RUNNING	Running

If you see the status as running, you will not be able to use this com port for anything other than communication to the processor. To stop the driver press the "Stop" on the side of the window:



5 After you have stopped the driver you will see the following:



Configure Drivers

Available Driver Types:

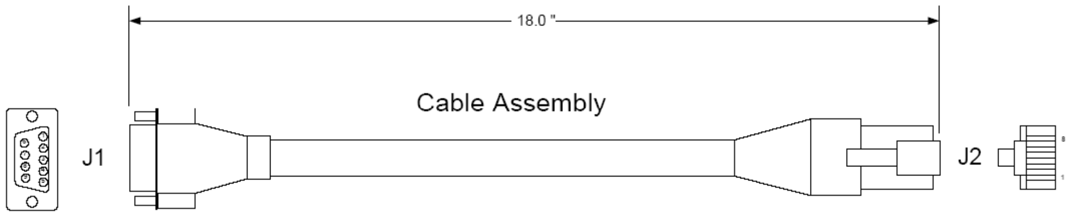
Configured Drivers:

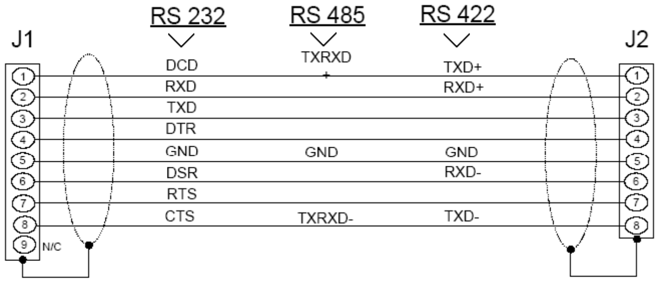
Name and Description	Status
AB_DF1-1 DH485 Sta: 10 COM1: STOPPED	Stopped

6 Upon seeing this, you may now use that com port to connect to the debug port of the module.

> **Note:** You may need to shut down and restart your PC before it will allow you to stop the driver (usually only on Windows NT machines). If you have followed all of the above steps, and it will not stop the driver, then make sure you do not have RSLogix open. If RSLogix is not open, and you still cannot stop the driver, then reboot your PC.

5.3.3 DB9 to RJ45 Adaptor (Cable 14)





Wiring Diagram

5.4 MVI46-DNPSNET Status Data

This section contains a listing of the MVI46-DNPSNET module's status data area.

Word	Variable Name	Description
0	Scan Counter	Program scan counter incremented each time the program loop is executed.
1 to 2	Product Name (ASCII)	These two words contain the product name of the module in ASCII format.
3 to 4	Revision (ASCII)	These two words contain the product revision level of the firmware in ASCII format.
5 to 6	Operating System Revision (ASCII)	These two words contain the module's internal operating system revision level in ASCII format.
7 to 8	Production Run Number (ASCII)	These two words contain the production 'batch' number for the particular chip in the module in ASCII format.
9	Read Block Count	Total number of blocks transferred from the module to the processor.

Word	Variable Name	Description
10	Write Block Count	Total number of blocks transferred from the processor to the module.
11	Parse Block Count	Total number of blocks parsed by the module that were received from the processor.
12	Block number error	Number of BTW requests that resulted in an incorrect BTW identification code.
13	DNP Slave Port total number of message frames received by slave	This value represents the total number of message frames that have matched this slaves address on this port. This count includes message frames which the slave may or may not be able to parse and respond.
14	DNP Slave Port total number of response message frames sent from slave	This value represents the number of good (non-error) responses that the slave has sent to the master on this port. The presumption is that if the slave is responding, the message was good. Note: This is a frame count.
15	DNP Slave Port total number of message frames seen by slave	This value represents the total number of message frames received by the slave, regardless of the slave address.
16	DNP Slave synchronization error count (Physical Layer Error)	This value counts the number of times a sync error occurs. The error occurs when extra bytes are received before the start bytes (0x05 and 0x64) are received.
17	DNP Slave overrun error count (Physical Layer Error)	This value counts the number of times the overrun error occurs. This error occurs when the mainline Data Link Layer routine cannot read the data received on the communication port before it is overwritten.
18	DNP Slave length error count (Physical Layer Error)	This value counts the number of times an invalid length byte is received. If the length of the message does not match the length value in the message, this error occurs.
19	DNP Slave bad CRC error (Data Link Layer Error)	This value counts the number of times a bad CRC value is received in a message.
20	DNP Slave user data overflow error (Transport Layer Error)	This value counts the number of times the application layer receives a message fragment buffer which is too small.
21	DNP Slave sequence error (Transport Layer Error)	This value counts the number of times the sequence numbers of multi-frame request fragments do not increment correctly.
22	DNP Slave address error (Transport Layer Error)	This value counts the number of times the source addresses contained in a multi-frame request fragments do not match.
23	DNP Slave Binary Input Event count	This value contains the total number of binary input events which have occurred.
24	DNP Slave Binary Input Event count in buffer	This value represents the number of binary input events which are waiting to be sent to the master.
25	DNP Slave Analog Input Event count	This value contains the total number of analog input events which have occurred.

Word	Variable Name	Description
26	DNP Slave Analog Input Event count in buffer	This value represents the number of analog input events which are waiting to be sent to the master.
27	DNP Slave bad function code error (Application Layer Error)	This value counts the number of times a bad function code for a selected object/variation is received by the slave device.
28	DNP Slave object unknown error (Application Layer Error)	This value counts the number of times a request for an unsupported object is received by the slave device.
29	DNP Slave out of range error (Application Layer Error)	This value counts the number of times a parameter in the qualifier, range or data field is not valid or out of range.
30	DNP Slave message overflow error (Application Layer Error)	This value counts the number of times an application response message from the slave is too long to transmit.
31	DNP Slave multi-frame message from DNP Master error (Application Layer Error)	This value counts the number of times the slave receives a multi-frame message from the master. The application does not support multi-frame master messages.
32	Free Memory MSB Free Memory LSB	Free memory in module

5.5 MVI46-DNPSNET Module Internal Indication Bits (IIN Bits) for DNP Server

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. The following description describes the word.

5.5.1 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	Not used
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.

5.5.2 *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.

5.6 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 8	Slave will return this variation
	2	Binary Input with Status			129, 130	00, 01		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

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	1	Control Relay Output Block	3, 4, 5, 6	17, 28	129	Echo of request	88	Slave will variation respond correctly to this
	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
	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

OBJECT			**REQUEST**		**RESPONSE**			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	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
	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

OBJECT			**REQUEST**		**RESPONSE**			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	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
30	0	Analog Input - All Variations	1	06			16	Slave will data respond with variation 4
	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

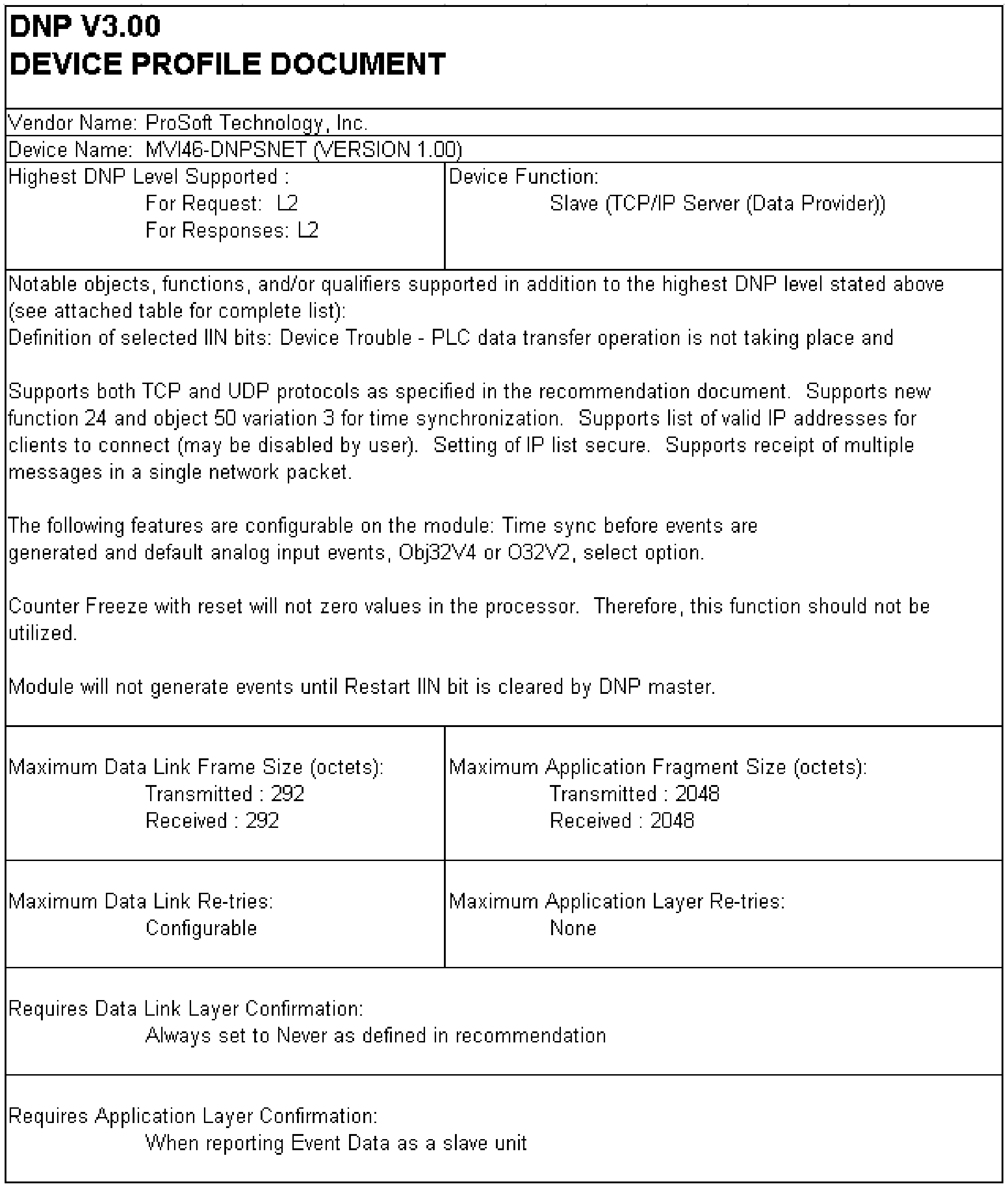
OBJECT			**REQUEST**		**RESPONSE**			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
31	0	Frozen Analog Input - All Variations						Slave will return Unknown Object to this request
	1	32-Bit Frozen Analog Input					40	Slave will return Unknown Object 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
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
	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
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

OBJECT			**REQUEST**		**RESPONSE**			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	4	16-Bit Frozen Analog Event With Time					72	Slave will return Unknown Object to this request
40	0	Analog Output Status - All Variations	1	06			24	Slave will return variation 2 data
	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
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
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 ith Date WInterval					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
	3	Date and Time at Last Recorded Time	2				48	Slave will process the data in this object for time synchronization.

OBJECT			**REQUEST**		**RESPONSE**			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
60	0	Not Defined						Not Defined in DNP
	1	Class 0 Data	1	06				Slave will respond to this variation with all static data
	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	

OBJECT			**REQUEST**		**RESPONSE**			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	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	
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 w ill return Obj 52, Var 2, Qual 7, Cnt 1
			24					Slave supports use of this new time synchronization function. Used with Obj 52, Var 3.

5.7 Device Profile



DNP V3.00
DEVICE PROFILE DOCUMENT

Vendor Name: ProSoft Technology, Inc.

Device Name: MVI46-DNPSNET (VERSION 1.00)

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

Notable objects, functions, and/or qualifiers supported in addition to the highest DNP level stated above (see attached table for complete list):
Definition of selected IIN bits: Device Trouble - PLC data transfer operation is not taking place and

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.

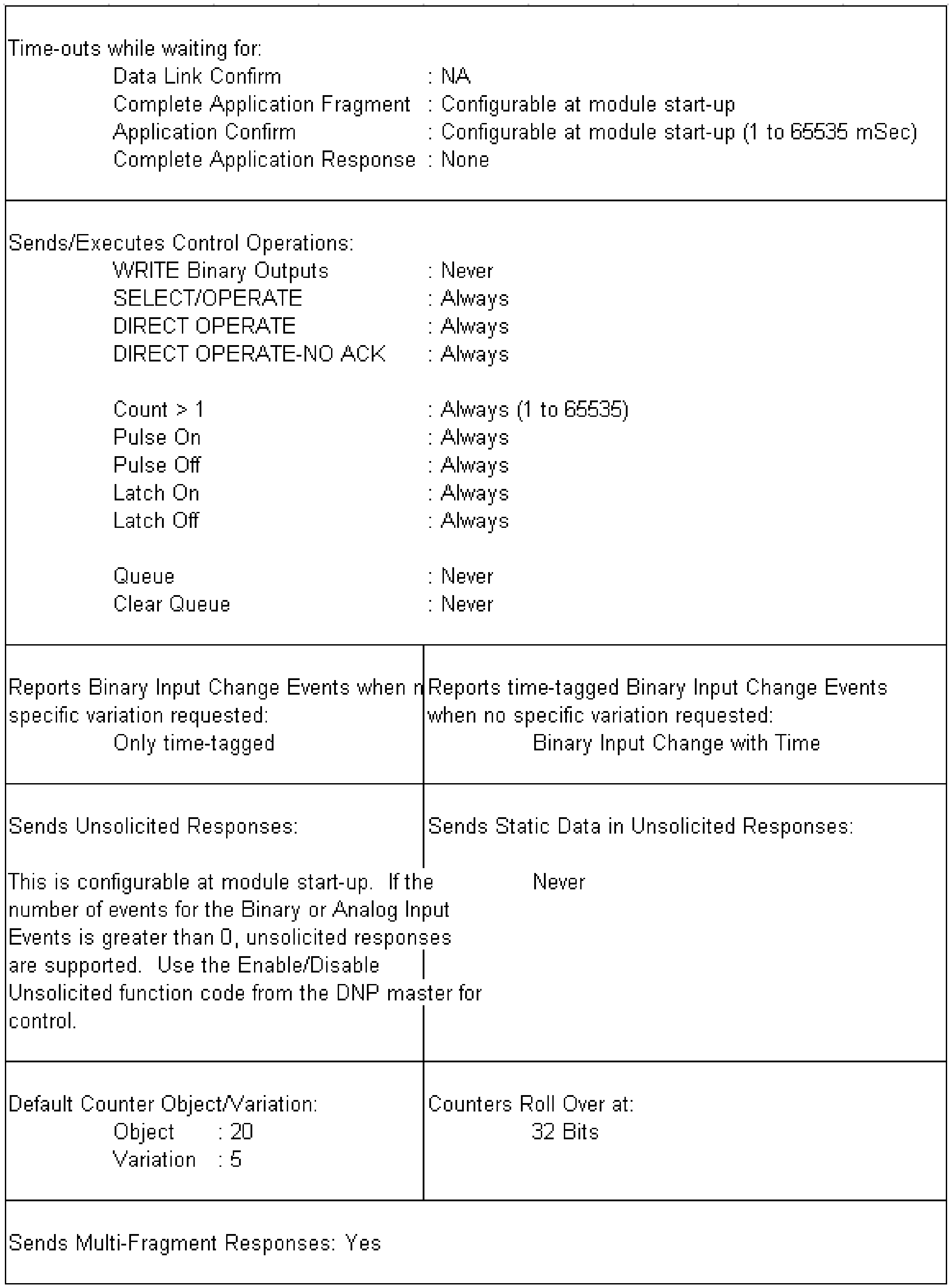
Counter Freeze with reset will not zero values in the processor. Therefore, this function should not be utilized.

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

Maximum Data Link Frame Size (octets): Transmitted : 292 Received : 292	Maximum Application Fragment Size (octets): Transmitted : 2048 Received : 2048
Maximum Data Link Re-tries: Configurable	Maximum Application Layer Re-tries: 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
Complete Application Fragment	: Configurable at module start-up
Application Confirm	: Configurable at module start-up (1 to 65535 mSec)
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 n specific variation requested:
Only time-tagged

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

Sends 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.

Sends Static Data in Unsolicited Responses:
Never

Default Counter Object/Variation:
Object : 20
Variation : 5

Counters Roll Over at:
32 Bits

Sends Multi-Fragment Responses: Yes

5.8 MVI46-DNPSNET Application Design

This documentation describes the MVI46-DNPSNET module configuration and setup as it applies to application design. Before attempting to implement this module with a DNP network, verify that the whole design of the system is complete. This includes definition of all the data types and point counts required for each type, all communication parameters required for the network including media type and the use of advanced features such as unsolicited messaging. These must be defined for all master and slave devices on the network. Additionally, the DNP Device Profiles and DNP Subset Definition documents for each device must be reviewed to make sure all the devices will interact on the network as expected. Failure to fully understand these important documents for all devices on the network will usually lead to many problems when implementing the design.

It is important to fully understand the DNP specification as outlined in the Basic Four Documents. These are available to users of the DNP users group. It is recommended that all users of the module have access to these important documents as they define the DNP data types, functions and variations. It will be very difficult to implement the module without an understanding of the protocol and the rules that are defined in the specification. Additionally, potential users should review the DNP Subset and Conformance Test documents and the document that discusses DNP protocol support on Ethernet using the UDP and TCP protocols. These documents provide auxiliary information on the protocol. All of these documents are available to members of the DNP User Group at http://www.dnp.org (http://www.dnp.org). Please check this site for other important information regarding the DNP protocol.

5.8.1 Design

In order to implement a solution using the module, the SLC processor must be set up using predefined user data structures. The data transfer interface requires ladder logic in order to interface data in the module with that in the processor. The program required for data transfer is developed in ladder and is discussed in the **Module Set Up** section of this manual. This program will interact with the module by sending and receiving data and issuing special control commands.

User data files in the SLC processor contain the data to be used by the module and the configuration information is stored in the text file, DNPSNET.CFG, stored on the module's Compact Flash Disk. Before you generate the program or layout the data files, you must first design your system. Time spent doing system design at the outset of the project will greatly enhance the success and ease of development of the project.

Designing the system

System design defines the data requirements of the system, communication parameters, and module functionality. The application developer should refer to the person responsible for the DNP master and slave device configurations to verify that the functionality and data types required for the whole system are consistent. Review the DNP Device Profile and DNP Subset documentation for a definition of the level of DNP support offered by the module.

The following topics describe each element of system design.

Data Requirements

This phase of design defines what data elements are to be interfaced in the SLC processor with the DNP master. The module provides the following data types: digital input, digital output, counter, analog input and analog output. All communications between the DNP master and the SLC is through these data types. Therefore, all data to be used by the system must be contained and configured in one of these data types.

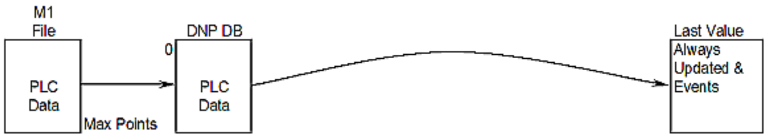
The following illustration shows the databases maintained by the module for the DNP data.

DATA AREA	
DNP DATA	BINARY INPUTS
	ANALOG INPUTS
	COUNTER DATA
	BINARY OUTPUTS
	ANALOG OUTPUTS
FROZEN DATA	FROZEN COUNTER DATA
LAST VALUE DATA	BINARY INPUTS
	ANALOG INPUTS
EVENT DATA	BINARY INPUT EVENTS
	ANALOG INPUT EVENTS

The module is responsible for maintaining the databases using data acquired from the SLC and DNP master attached network port.

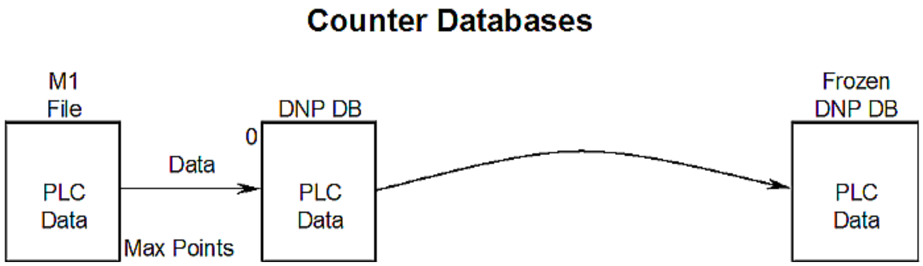
The following illustration shows the interaction of the binary and analog input points with the databases.

Binary and Analog Input Databases



All data for these data types is derived from the processor and is passed to the module over the backplane. The module will constantly monitor for changes in this data and generate event messages when point values change. For binary input points, events will be generated on any state change. For analog input points, events will be generated for points that have a current value outside of the user-set deadband based on the last value used for an event.

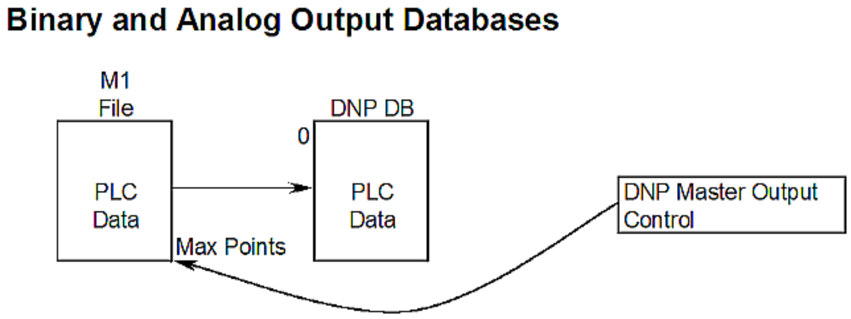
The following illustration shows the interaction of the counter points with the databases.



This data is constantly sourced from the processor and placed in the module's internal database. This information is available to the remote master for monitoring. When the module receives a freeze command from the master unit, it will copy the current counter values into the frozen counter database area. The remote master can then monitor this information. If the module receives a counter freeze with reset command, the current counter values will be passed to the frozen counter database and only the module's values will be set to 0.

Note: This data is not sent to the controller, and the zero data will be overwritten by the counter data contained in the controller. Therefore, the freeze with reset should not be used with this module. The results will not be as expected. There is no way to guarantee that counts will not be lost during the reset step in the module and controller. As a result, this feature was not implemented in the module.

The following illustration shows the interaction of the binary and analog output points with the databases.



Output data is sourced from the controlling master station and passed directly to the processor over backplane from the module and module's internal DNP database. These data are used in the ladder logic to control operations and I/O in the processor.

A more detailed discussion of each data type is given in the following topics.

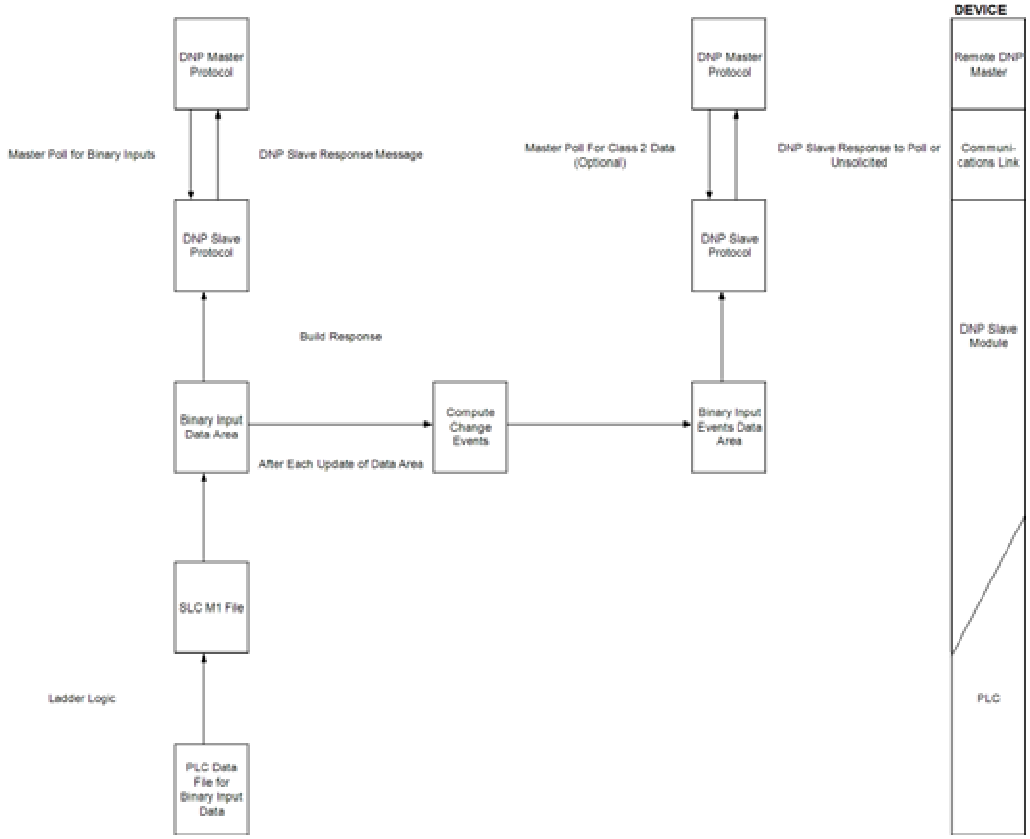
DNP Digital Input Data

This data type stores the binary value of 1 or 0. The size of this data area is determined from the configuration parameter Binary Inputs (number of words, each containing 16 binary input points). These data are transferred to the module's M1 file from user data files in the SLC. Therefore, these data are read-only for the module and the DNP master unit communicating with the module. The module continuously copies the binary input data of the M1 file into the module's DNP database. This data is constantly monitored for changes in state of any of the contained data. If there is a change in any of the data, the module will generate an event message for the points that change.

The remote DNP master unit can read the current status data and the event data from the module. Event messages generated by the module can be retrieved using a poll for Class 2 data, as all digital input events are considered a Class 2 data type. If unsolicited message generation is enabled in the application, the events will automatically be sent by the module to the DNP master unit when the maximum event count for Class 2 data is reached or when the timeout for unsolicited messages is exceeded.

A data flow diagram for the digital input data is shown in the following figure.

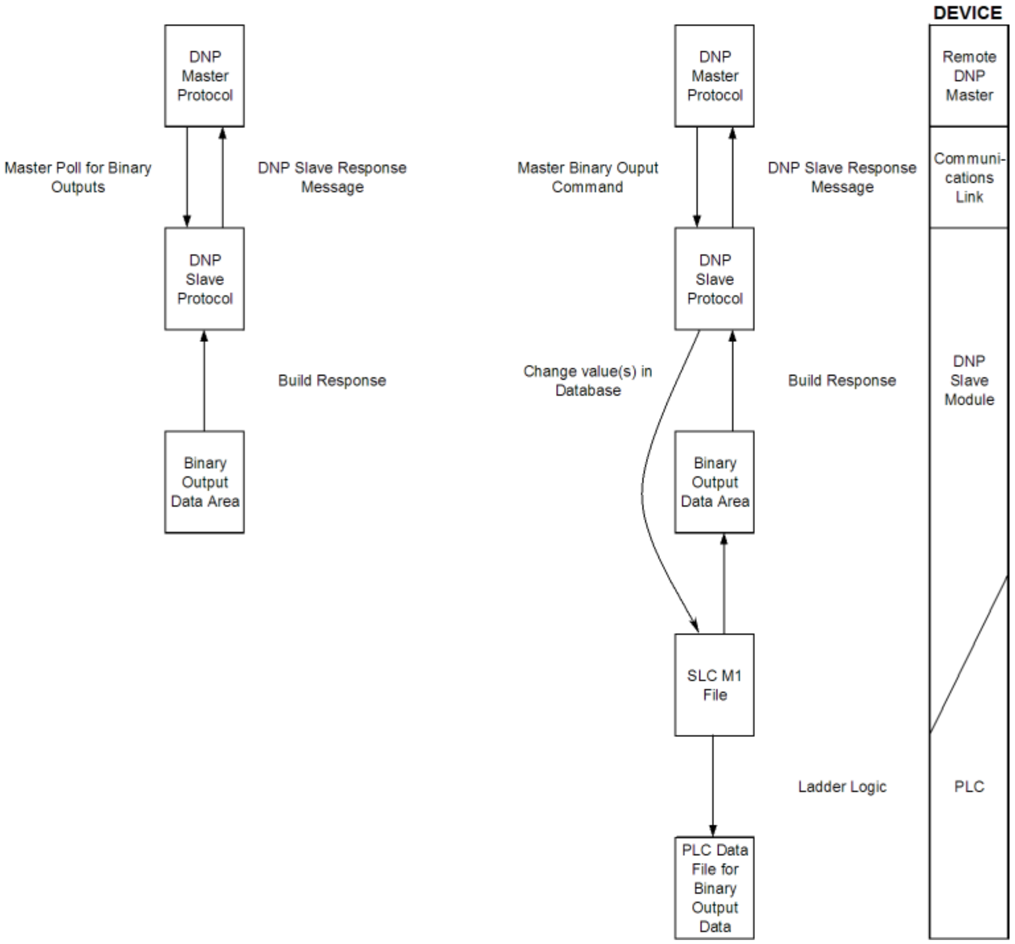
Binary Input Data Flow Diagram


DEVICE
DNP Master Protocol
Master Poll for Binary Inputs
DNP Slave Response Message
DNP Master Protocol
Master Poll For Class 2 Data (Optional)
DNP Slave Response to Poll or Unsolicited
Remote DNP Master
Communi-cations Link
DNP Slave Protocol
DNP Slave Protocol
Build Response
DNP Slave Module
Binary Input Data Area
Compute Change Events
Binary Input Events Data Area
After Each Update of Data Area
SLC M1 File
Ladder Logic
PLC
PLC Data File for Binary Input Data

DNP Digital Output Data

This data type stores digital control and command state data received from the DNP master unit with a value of 1 or 0. The size of this data area is determined from the configuration parameter Binary Outputs (defines number of words, each containing 16 binary output points). These data are transferred directly from the DNP driver to the M1 file when a command is received. A data flow diagram for the digital output data is shown below.

Binary Output Data Flow Diagram



The dual-point relay control database (Trip/Close) is overlaid on the DNP Binary Output database of the module. Each DNP point index sent has 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, our module will not require any configuration for the dual-point control. And, our module will be backward compatible for current customer applications.

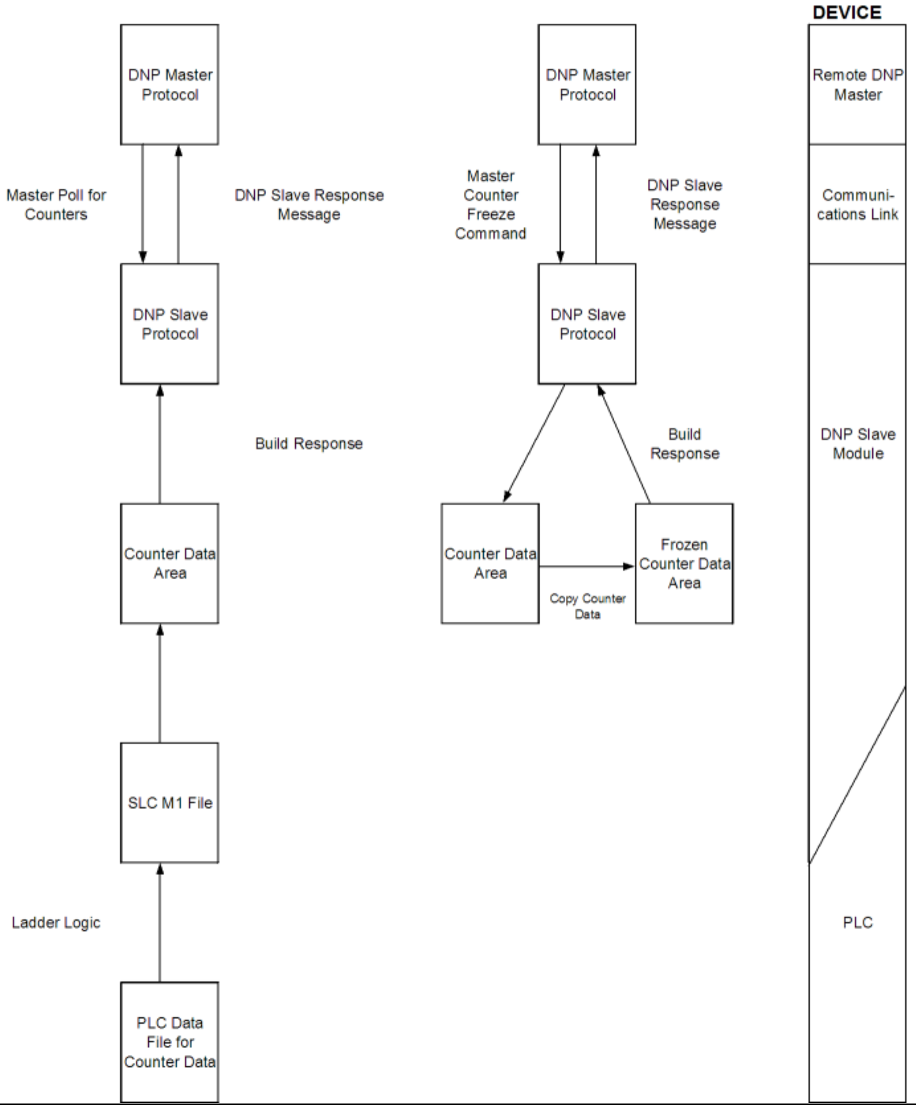
DNP Counter Data

This data type stores accumulated count data. These data are stored in the module in a double word value and have a data range of 0 to 4,294,967,296. The size of this data area is determined from the configuration parameter Counters. The SLC transfers data of this type to the module using the user data area in the SLC and the module's M1 file. The module maintains two values for each counter point: a current running value and a frozen value. The DNP master must send the freeze command to the module in order to transfer the current running values to the frozen area.

Note: The freeze-reset command is not supported in the data transfer operation. There is no way to guarantee counts will not be lost using the freeze-reset operation, therefore, this feature is not implemented.

A data flow diagram for the counter data is shown below.

Counter Data Flow Diagram



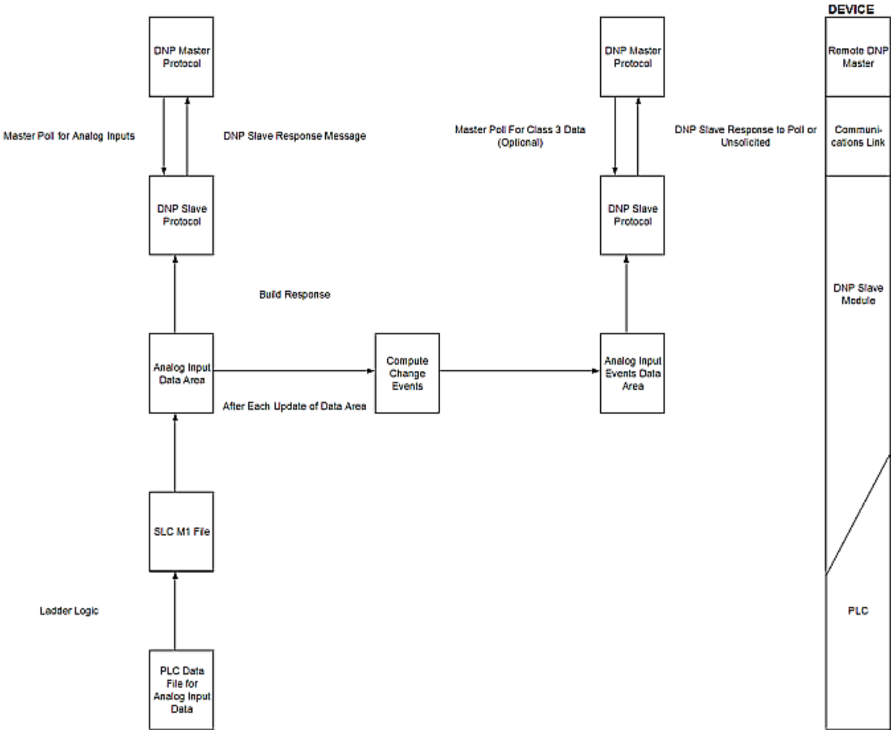
DNP Analog Input Data

This data type stores analog data with a data range of 0 to 65535 or -32768 to 32767. The size of this data area is determined from the configuration parameter Analog Inputs. These data are transferred to the module's M1-file from the SLC using the data in the user data files. Therefore, these data are read-only for the module and the DNP master unit. When the module receives a new block of this data from the SLC, it compares the new values to those currently in the database. If there is a change in any of the data, the module will generate an event message for the points that change. The dead-band parameter configured for the module determines the variance required for the event message.

The DNP master unit can read the current value data and the event data from the module. Event messages generated by the module can be retrieved using a poll for Class 3 data, as all analog input events are considered a Class 3 data type. If unsolicited message generation is enabled in the application, the events will automatically be sent by the module to the DNP master unit when the maximum event count for Class 3 data is reached or when the timeout for unsolicited messages is exceeded.

A data flow diagram for the analog input data is shown below.

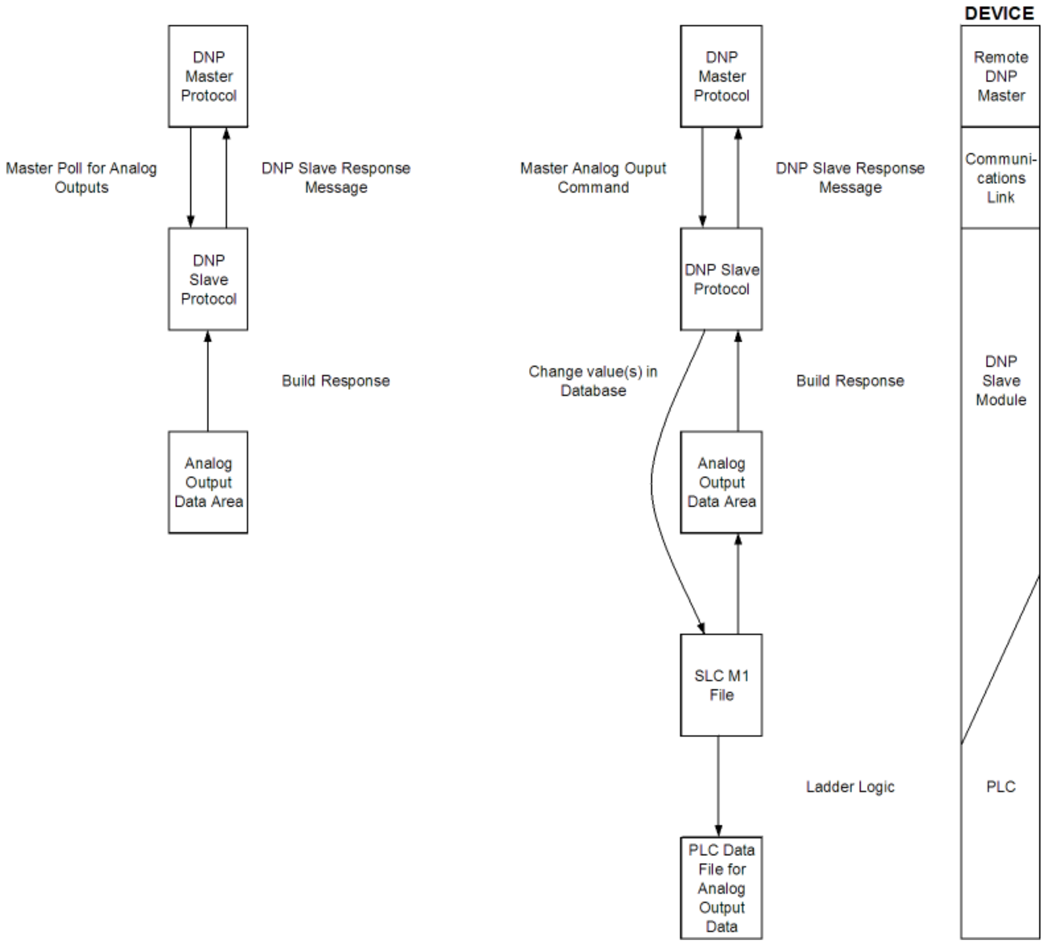
Analog Input Data Flow Diagram


DEVICE
DNP Master Protocol
DNP Master Protocol
Remote DNP Master
Master Poll for Analog Inputs
DNP Slave Response Message
Master Poll For Class 3 Data (Optional)
DNP Slave Response to Poll or Unsolicited
Communi-cations Link
DNP Slave Protocol
DNP Slave Protocol
Build Response
DNP Slave Module
Analog Input Data Area
Compute Change Events
Analog Input Events Data Area
After Each Update of Data Area
SLC M1 File
Ladder Logic
PLC
PLC Data File for Analog Input Data

DNP Analog Output Data

This data type stores analog values sent from the DNP master unit to the module and SLC with a data range of 0 to 65535 or -32768 to 32767. The size of this data area is determined from the configuration parameter Analog Outputs. These data are transferred from the module's M1 file to the SLC user data files. A data flow diagram for the analog output data is shown below.

Analog Output Data Flow Diagram



Communication Parameters

This phase of design defines the communication parameters required for successful communications between the module and DNP master and slave units over the Ethernet network. Determine the IP address for the module and the list of IP addresses that can connect to the unit if this feature is enabled. Refer to the MIS person in charge of assigning these addresses and setting up the network configuration. The Reference chapter contains a form to aid in setting these parameters. Fill out this form before attempting to configure the module. You must also determine if the UDP or the TCP protocol or both will be used in your application. The module supports a single connection for the TCP protocol. The UDP server supports receipt of messages from multiple clients. Access to both servers can be limited by using the IP address list filtering.

Functionality

This phase of design defines the features of the DNP Level 2 Subset supported by the module and to be utilized in the specific application. For example, will the unit use unsolicited messaging? Coordination with the DNP master developer is required to verify that the host will support the functionality you select. The features that must be defined in this design step are as follows:

- Will analog events be returned with or without a time value?
- Will events be logged before time synchronization has occurred?
- Will the module start with database values initialized by the processor?

For a complete description of the module configuration, refer to the **Module Setup** section.

Data Transfer at Startup

The module can be configured to have the internal databases initialized with data contained in the processor. This feature requires ladder logic. Data to be initialized are as follows: Binary and Analog Output data. This feature can be used to bring the module to a known state (last state set in controller) when the module is first initialized. For example, in order to have the module startup using the last set of binary output values and setpoint values (analog outputs), enable this feature.

If this feature is implemented, the module will request the data from the processor. Ladder logic must handle the Command Control Code value 1000. When the SLC receives the command, it must place any initialization data in the M1 file. After completing this operation, the Command Control Register (M1:1.8900) must be set to a value of 1001. The module will use the data in the M1 file to initialize the output data sets.

5.8.2 Module Operation

After the system has been designed and the system is set up, the module will be ready to operate. When the module is first initialized, it will read the configuration file (DNPSNET.CFG on the module's Compact Flash Disk). After the file is processed, the module will use the data to set up the data structures of the application. If any errors are encountered during the initialization process, the default value for the parameter will be assigned and used.

The module will next check if the output initialization feature is utilized. The option permits the SLC to set these read-only data at startup. There is no static memory available on the module to remember the last values for these data types. In order to prevent a "shock" to the system at boot time, this option can be used to set the module's database to the last transferred set of data. If this option is enabled, the module will request the binary and analog output from the SLC. Ladder logic must transfer the data for this feature to operate.

After the successful initialization of the module, the program will start the normal data transfer between the module and the SLC processor. The module will constantly place the M1 file image into the DNP data sets. When a command is received from a DNP master unit, the output image in the module's database and the M1 file will both be set to the value received.

If the module is configured for unsolicited messaging, the module will immediately send an unsolicited response once the remote master connects to the module, informing the master of a module restart. The module will not log events or process any data read operations from the master until the master clears the restart IIN data bit. The master must also synchronize the time with the module before events will be generated if the module is so configured. The master is also responsible for enabling the unsolicited message facility in the module by sending the Enable Unsolicited Messaging command to the module.

If the module is not configured for unsolicited messaging, the DNP master must clear the restart IIN bit before the module will start logging events. The master must also synchronize the time with the module before events will be generated if the module is so configured.

Additionally, the program will listen on Port 1 for requests. This is the debug port for the module and transfers module information to an attached terminal. Refer to the **Diagnostics and Troubleshooting** section for a complete discussion on the use of this important feature.

5.9 Event Size Computation

The minimum event buffer size required to avoid overflow can be computed as follows:

```
((number of static points)*(rate per second scan of change function)) /(rate
per second of master event data poll)
```

For example: 51 binary input points are scanned 2 times each second and polled by the master station about every 5 seconds. The minimum number of binary input events is:

```
(51 * 2)/.02 = 510 events
```

This computation assumes the unlikely event that all data points will change in consecutive calls to the scan of change function. If an event buffer overflow condition occurs, the internal indication bit, BUFFER OVERFLOW, will be set. If the system you are working with is fairly stable, the following equation can be used to compute the event buffer size:

```
(number of points that change per change function * rate per second of scan of
change function)*             (number of seconds between master event data
poll)
```

For example: 1000 binary input points are scanned 2 times each second and polled by the master station about every 5 seconds. Only about 5 points change state every scan of the change function call.

```
(5 * 2) * 5 = 50 events required
```

The number of events that can be defined in the system is limited to 400. The event buffer will overflow in systems which are very dynamic unless one of the following conditions exist:

The master frequently polls the slave device for events to keep the buffer empty.

OR

The slave is configured to send unsolicited messages to the master station. This method requires full-duplex operation of the network because the slave may be sending a message during a request from the master station.

In order to disable the report by exception feature in the module, set the number of events to 0 for both the binary and analog input events in the configuration. This will cause the DNP slave port driver to never return any data on object 2 and 32 and class 2 and 3 master station requests.

6 Support, Service & Warranty

In This Chapter

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 contents of file
   - Module Operation
   - Configuration/Debug status information
   - LED patterns
2. Information about the processor and user data files as viewed through and LED patterns on the processor.
3. Details about the serial devices interfaced, if any.

6.1 How to Contact Us: Technical Support

Internet	Web Site: http://www.prosoft-technology.com/support (http://www.prosoft-technology.com/support) E-mail address: support@prosoft-technology.com (mailto:support@prosoft-technology.com)

Asia Pacific

+603.7724.2080, support.asia@prosoft-technology.com (mailto:support.asia@prosoft-technology.com)

Languages spoken include: Chinese, English

Europe (location in Toulouse, France)

+33 (0) 5.34.36.87.20, support.EMEA@prosoft-technology.com (mailto:support.emea@prosoft-technology.com)

Languages spoken include: French, English

North America/Latin America (excluding Brasil) (location in California)

+1.661.716.5100, support@prosoft-technology.com (mailto:support@prosoft-technology.com)

Languages spoken include: English, Spanish

*For technical support calls within the United States, an after-*hours *answering system allows pager access to one of our qualified technical and/or application support engineers at any time to answer your questions.*

Brasil (location in Sao Paulo)

+55-11-5084-5178 , eduardo@prosoft-technology.com (mailto:eduardo@prosoft-technology.com)

Languages spoken include: Portuguese, English

6.2 Return Material Authorization (RMA) Policies and Conditions

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

6.2.1 All Product Returns:

a) In order to return a Product for repair, exchange or otherwise, the Customer must obtain a Returned Material Authorization (RMA) number from ProSoft and comply with ProSoft 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 89). 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, 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 using a shipment method other than that specified by ProSoft 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.

6.2.2 Procedures for Return of Units Under Warranty:

A Technical Support Engineer must approve the return of Product under ProSoft'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 at designated location referenced on the Return Material Authorization.

6.2.3 Procedures for Return of Units Out of Warranty:

a) Customer sends unit in for evaluation
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:

- 3150 - All
- 3750
- 3600 - All
- 3700
- 3170 - All
- 3250
- 1560 - Can be repaired, only if defect is the power supply
- 1550 - Can be repaired, only if defect is the power supply
- 3350
- 3300
- 1500 - All

6.2.4 Purchasing Warranty Extension:

a) ProSoft's standard warranty period is three (3) years from the date of shipment as detailed in "Limited Warranty (page 92)". The Warranty Period may be extended at the time of equipment purchase for an additional charge, as follows:

- Additional 1 year = 10% of list price
- Additional 2 years = 20% of list price
- Additional 3 years = 30% of list price

6.3 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, 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.

6.3.1 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 years from the date of shipment (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. 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 used replacement parts. 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 warranteed 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.

6.3.2 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.

6.3.3 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 of 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.

6.3.4 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.

6.3.5 Disclaimer of all Other Warranties

The Warranty set forth in What Is Covered By This Warranty (page 92) 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.

6.3.6 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 included, 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.

6.3.7 Time Limit for Bringing Suit

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

6.3.8 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.

6.3.9 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.

6.3.10 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