ProSoft TECHNOLOGY

Where Automation Connects.

inRAx®

MVI69-DNP

CompactLogix or MicroLogix Platform

DNP 3.0 Master/Slave Communication Module

11/3/2008

USER MANUAL

Please Read This Notice

Successful application of this module requires a reasonable working knowledge of the Rockwell Automation CompactLogix or MicroLogix hardware, the MVI69-DNP 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.

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

MVI69-DNP User Manual
11/3/2008

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

Please Read This Notice 2

Battery Life Advisory 2
Your Feedback Please 2
ProSoft® Product Documentation 3

Guide to the MVI69-DNP User Manual 7

1 Start Here 9

1.1 System Requirements 9
1.2 Package Contents 10
1.3 Setting Jumpers 11
1.4 Install the Module in the Rack 12
1.5 Connect your PC to the Processor 15
1.6 Download the Sample Program to the Processor 16
1.7 Connect your PC to the Module 19

2 Configuring the MVI69-DNP Module 21

2.1 Configuration File 21
2.2 MVI69-DNP Communication Module Configuration 23
2.3 Slave List 29
2.4 Command List 29
2.5 Uploading and Downloading the Configuration File 34

3 Ladder Logic 41

3.1 Module Data Objects 41
3.2 Special Objects 45
3.3 Adding the Module to an Existing CompactLogix Project 47
3.4 Adding the Module to an Existing MicroLogix Project 50

4 Diagnostics and Troubleshooting 53

4.1 Reading Status Data from the Module 53
4.2 LED Status Indicators 66

5 Reference 69

5.1 Product Specifications 69
5.2 Functional Overview 72
5.3 Cable Connections 89
5.4 Error Status Table 95
5.5 Internal Indication Word 103
5.6 DNP V3.00 Device Profile Document 104

5.7 DNP Subset Definition - Slave 105
5.8 DNP Subset Definition - Master 112

6 Support, Service & Warranty 119

6.1 How to Contact Us: Technical Support 119
6.2 Return Material Authorization (RMA) Policies and Conditions 120
6.3 LIMITED WARRANTY 122

Index 127

Guide to the MVI69-DNP 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 66) Diagnostics and Troubleshooting (page 53)	This section describes how to verify communications with the network. Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview Glossary	→	Reference (page 69) Functional Overview (page 72) Product Specifications (page 69)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 119)	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

- System Requirements 9
- Package Contents 10
- Setting Jumpers 11
- Install the Module in the Rack 12
- Connect your PC to the Processor 15
- Download the Sample Program to the Processor 16
- Connect your PC to the Module 19

Installing the MVI69-DNP module requires a reasonable working knowledge of the Rockwell Automation hardware, the MVI69-DNP 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 MVI69-DNP module requires the following minimum hardware and software components:

- Rockwell Automation CompactLogix or MicroLogix processor, with compatible power supply and one free slot in the rack, for the MVI69-DNP module. The module requires 800mA of available power.

Important: The MVI69-DNP module has a power supply distance rating of 2 (L43 and L45 installations on first 2 slots of 1769 bus).
Important: For 1769-L23x processors, please make note of the following limitations.

- 1769-L23-QBFC1B = 800mA at 5Vdc (1 MVI69-DNP will use all 800mA of available power. No other modules can be used with an MVI69 module connected to this processor).
- 1769-L23E-QB1B = 1000mA at 5Vdc (1 MVI69-DNP will use 800mA of available power. One other module can be used on this rack provided it consumes less than 200mA at 5Vdc.
- 1769-L23E-QBFC1B = 450mA at 5Vdc (no MVI69 module can be used with this processor)

- Rockwell Automation RSLogix 5000 (CompactLogix) or RSLogix 500 (MicroLogix) programming software
- Rockwell Automation RSLinx communication software

- Pentium® II 450 MHz minimum. Pentium III 733 MHz (or better) recommended
- Supported operating systems:
  - Microsoft Windows XP Professional with Service Pack 1 or 2
  - Microsoft Windows 2000 Professional with Service Pack 1, 2, or 3
  - Microsoft Windows Server 2003
- 128 Mbytes of RAM minimum, 256 Mbytes of RAM recommended
- 100 Mbytes of free hard disk space (or more based on application requirements)
- 256-color VGA graphics adapter, 800 x 600 minimum resolution (True Color 1024 × 768 recommended)
- CD-ROM drive
- HyperTerminal or other terminal emulator program capable of file transfers using Ymodem protocol.

1.2 Package Contents

The following components are included with your MVI69-DNP module, and are all required for installation and configuration.

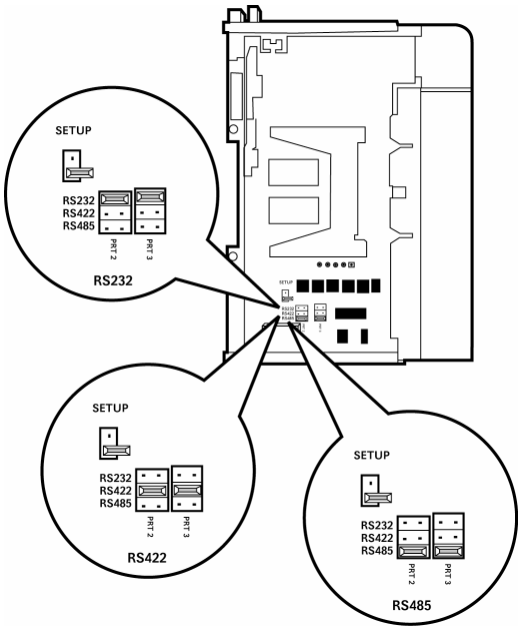
Important: Before beginning the installation, please verify that all of the following items are present.

Qty.	Part Name	Part Number	Part Description
1	MVI69-DNP Module	MVI69-DNP	DNP 3.0 Master/Slave Communication Module
1	Cable	Cable #15, RS232 Null Modem	For RS232 Connection to the CFG Port
3	Cable	Cable #14, RJ45 to DB9 Male Adapter cable	For DB9 Connection to Module's Port
2	Adapter	1454-9F	Two Adapters, DB9 Female to Screw Terminal. For RS422 or RS485 Connections to Port 1 and 2 of the Module
1	ProSoft Solutions CD		Contains sample programs, utilities and documentation for the MVI69-DNP module.

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

1.3 Setting Jumpers

When the module is manufactured, the port selection jumpers are set to RS-232. To use RS-422 or RS-485, you must set the jumpers to the correct position. The following diagram describes the jumper settings.



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.

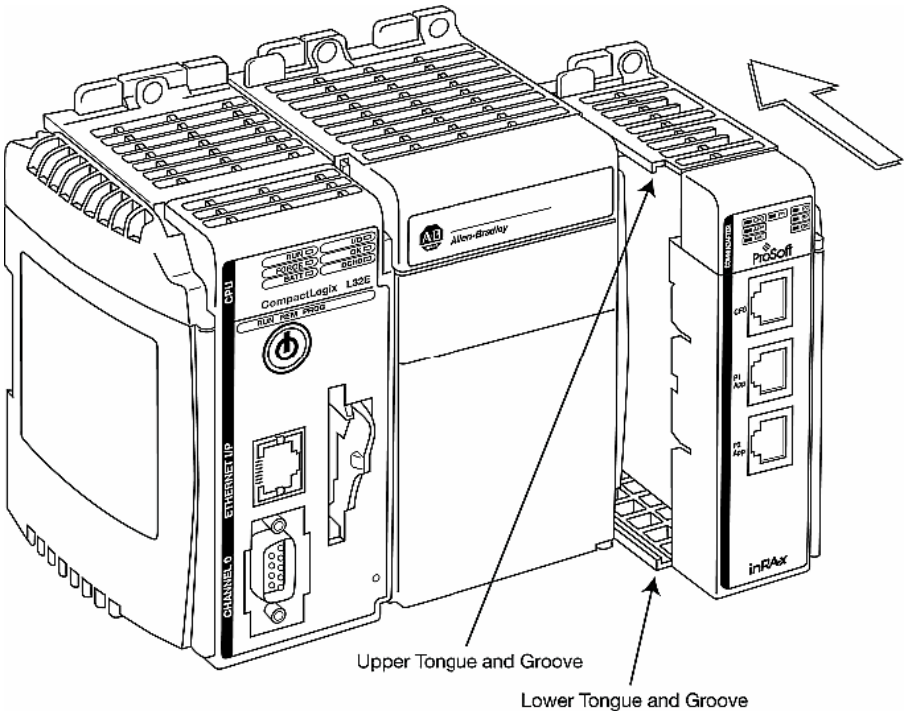
1.4 Install the Module in the Rack

This section describes how to install the module into a CompactLogix or MicroLogix rack

Before you attempt to install the module, make sure that the bus lever of the adjacent module is in the unlocked (fully right) position.

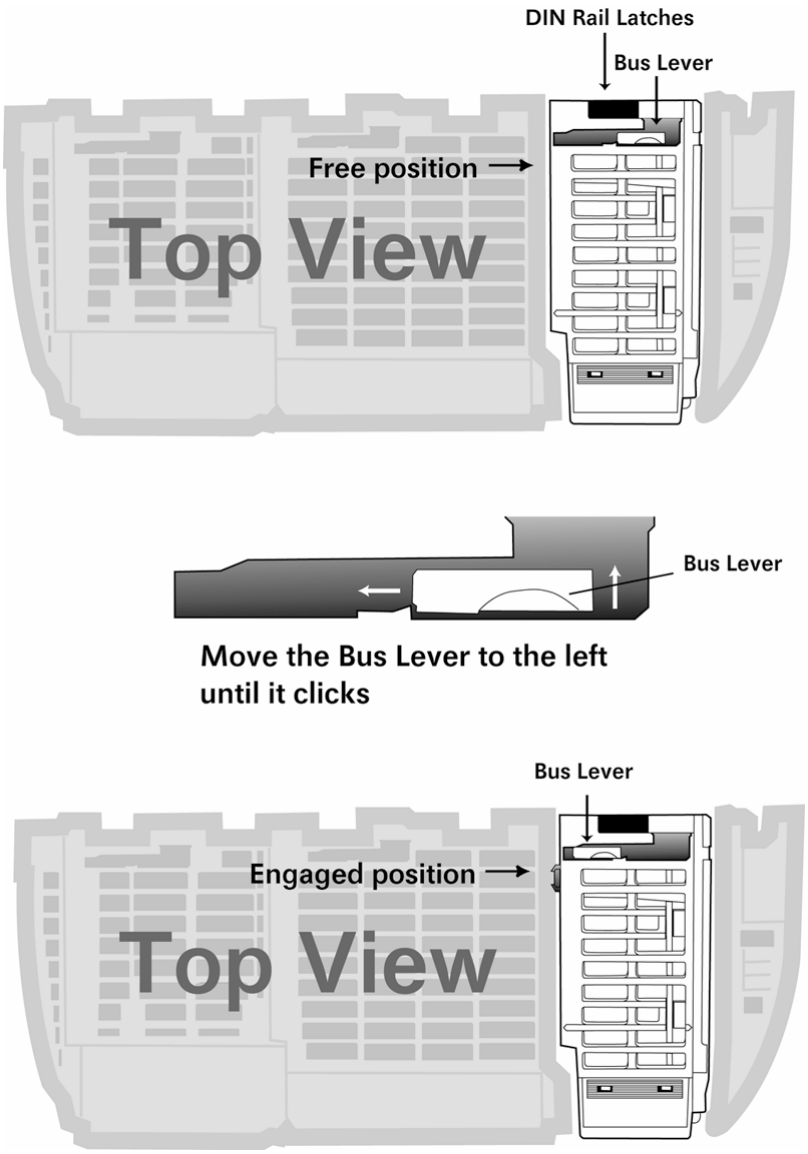
Warning: This module is not hot-swappable! Always remove power from the rack before inserting or removing this module, or damage may result to the module, the processor, or other connected devices.

1 Align the module using the upper and lower tongue-and-groove slots with the adjacent module and slide forward in the direction of the arrow.



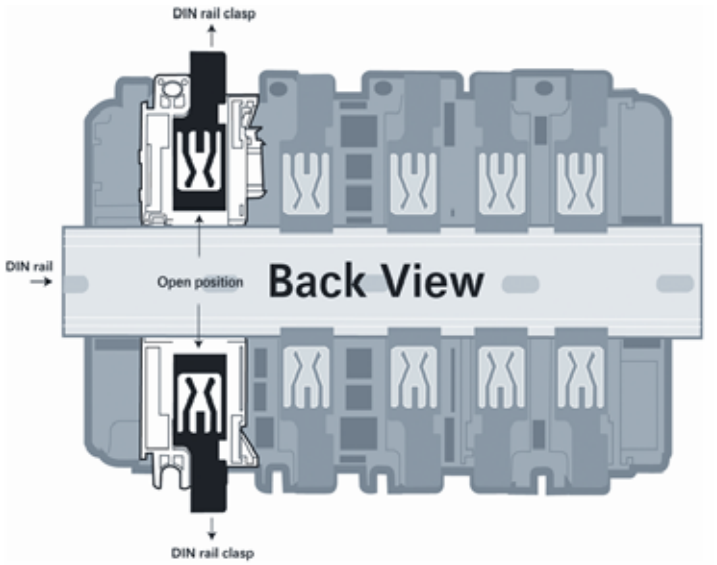
2 Move the module back along the tongue-and-groove slots until the bus connectors on the MVI69 module and the adjacent module line up with each other.

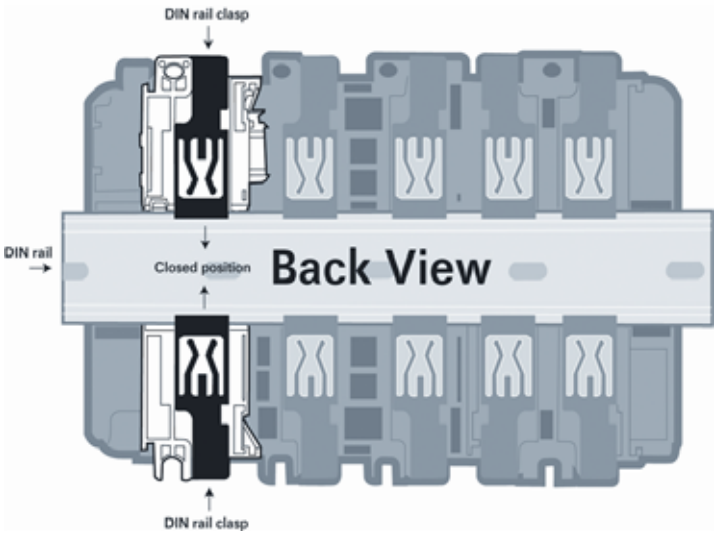
3 Push the module's bus lever back slightly to clear the positioning tab and move it firmly to the left until it clicks. Ensure that it is locked firmly in place.



4 Close all DIN rail latches.

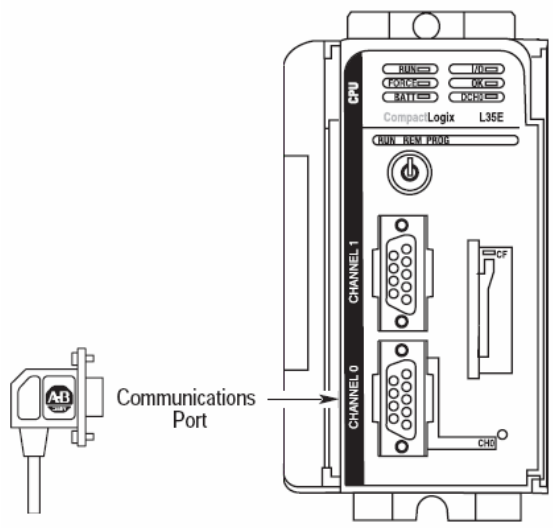
5 Press the DIN rail mounting area of the controller against the DIN rail. The latches will momentarily open and lock into place.



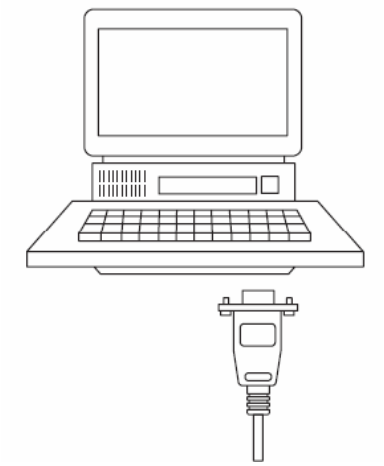


1.5 Connect your PC to the Processor

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



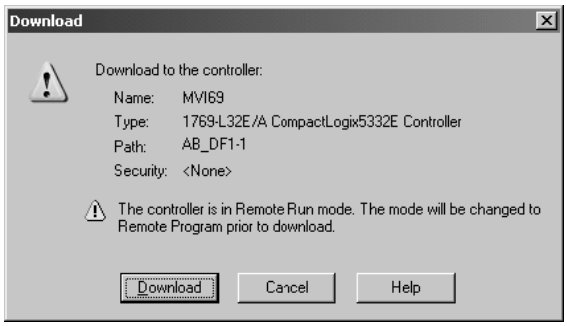
2 Connect the straight connector end of the cable to the serial port on your computer.



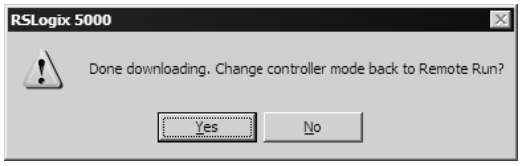
1.6 Download the Sample Program to the Processor

Important: For most applications, the sample program will work without modification.
Note: The key switch on the front of the CompactLogix 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.
2. When communication is established, RSLogix will open a confirmation dialog box. 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.
4. When the download is complete, RSLogix will open another confirmation dialog box. Click OK to switch the processor from Program mode to Run mode.

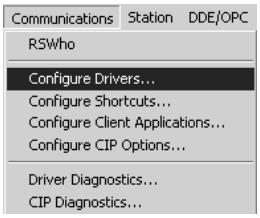


Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

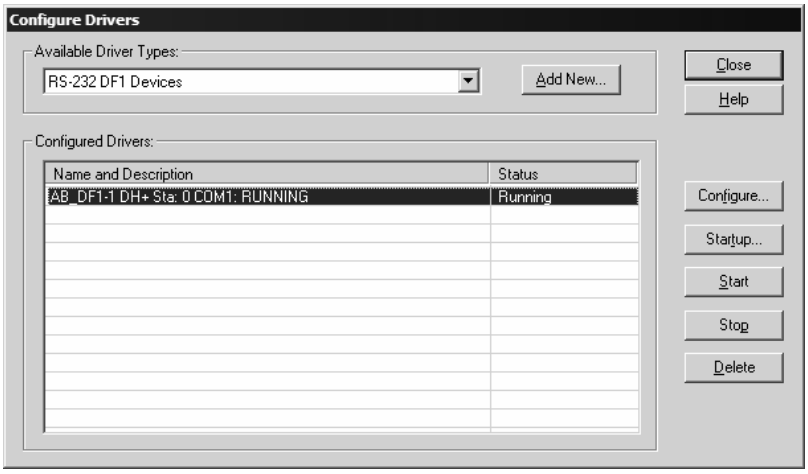
1.6.1 Configuring RSLinx

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.



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.

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

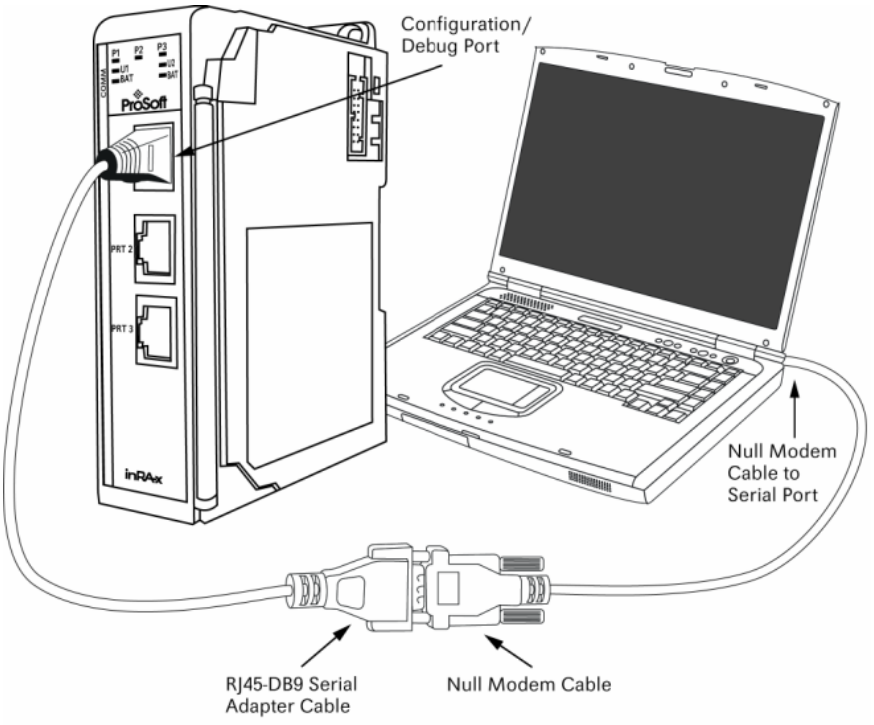
5 When you see the message "Auto Configuration Successful", click the OK button to dismiss the dialog box.

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.

1.7 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 Configuring the MVI69-DNP Module

In This Chapter

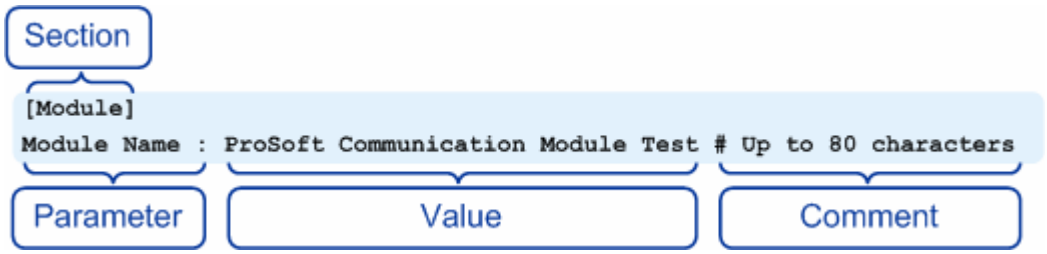
- Configuration File 21
- MVI69-DNP Communication Module Configuration 23
- Slave List 29
- Command List 29
- Uploading and Downloading the Configuration File 34

2.1 Configuration File

The MVI69-DNP module stores its configuration in a text file called DNP.CFG, located in the module's flash memory. When the module starts up, it reads the configuration file and uses the information to control how the DNP 3.0 Master/Slave protocol interacts with the module's application port(s).

The configuration file is arranged in *Sections,* with a heading in **[ ]** characters at the beginning of each section. Each *Section* contains a list of *Parameters* and *Values,* followed by an optional *Comment* that explains the parameter.

The following illustration shows an example of a *Section,* a *Parameter,* a *Value,* and a *Comment.*



The *Parameter* must be followed by a **[:]** (colon) character. The text following the **[:]** is a *Value.*

The module ignores *"comment"* text following the **[#]** character. Use comments to document your configuration settings.

You can get a sample configuration file for the module in the following places:

- Copy (page 34) the DNP.CFG from the module's flash memory to your PC
- Copy the DNP.CFG from the ProSoft Solutions CD-ROM supplied with the module
- Download the DNP.CFG from the ProSoft Technology web site at http://www.prosoft-technology.com

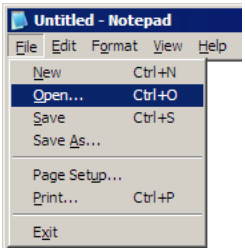
2.1.1 Editing the Configuration File

The DNP.CFG file is a plain ASCII text file. Use a text editor such as Notepad.exe (included with Microsoft Windows) to open and edit the file.

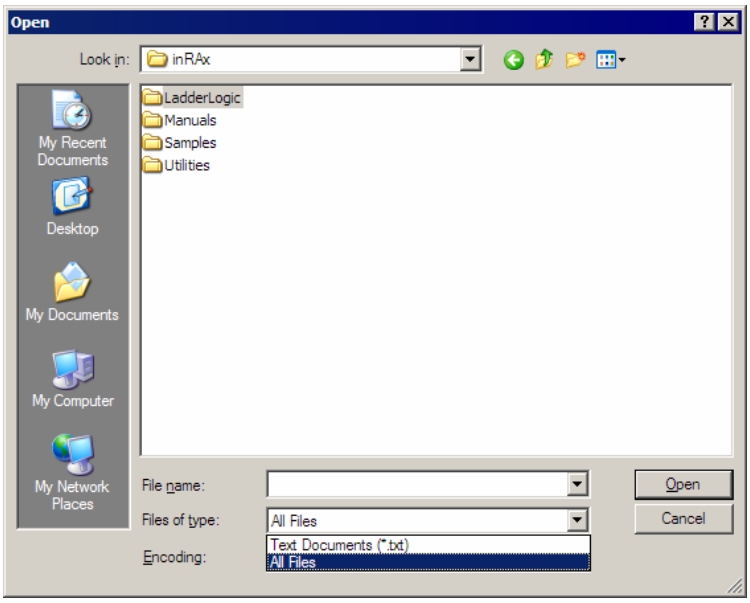
To open the configuration file in Notepad

1. Click the Start button, and then choose Programs
2. Expand the Programs menu, and then choose Accessories.
3. On the Accessories menu, choose **Notepad.**

4. In Notepad, open the File menu, and then choose Open



5. In the Open dialog box, select "All Files" in the Files of Type: dropdown list.



Tip: Sample configuration files are stored under the LadderLogic folder on the ProSoft Solutions CD-ROM.

6. Navigate to the folder containing the configuration file, and then select the file to edit.
7. Click Open to open the file.
8. When you have finished editing, save the file and close Notepad.

Important: Changes to the configuration file will not take effect until you download the file to the module, and then reboot the module.

2.2 MVI69-DNP Communication Module Configuration

[Section]/Item	Value	Range	Description
[MODULE]			General module configuration section
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.

[Section]/Item	Value	Range	Description
[DNP Slave]			DNP Slave configuration information
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.
Baud Rate:		Baud rate value	Primary DNP Port Baud Rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 384 (38400) , 576 (57600), 115 (115200)
RTS On:		0 to 65535 milliseconds	This value represents the number of 1 ms increments to be inserted between asserting the RTS modem line and the actual transmission of the data.
RTS Off:		0 to 65535 milliseconds	This value represents the number of 1 ms increments to be inserted after the last character of data is transmitted before the RTS modem line is dropped.
Min Response Delay:		0 to 65535 milliseconds	Minimum time between receiving a request and transmitting a response. Allows master time to disable transmitter on an RS-485 network.
Modem:		Yes or No	This parameter defines if a dial-up modem is used on the secondary DNP slave port. A modem cannot be used if the port is configured as a master.
Connect Timeout:		0 to 65535	Defines the number of milliseconds to wait for the CD signal to be set high. The CD signal indicates a connection is made using a dial-up modem.
First Character Delay:		0 to 65535	Defines the number of milliseconds to wait before sending the first message after the connection is first made. This delay only applies to the first packet sent to the modem.
Redial Delay Time:		0 to 32000	Defines the minimum number of milliseconds to wait before a redial attempt is made by the slave.
Redial Random Delay:		0 to 32000	Defines a random millisecond time range to be added to the redial delay time before the modem is accessed.
Idle Timeout:		0 to 65535	Defines the number of milliseconds the modem is inactive before it will disconnect.
Phone Number:		ASCII String Data	This field contain a null-terminated, ASCII character string used by the dial-up modem. The string must contain all characters required by the modem. An example string is ATDT1800222333. Maximum length is 34 bytes including the terminating 0.
Collision Avoidance:		Yes or No	This parameter defines if collision avoidance will be utilized on the primary DNP slave port.
CD Idle Time:		0 to 32000	Defines the minimum number of milliseconds to wait before transmitting a message after the CD signal is recognized as low.
CD Random Time:		0 to 32000	Defines the range of random time to be added to the CD Idle Time before a message will be transmitted from the slave.

[Section]/Item	Value	Range	Description
CD Time Before Receive:		0 to 65535	Defines the number of milliseconds to wait before receiving characters after the CD signal is recognized as high.
BI Class:		0 to 3	This parameter specifies the default class to be utilized for all the binary input points in the DNP database that are not defined in the override list section.
AI Class:		0 to 3	This parameter specifies the default class to be utilized for all the analog input points in the DNP database that are not defined in the override list section.
Float Class:		0 to 3	This parameter specifies the default class to be utilized for all the floating-point input points in the DNP database that are not defined in the override list section.
AI Deadband:		0 to 32767	This parameter specifies the default deadband value assigned to all points not defined in the override list for the analog input point type in the DNP database.
Float Deadband:		0 to maximum float value	This parameter specifies the default deadband value assigned to all points not defined in the override list for the floating-point input point type in the DNP database.
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.
Data Link Confirm Mode:		Coded Value (N=Never, S=Sometimes, A=Always)	IED can request acknowledgement from master station when sending data. The codes are as follows: 0=Never, 1=Sometimes, 2=Always
Data Link Confirm Tout:		1 to 65535 milliseconds	Time period to wait for Master Data Link confirmation of last frame sent. This time is in milliseconds. This parameter is only used if the frame is sent with confirmation requested.
Data Link Max Retry:		0 to 255 retries	Maximum number of retries at the Data Link level to obtain a confirmation. If this value is set to 0, retries are disabled at the data link level of the protocol. This parameter is only used if the frame is sent with confirmation requested.
App Layer Confirm Tout:		1 to 65535 milliseconds	Event data contained in the last response may be sent again if not confirmed within the millisecond time period set. If application layer confirms are used with data link confirms, ensure that the application layer confirm timeout is set long enough.
Unsolicited Response:		Yes or No	Set if the slave unit will send unsolicited response messages. If set to No, the slave will not send unsolicited responses. If set to Yes, the slave will send unsolicited responses.
Class 1 Unsol Resp Min:		1 to 255 events	Minimum number of events in Class 1 required before an unsolicited response will be generated.
Class 2 Unsol Resp Min:		1 to 255 events	Minimum number of events in Class 2 required before an unsolicited response will be generated.
Class 3 Unsol Resp Min:		1 to 255 events	Minimum number of events in Class 3 required before an unsolicited response will be generated.

[Section]/Item	Value	Range	Description
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.
Uresp Retry Count:		0 to 255 retries	Determines the number of unsolicited message retries sent on primary DNP port before changing to secondary port. If the value is 0, port switching will be disabled.
AI Events with time:		Yes or No	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 No, the default is set to no time data. If the parameter is set to Yes, the default object will include the time of the event.
Time Sync Before Events:		Yes or No	This parameter determines if events are to be generated by the module before the time synchronization from the master unit. If the parameter is set to No, no events will be generated until the module's time has been synchronized. If the parameter is set to Yes, events will always be generated.
Initialize DNP Database:		Y or N	This parameter determines if the module will request data from the processor to initialize the DNP database output data areas. If this option is utilized, ladder logic is required to send the requested block from the processor to the module.
Use Trip/Close Single Point		Y or N	Used for backwards compatibility with older MVI69-DNP modules, will cause Trip/Close operations to use a single point operation.

[Section]/Item	Value	Range	Description
[DNP Slave Database]			DNP Slave Database definition
Binary Inputs:		0 to 7680 points	Number of digital input points to configure in the DNP slave device. Each point will be stored as a single bit in the module memory.
PLC Binary Inputs:		0 to 7680 points	Number of digital input points configured above that are to be obtained from the CompactLogix processor. All other binary input points must come from the attached IED units.
Analog Inputs:		0 to 480 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.
PLC Analog Inputs:		0 to 480 points	Number of analog input points configured above that are to be obtained from the CompactLogix processor. All other analog input points must come from the attached IED units.
Float Inputs:		0 to 240 points	Number of floating-point input points to configure in the DNP slave device. Each point will occupy a two-word area in the module memory.
PLC Float Inputs:		0 to 240 points	Number of floating-point input points configured above that are to be obtained from the PLC.
Counters:		0 to 240 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.

[Section]/Item	Value	Range	Description
PLC Counters:		0 to 240 points	Number of counter points configured above that are to be obtained from the CompactLogix processor. All other counter points must come from the attached IED units.
Binary Outputs:		0 to 7680 points	Number of digital output points to configure in the DNP slave device. Each point will be stored as a single bit in the module memory.
PLC Binary Outputs:		0 to 7680 points	Number of digital output points configured above that are to be sent to the CompactLogix processor. All other binary output points will be sent to the attached IED units.
Analog Outputs:		0 to 480 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.
PLC Analog Outputs:		0 to 480 points	Number of analog output points configured above that are to be sent to the CompactLogix processor. All other analog output points will be sent to the attached IED units.
Float Outputs:		0 to 240 points	Number of floating-point output points to configure in the DNP slave device. Each point will occupy a two-word area in the module memory.
PLC Float Outputs:		0 to 240 points	Number of floating-point output points configured above that are to be sent to the CompactLogix.

[Section]/Item	[redacted]	Description
[DNP Slave Binary Inputs]	[redacted]	DNP database binary input override values
# This area is to override the class assignment for binary input database points. Enter list of points between the		
# START and END labels.		
#		
# Point# Class		
START		
END		

[Section]/Item	[redacted]	Description
[DNP Slave Analog Inputs]	[redacted]	DNP database analog input override values
START		
# This area is to override the class and deadband assignment for analog input database points.		
# Enter list of points between the START and END labels.		
#		
# Point# Class Deadband		
START		
END		

[Section]/Item	[redacted]	Description
[DNP Slave Float Inputs]	[redacted]	DNP database floating-point input override values
# This area is to override the class and deadband assignment for float input database points.		
# Enter list of points between the START and END labels.		
#		
# Point# Class Deadband		

[Section]/Item			Description
START			
END			

[Section]/Item	Value	Range	Description
[Secondary Port]			Definitions for secondary port on module
Type:		M or S or blank	This parameter defines the functionality of the secondary port on the module. M = emulate a DNP master port S = back-up DNP slave port to the primary port. Any other value will disable the port.
Baud Rate:		Baud rate value	Secondary DNP Port Baud Rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 384 (38400) , 576 (57600), 115 (115200)
RTS On:		0 to 65535 milliseconds	This value represents the number of 1 ms increments to be inserted between asserting the RTS modem line and the actual transmission of the data.
RTS Off:		0 to 65535 milliseconds	This value represents the number of 1 ms increments to be inserted after the last character of data is transmitted before the RTS modem line is dropped.
Min Response Delay:		0 to 65535 milliseconds	Minimum time between receiving a request and transmitting a response. Allows master time to disable transmitter on an RS-485 network.
Collision Avoidance:		Yes or No	This parameter defines if collision avoidance will be utilized on the primary DNP slave port.
CD Idle Time:		0 to 32000	Defines the minimum number of milliseconds to wait before transmitting a message after the CD signal is recognized as low.
CD Random Time:		0 to 32000	Defines the range of random time to be added to the CD Idle Time before a message will be transmitted from the slave.
CD Time Before Receive:		0 to 65535	Defines the number of milliseconds to wait before receiving characters after the CD signal is recognized as high.

[Section]/Item	Value	Range	Description
[DNP Master]			Definitions for DNP Master port if utilized.
Internal 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.
Initialize IED Database:		Yes or No	This parameter determines if the module will request data from the processor to initialize the IED database input data areas. If this option is utilized, ladder logic is required to send the requested block from the processor to the module.
Event Messages to PLC:		Yes or No	This parameter determines if event messages received on the master port will be sent to the processor. If this option is utilized, ladder logic must handle the 9903 blocks generated by the module.

[Section]/Item	Value	Range	Description
[IED Database]			Database definition for DNP master port if utilized
Binary Inputs:		0 to 7680 points	Number of binary input points contained in the IED database to transfer to the CompactLogix processor and obtained from the attached IED units..
Analog Inputs:		0 to 480 points	Number of analog input points contained in the IED database to transfer to the CompactLogix processor and obtained from the attached IED units..
Counters:		0 to 240 points	Number of counter points contained in the IED database to transfer to the CompactLogix processor and obtained from the attached IED units..
Binary Outputs:		0 to 7680 points	Number of binary output points contained in the IED database which are transferred from the CompactLogix processor and used by the attached IED units..
Analog Outputs:		0 to 480 points	Number of analog output points contained in the IED database which are transferred from the CompactLogix processor and used by the attached IED units..

[Section]/Item		Description
[DNP Master Slave List]		Definition of the IED units to be interfaced with the DNP master port if utilized

This section stores information about each slave to be used by the master port.
There must be an entry in this table for each node to be used in the command list.
Two of the parameters in this list are coded values:
Conf Mode ==> 0=Never, 1=Sometimes and 2=Always (select 0).
Flags is bit coded as follows:
Bit 0 (decimal 1) ==> Enable the slave
Bit 1 (decimal 2) ==> Use Unsolicited messaging with this slave
Bit 2 (decimal 4) ==> Use delay measurement with this slave
Bit 3 (decimal 8) ==> Auto time synchronization enabled
#
Node DL Conf Conf Conf App Rsp
Address Mode Timeout Retry Timeout Flags

START

END

[Section]/Item		Description
[DNP Master Commands]		Definition of the commands to be issued to the IED units by the DNP master port.

This section contains the list of commands to process on the master port.
Node addresses present in the command list must have an entry in the
[DNP Slave List]. Commands with nodes not present in the list will not be
executed.
1 2 3 4 5 6 7 8 9 10
#Flags/ Node Data Data Cmd Device Point DNP DB IED DB Poll
#Enable Address Object Variation Func Address Count Address Address Interval

START

END

2.3 Slave List

The slave list defines the IED units and their specific communication parameters for a DNP master port. Up to 40 IED units can be defined in the module to be associated with the master port. The structure of each row in the list is defined in the following table.

Column	Variable Name	Data Range	Description	If Error
1	DNP Slave Address	0 to 65534	This is the slave address for the unit to override the default values.	Ignore
2	Data Link Confirm Mode	Coded Value (0 = Never, 1 = Sometimes, 2 = Always)	This value specifies if data link frames sent to the remote device require a data link confirm. This value should always be set to zero for almost all applications.	0
3	Data Link Confirm Tout	1 to 65535 milliseconds	This parameter specifies the time to wait for a data link confirm from the remote device before a retry is attempted.	300
4	Maximum Retries for Data Link Confirm	0 to 255 retries	Maximum number of retries at the Data Link level to obtain a confirmation. If this value is set to 0, retries are disabled at the data link level of the protocol. This parameter is only used if the frame is sent with confirmation requested.	3
5	Application Layer Response Timeout	1 to 65535 milliseconds	Timeout period the master will wait for each response message fragment. If data link confirms are enabled, make sure the timeout period is set long enough to permit all data confirm retries.	5000
6	Slave Mode	Coded Value (Bit 0 = Enable, Bit 1 = Unsol Msg., Bit 2 = Use DM, Bit 3 = Auto Time Sync)	This word contains bits that define the slave mode. The slave mode defines the functionality of the slave device and can be combined in any combination. The fields have the following definition: Enable: determines if this slave will be used. Unsol Msg.: causes an enabled unsolicited message to be sent to the slave when its RESTART IIN bit is set. This parameter is also required for unsolicited message reporting by the IED unit. Use DM: uses delay measurement. Auto Time Sync: time synchronization used when NEED TIME IIN bit set.	

2.4 Command List

The command list stores the command list used by the DNP master port. This list only must be defined if the DNP master port is used. Up to 300 commands can be defined for the master port. The structure of each row in the list is shown in the following table.

Word Offset	Definitions
0	Port/Flags
1	Slave Address
2	Object
3	Variation
4	Function

Word Offset	Definitions
5	Address in Slave
6	Point Count
7	DNP DB Address
8	IED DB Address
9	Poll Interval

The definition of each parameter required for each command is provided in the following table.

Bits in the Port/Flags parameter are dependent on the data type. The following table defines the Port/Flags bits for binary input, analog input and counter data points.

Port/Flags Bits	Description	Decimal Equivalent
0 to 1	Communication port (0=Internal, 2=Port 2)	0 or 2
2	Enable/Disable Command (1=Enable, 0=Disable)	4
3	RBE Flag (0=Events from IED, 1=Events by module)	8
4 to 7	Not Used	

For these data types the qualifier used in the data request is dependent on the Point Count and Address in Slave fields in the command as follows:

```
If Point Count < 0, then use Qualifier 06h (All points, packed & -Point Count =
# of points to consider)
If Address in Slave = 0 & Point Count > 0, then use Qualifier 00h or 01h
(points 0 to Point Count -1)
If Address in Slave > 0 & Point Count > 0, then use Qualifier 00h or 01h
(Address in Slave to Address in Slave + Point Count - 1)
```

The following table defines the Port/Flags bits for binary output and analog output points.

Port/Flags Bits	Description	Decimal Equivalent
0 to 1	Communication port (0=Internal, 2=Port 2)	0 or 2
2	Enable/Disable Command (1=Enable, 0=Disable)	4
3	Poll Type (0=Poll, 1=Exception)	8
4	Data Source (0=DNP Database, 1=IED Database)	16
5 to 7	Not Used	

For these data types the qualifier used in the data request is dependent on the Point Count and Address in Slave fields in the command as follows:

```
If Address in Slave = 0 & Point Count > 0, then use Qualifier 17h or 28h (Point
Count specified starting at point 0)
If Address in Slave > 0 & Point Count > 0, then use Qualifier 17h or28h (points
from Address in Slave to Address in Slave + Point Count - 1)
If Point Count <= 0, then ignore because this is illegal for outputs.
```

	Digital input	Digital input Events	Digital Output	Counter
0	Port/Flags	Port/Flags	Port/Flags	Port/Flags
1	Slave Address	Slave Address	Slave Address	Slave Address
2	1	2	12	20
3	0, 1 or 2	0, 1, 2 or 3	1*	0, 5 or 6
4	1*	1*	3, [4], 5 or 6	1, 7, 8, 9 or 10
5	Address in Slave	Address in Slave	Address in Slave	Address in Slave

	Digital input	Digital input Events	Digital Output	Counter
6	# of Points	# of Points	# of Points	# of Points
7	DNP DB Address		DNP DB Address	DNP DB Address
8	IED DB Address		IED DB Address	IED DB Address
9	Poll Interval	Poll Interval	Poll Interval	Poll Interval

	Frozen Counter	Analog Input	Analog Input Events	Analog Output
0	Port/Flags	Port/Flags	Port/Flags	Port/Flags
1	Slave Address	Slave Address	Slave Address	Slave Address
2	21	30	32	41
3	0, 9 or 10	0, 1, 2, 3 or 4	0, 1, 2, 3 or 4	2*
4	1*	1*	1*	3, [4], 5 or 6
5	Address in Slave	Address in Slave	Address in Slave	Address in Slave
6	# of Points	# of Points	# of Points	# of Points
7	DNP DB Address	DNP DB Address		DNP DB Address
8	IED DB Address	IED DB Address		IED DB Address
9	Poll Interval	Poll Interval	Poll Interval	Poll Interval

	Time and Date	Class 0	Class 1	Class 2
0	Port/Flags	Port/Flags	Port/Flags	Port/Flags
1	Slave Address	Slave Address	Slave Address	Slave Address
2	50	60	60	60
3	1*	1	2	3
4	2*			
5				
6	1	1	1	1
7				
8				
9	Poll Interval	Poll Interval	Poll Interval	Poll Interval

	Class 3	Cls 1, 2 & 3	Cls 0, 1, 2 & 3	[Clear Restart Bit]
0	Port/Flags	Port/Flags	Port/Flags	Port/Flags
1	Slave Address	Slave Address	Slave Address	Slave Address
2	60	60	60	80
3	4	5	6	1
4				2
5				7
6	1	1	1	1
7				
8				
9	Poll Interval	Poll Interval	Poll Interval	

	Cold Restart	Warm Restart	Enable Unsol. Msg	Disable Unsol. Msg
0	Port/Flags	Port/Flags	Port/Flags	Port/Flags
1	Slave Address	Slave Address	Slave Address	Slave Address
2	0	0	0	0
3				
4	13	14	20	21
5				
6	1	1	1	1
7				
8				
9				

*	Value assumed		
[ ]	Automatically implemented	Word Offset	Definitions
		0	Port/Flags
		1	Slave Address
		2	Object
		3	Variation
		4	Function
		5	Address in Slave
		6	Point Count
		7	DNP DB Address
		8	IED DB Address
		9	Poll Interval

Commands are issued based on the following criteria:

Commands Issued Each Scan	Enabled Poll Interval = 0 BO & AO have Exception Bit = 0
Commands Issued at Poll Time	Enabled Poll Interval > 0 BO and AO have Exception Bit = 0
Commands Issued on Data Change (BO and AO)	Enabled BO and AO have Exception Bit = 1
Commands Issued by PLC	

2.4.1 Slave Address

This parameter specifies the IED unit address on the DNP network to consider with the command. The parameter has a range of 0 to 65535. The value of 65535 is reserved for broadcast messages. Verify that the slave configuration information is set up in the module for each slave defined in the command list.

2.4.2 Object

This parameter specifies the DNP object type in the command. Valid objects for the module are 1, 2, 12, 20, 21, 30, 32, 41, 50, 60 and 80. A value of 0 is permitted in this field for a set of special commands.

2.4.3 Variation

This parameter is specific to the object type selected.

2.4.4 Function

This parameter specifies the DNP function for the command list object. The object type determines the value of the functions permitted. For example, the only function permitted for binary input data points is the read function (Function Code 1). For counter and output objects, more functions are available.

2.4.5 Address In Slave

This value must be greater-than or equal to zero. If it is set to a value less-than zero, the command will be ignored. This parameter specifies the starting point address to consider in the IED unit.

2.4.6 Point Count

This parameter defines the number of points in the IED unit. Refer to the discussion above for the interpretation of this parameter's values for the different object types.

2.4.7 DNP DB Address

This parameter defines the starting location in the DNP database to be used with the command. If the parameter has a value of -1, the DNP database is not used with the point.

2.4.8 IED DB Address

This parameter defines the starting location in the IED database to be used with the command. If the parameter has a value of -1, the IED database is not used with the point.

2.4.9 Poll Interval

This parameter specifies the minimum frequency at which the module should execute the command. The value is entered in units of seconds. For example, to execute a command every 10 seconds, enter a value of 10 in the field. A value of 0 for the parameter implies that the command should be executed every scan of the list.

2.5 Uploading and Downloading the Configuration File

ProSoft modules are shipped with a pre-loaded configuration file. In order to edit this file, you must transfer the file from the module to your PC. After editing, you must transfer the file back to the module.

This section describes these procedures.

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. For specific information about the configuration/debug menus in your module, refer to The Configuration/Debug Menu (page 53).

2.5.1 Transferring the Configuration File to Your PC

1 Connect your PC to the Configuration/Debug port of the module using a terminal program such as HyperTerminal. Press **[?]** to display the main menu.

```
******* DNP DEBUG PORT HELP *******
 KEY     FUNCTION                  | KEY FUNCTION
 ------- --------------------------+---- --------------------------
 0-9,A-F Sets debug level          | Y   Class/Deadband Assignments
 L       Display error list        | U   Show DNP Databases
 P       Display setup & pointers  | <   Receive Configuration
 O       Operating parameters      | >   Send Configuration
 R       Reboot module             |
 S       Display Comm Stats        |
 W       Clear error list          | N   Display Blk X-fer Stats
 V       List COM States           | X   Master Port Commands
 T       Master Port Slave Setup   | Z   Master Port Slave Errs
 G       Version Information       | ?   Display this screen

PRODUCT = DNP5   REVISION = 2.35   OP SYS REV = 1206   PROD RUN # = 1501
```

2 Press **[>]** key (Send Module Configuration). The message "Press Y to confirm configuration send!" is displayed at the bottom of the screen.

```
******* DNP DEBUG PORT HELP *******
 KEY     FUNCTION                  | KEY FUNCTION
 ------- --------------------------+---- -------------------------
 0-9,A-F Sets debug level          | Y   Class/Deadband Assignments
 L       Display error list        | U   Show DNP Databases
 P       Display setup & pointers  | <   Receive Configuration
 O       Operating parameters      | >   Send Configuration
 R       Reboot module             |
 S       Display Comm Stats        |
 W       Clear error list          | N   Display Blk X-fer Stats
 V       List COM States           | X   Master Port Commands
 T       Master Port Slave Setup   | Z   Master Port Slave Errs
 G       Version Information       | ?   Display this screen

PRODUCT = DNP5   REVISION = 2.35   OP SYS REV = 1206   PROD RUN # = 1501

Confirm Receive Configuration File from Remote PC by pressing 'Y' key....
```

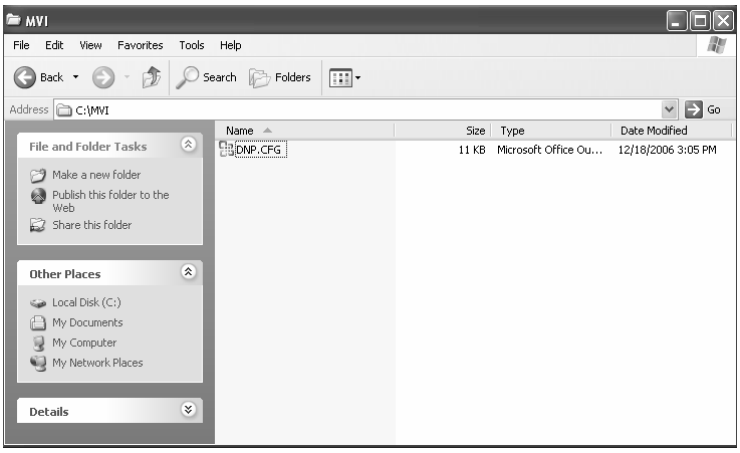
3 Press **[Y].** The screen now indicates that the module is ready to send.

4 From the **Transfer** menu in HyperTerminal, select **Receive File**. This action opens the Receive File dialog box.

5 Use the Browse button to choose a folder on your computer to save the file, and then click Receive.

- Note: ProSoft Technology suggests that you download the configuration file pre-loaded on your module. However, configuration files are also available on the ProSoft CD as well as the ProSoft Technology web site at http://www.prosoft-technology.com.

When the configuration file has been transferred to your PC, the dialog box will indicate that the transfer is complete.



6 You can now open and edit the file in a text editor such as Notepad. When you have finished editing the file, save it and close Notepad.

Important: You must name this file DNP.CFG before you transfer it to the module. The module will not recognize configuration files with any other name or extension.

2.5.2 Transferring the Configuration File to the Module

Perform the following steps to transfer a configuration file from your PC to the module.

1 Connect your PC to the Configuration/Debug port of the module using a terminal program such as HyperTerminal. Press **[?]** to display the main menu.

```
******* DNP DEBUG PORT HELP *******
 KEY     FUNCTION                  | KEY FUNCTION
 ------- --------------------------+---- -------------------------
 0-9,A-F Sets debug level          | Y   Class/Deadband Assignments
 L       Display error list        | U   Show DNP Databases
 P       Display setup & pointers  | <   Receive Configuration
 O       Operating parameters      | >   Send Configuration
 R       Reboot module             |
 S       Display Comm Stats        |
 W       Clear error list          | N   Display Blk X-fer Stats
 V       List COM States           | X   Master Port Commands
 T       Master Port Slave Setup   | Z   Master Port Slave Errs
 G       Version Information       | ?   Display this screen

PRODUCT = DNP5   REVISION = 2.35   OP SYS REV = 1206   PROD RUN # = 1501
```

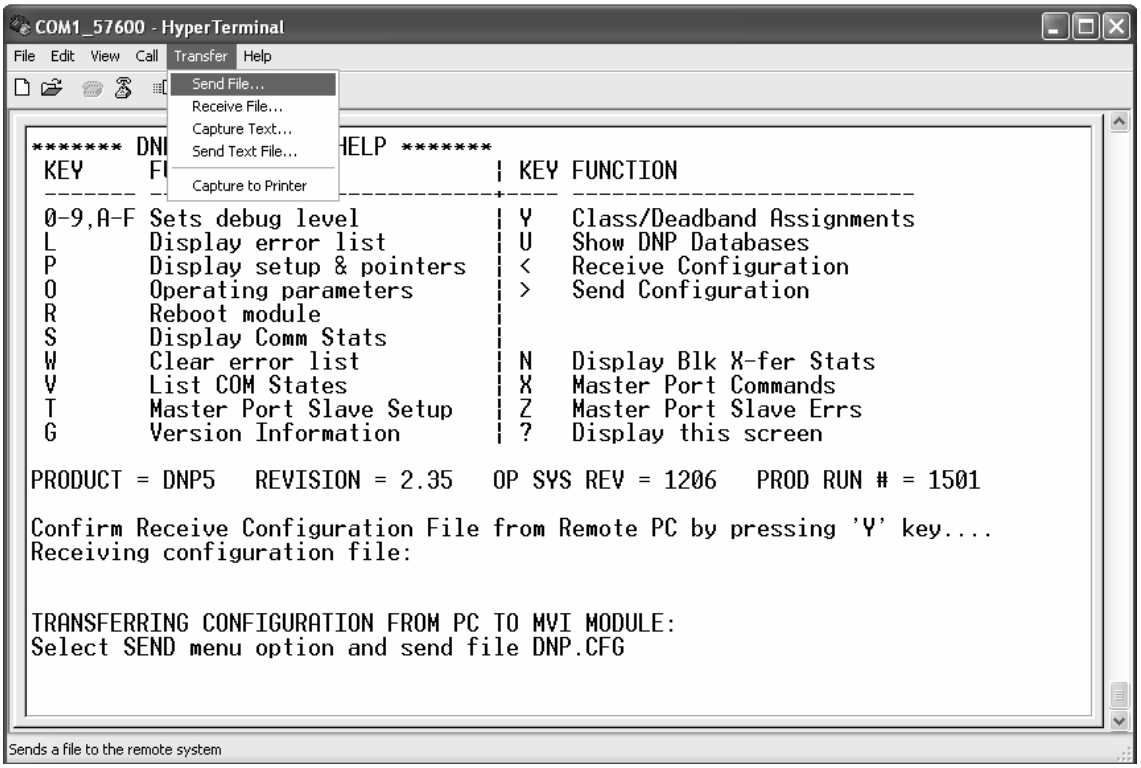
2 Press **[<]** (Receive Module Configuration). The message "Press Y key to confirm configuration receive!" is displayed at the bottom of the screen.

```
******* DNP DEBUG PORT HELP *******
 KEY     FUNCTION                  | KEY FUNCTION
 ------- --------------------------+---- -------------------------
 0-9,A-F Sets debug level          | Y   Class/Deadband Assignments
 L       Display error list        | U   Show DNP Databases
 P       Display setup & pointers  | <   Receive Configuration
 O       Operating parameters      | >   Send Configuration
 R       Reboot module             |
 S       Display Comm Stats        |
 W       Clear error list          | N   Display Blk X-fer Stats
 V       List COM States           | X   Master Port Commands
 T       Master Port Slave Setup   | Z   Master Port Slave Errs
 G       Version Information       | ?   Display this screen

PRODUCT = DNP5   REVISION = 2.35   OP SYS REV = 1206   PROD RUN # = 1501

Confirm Receive Configuration File from Remote PC by pressing 'Y' key....
```

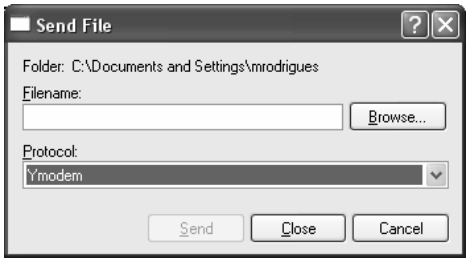
3 Press **[Y]**. The screen now indicates that the PC is ready to send.

```
******* DNP DEBUG PORT HELP *******
 KEY     FUNCTION                  | KEY FUNCTION
 ------- --------------------------+---- -------------------------
 0-9,A-F Sets debug level          | Y   Class/Deadband Assignments
 L       Display error list        | U   Show DNP Databases
 P       Display setup & pointers  | <   Receive Configuration
 O       Operating parameters      | >   Send Configuration
 R       Reboot module             |
 S       Display Comm Stats        |
 W       Clear error list          | N   Display Blk X-fer Stats
 V       List COM States           | X   Master Port Commands
 T       Master Port Slave Setup   | Z   Master Port Slave Errs
 G       Version Information       | ?   Display this screen

PRODUCT = DNP5   REVISION = 2.35   OP SYS REV = 1206   PROD RUN # = 1501

Confirm Receive Configuration File from Remote PC by pressing 'Y' key....
Receiving configuration file:


TRANSFERRING CONFIGURATION FROM PC TO MVI MODULE:
Select SEND menu option and send file DNP.CFG
```

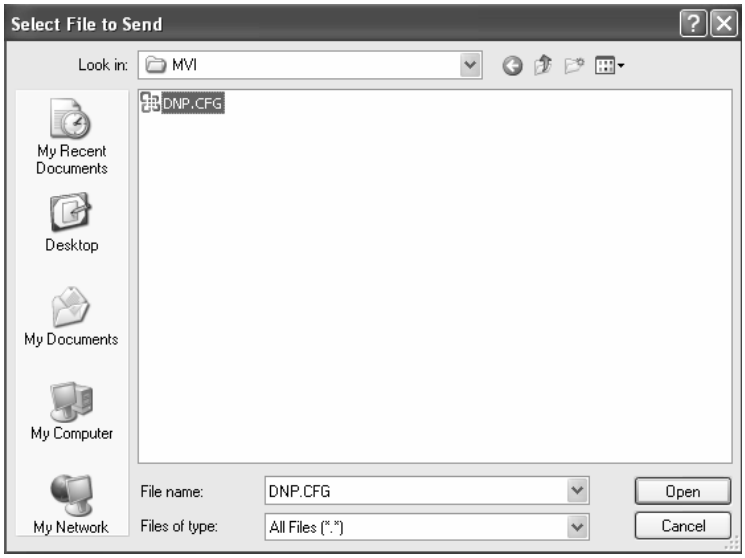
4 From the **Transfer** menu in HyperTerminal, select **Send File**.



The Send File dialog appears.

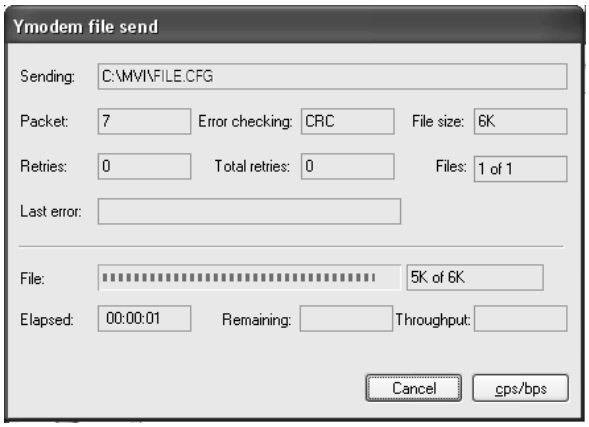


5 Use the Browse button to locate the configuration file your computer.

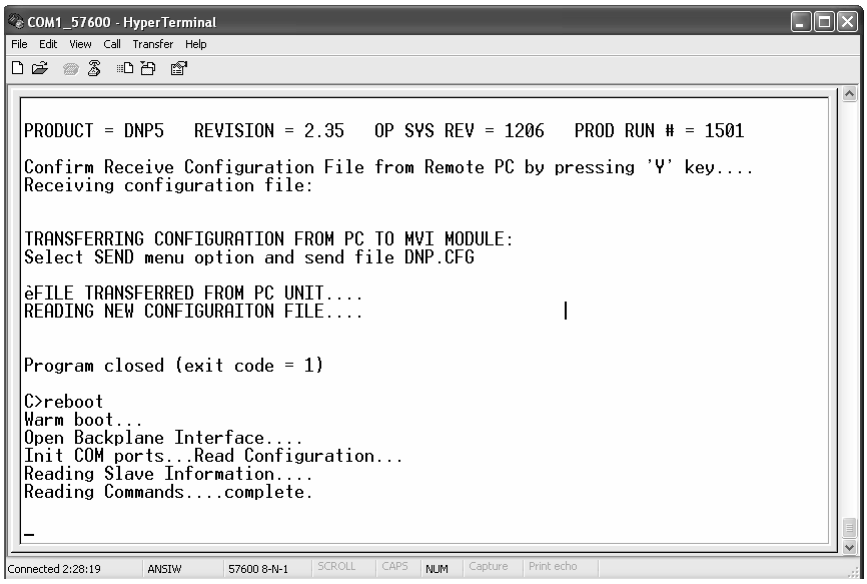


Note: This procedure assumes that you are uploading a newly edited configuration file from your PC to the module. However, configuration files are also available on the ProSoft CD as well as the ProSoft Technology web site.

6 Select **Ymodem** as the protocol.
7 Click the Send button. This action opens the Ymodem File Send dialog box.



When the upload is complete, the screen indicates that the module has reloaded program values and displays information about the module.



8 Your module now contains the new configuration. Press **[?]** to see the module's main menu.

3 Ladder Logic

In This Chapter

- Module Data Objects 41
- Special Objects 45
- Adding the Module to an Existing CompactLogix Project 47
- Adding the Module to an Existing MicroLogix Project 50

Ladder logic is required for application of the MVI69-DNP 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 Objects

All data related to the MVI69-DNP is stored in two user defined data types. One contains the status data and the other contains the DNP and IED datasets. Two structures were used for ease of use. Any time an array's size is altered in the RSLogix 5000 software, all the data in the object is set to zero. Because the array sizes may need to be adjusted for the data types in an application, this data is placed in a separate object.

An instance of each data type is required before the module can be used. This is accomplished by declaring variables of the data types in the Controller Tags Edit Tags dialog box. Each object is discussed in the following topics:

3.1.1 DNPModuleDef Object

The DNPModuleDef object contains all the MVI69-DNP module status data. The structure of the object is displayed in the following example:

Tag Name	Data Type	Description
GenStat	DNPSlvStat	General status information
ErrList	INT[60]	List of last 60 slave errors
BP	DNPBackplane	Data to handle backplane logic
CmdLstErr	DNPCmdLstErr	
ComErrTble	DNPComErrTble	
Readclock	DNPtime_n_date	
WriteClock	DNPtime_n_date	

Tag Name	Data Type	Description
cmd1	DNPCROB	
cmd2	DNPCROB	
cmd3	DNPCROB	
cmd4	DNPCROB	

This object contains objects that define the status related to the module. Each of these object types is discussed in the following topics of the document. The ErrList member of the object stores the list of last 60 errors generated by the module. This data is passed to the processor from the module the read block 100. Ladder logic transfers this information from the processor into the ErrList array.

3.1.2 DNPSlvStat Object

The DNPSlvStat object stores the status data passed from the module to the processor in the read blocks 112 to 114. This block of data contains information that can be used to determine the "health" of the module and the tasks running. The structure of the object is shown in the following example:

Tag Name	Data Type	Description
Cur_Port	Decimal	Current DNP Slave port
Last_Err	Decimal	Last DNP slave error code
Msg_Me	Decimal	Total message frames for slave
Msg_Sent	Decimal	Total message frames sent
Msg_Rec	Decimal	Total message frames received
Err_Sync	Decimal	Total number of synchronization errors
Err_Overrun	Decimal	Total number of over runs errors
Err_Length	Decimal	Total number of length errors
Err_CRC	Decimal	Total number of CRC errors
Err_Overflow	Decimal	Total number of overflow errors
Err_Seq	Decimal	Total number of sequence errors
Err_Address	Decimal	Total multi-frame request fragment errors
BI_Events	Decimal	Total number of BI events
BI_Buffer	Decimal	Total number of BI events in buffer
AI_Events	Decimal	Total number of AI events
AI_Buffer	Decimal	Total number of AI events in buffer
Err_Func	Decimal	Total number of bad function errors
Err_Obj	Decimal	Total number of object unknown errors
Err_Range	Decimal	Total number of range errors
Err_MOverflow	Decimal	Total number of app msg overflow errors
Err_Frame	Decimal	Total number of multi-frame errors
Blk_Total	Decimal	Total number of blocks transferred
Blk_Good	Decimal	Total number of blocks transferred successfully
Blk_Err	Decimal	Total number of blocks in error
Blk_RErr	Decimal	Total number of read block errors

Tag Name	Data Type	Description
Blk_WErr	Decimal	Total number of write block errors
Blk_NErr	Decimal	Total number of block number errors
Blk_ECntr	Decimal	Number of sequential block errors
Blk_EFlag	Decimal	Block error flag
Cfg_Type	Decimal	0=Single Slave, 1-Dual-Slave, 2=Slave/Master
Product	Decimal	Product name
Rev	Decimal	Revision
Op_Sys	Decimal	Operating system revision
Run	Decimal	Run number
Slave_Count	Decimal	Number of slaves configured
Cmd_Count	Decimal	Number of commands configured
Mem_Blk	Decimal	Number of memory blocks allocated
Mem_Frame	Decimal	Number of frame blocks allocated
Mem_DLRec	Decimal	Number of receive DL blocks allocated
Mem_DLTx	Decimal	Number of transmit DL blocks allocated
Mem_AppRec	Decimal	Number of receive app blocks allocated
Mem_AppTx	Decimal	Number of transmit app blocks allocated
Mem_DevErr	Decimal	Number of device mem alloc errors
Mem_PhyErr	Decimal	Number of physical layer mem alloc errors
Mem_DLRErr	Decimal	Number of data-link layer receive mem alloc errors
Mem_DLTErr	Decimal	Number of data-link layer transmit mem alloc errors
Mem_AppRErr	Decimal	Number of app layer receive mem alloc errors
Mem_AppTErr	Decimal	Number of app layer transmit mem alloc errors
Mstr_Sync	Decimal	Number of master port synchronization errors
Mstr_Length	Decimal	Number of master port length errors
Mstr_CRC	Decimal	Number of master port CRC errors
Scan_Count	Decimal	Program scan counter
Mem_Free	Decimal	Free memory for module
P1_TX_State	Decimal	State of Port 1 transmit state machine
FloatEvents	Decimal	Count of floating-point events
DoubleEvents	Decimal	Count of double floating-point events
EventQueue	Decimal	Total number of events in message queue for processor
EvtQueueOF	Decimal	Event message queue overflow flag (0=no, 1=yes)

Ladder logic is required to transfer the data sent from the module to the processor into this data object. If the ladder logic is present and the module is operating, this object can be viewed in the Controller Tags Monitor window to observer the current status of the module.

3.1.3 DNPBackplane Object

The DNPBackplane object stores the variables required for backplane data transfer between the module and the processor. The structure of the object is displayed in the following example:

Tag Name	Data Type	Description
LastRead	INT	Index of last read block
LastWrite	INT	Index of last write block
BlockIndex	INT	Computed block offset for data table

3.1.4 DNPData Object

The DNPData object stores all the data for an MVI69-DNP module. This includes the data to interface with the DNP slave port (DNP data set) and the DNP master port (IED data set). Contained within the object is an array for each data type that can exist for each of the two data sets. The array sizes are set to match the configuration set for the module. If multiple MVI69-DNP modules are used within a rack, a copy of this structure may have to be made to permit each module to have its own database sizes. Ladder logic is required to transfer the data in this structure between the module and the processor. Each data type has its own set of unique block identification codes to distinguish the data contained in the read or write block.

The structure of the object is shown in the following example:

Tag Name	Data Type	Description
DNP_BI	INT[480]	Number of DNP BI data words
DNP_BO	INT[480]	Number of DNP BO data words
DNP_Cntr	DINT[240]	Number of DNP counter double-words
DNP_AI	INT[480]	Number of DNP AI data words
DNP_FLTI	REAL[240]	
DNP_AO	INT[480]	Number of DNP AO data words
DNP_FLTO	REAL[240]	
IED_BI	INT[480]	Number of IED BI data words
IED_BO	INT[480]	Number of IED BO data words
IED_Cntr	DINT[240]	Number of IED counter double-wods
IED_AI	INT[480]	Number of IED AI data words
IED_AO	INT[480]	Number of IED AO data words

3.2 Special Objects

These objects utilize some of the advanced features the module provides. If your application does not require the object, then you need not declare an instance of the object. Each of the objects and associated function are discussed in the following topics.

3.2.1 DNPCROB Object

The DNPCROB object is used in conjunction with the command control block 9901. This block sends a pulse output command to a single-point relay or a trip/close relay. All the parameters required for each command to be used in the block are contained in the object. Up to six of these objects can be contained in a single block 9901 command. The structure of the object is shown in the following example:

Tag Name	Data Type	Description
Port_Flag	INT	This field is ignored in the current implementation
Slave_ID	INT	This is the DNP slave address to send the command to
Object	INT	This should always be 12
Variation	INT	This should always be 1
Function	INT	This should be 3, 5 or 6 depending on the write method
Address	INT	This is the binary output starting point to operate in the slave
Control_Code	INT	This determines the CROB operation
Pulse_Count	INT	This determines the number of pulses (0-255)
Pulse_On	INT	This determines the pulse on time
Pulse_Off	INT	This determines the pulse off time

3.2.2 DNPEventMsg Object

The DNPEventMsg object stores event messages received on the DNP master port and passed to the processor. The structure of this object is shown in the following example:

Tag Name	Data Type	Description
DevIndex	INT	Logical slave device index in module
IEDPoint	INT	Logical point address in IED database
DNPPoint	INT	Logical point address in DNP database
SlaveAddress	INT	Remote slave address that generated event
PointNum	INT	Point address in remote device
Object	INT	DNP object number for point
Variation	INT	DNP variation for event
LowTime	INT	Least-significant word of 48-bit DNP time
HighTime	DINT	Most-significant double-word of 48-bit DNP time
Value	DINT	Value for event

This information is passed to the processor from the module in a special read block with an identification code of 9903. Each block can send up to 20 event messages. Ladder logic must handle the receipt of this special data block and to place the data received into controller tags.

3.2.3 DNPSlave_Err Object

The DNPSlave_Err object stores the slave status information returned from the module after a 9949-block request from the ladder logic. An array of this object should be defined to hold the status data for each slave used by the module. The structure of the object is displayed in the following example:

Tag Name	Data Type	Description
Device_Index	INT	Index in the slave array for the master port
Slave_ID	INT	Slave address for device
Err_CRC	INT	Number of CRC errors
Err_Overflow	INT	Number of overflow errors
Err_Seq	INT	Number of sequence errors
Err_DLConf	INT	Number of data-link confirm retry errors
Err_DLCFail	INT	Number of data-link confirm failures
Err_AppResp	INT	Number of application response errors

3.2.4 DNP_BI_Event Object

The DNP_BI_Event object stores the information for a single binary input event to be sent from the processor to the module in a command block 9958. The structure shown contains all the parameters required for a binary input event.

Tag Name	Data Type	Description
EventCount	INT	Event Count
SeqCounter	INT	Sequence Counter
BinInptDataPt	INT	DNP Binary Input Data Point
MonDayState	INT	Month/Day/State
HourMin	INT	Hour/Minute
SecMilsec	INT	Second Millisecond
Year	INT	Year
Event2	INT[5]	Event #2
Event3	INT[5]	Event #3
Event4	INT[5]	Event #4
Event5	INT[5]	Event #5
Event6	INT[5]	Event #6
Event7	INT[5]	Event #7
Event8	INT[5]	Event #8
Event9	INT[5]	Event #9
Event10	INT[5]	Event #10
Event11	INT[5]	Event #11

3.2.5 DNP_AI_Event Object

The DNP_AI_Event object stores the information for a single analog input event to be sent from the processor to the module in a command block 9959. The structure shown contains all the parameters required for an analog input event.

Tag Name	Data Type	Description
EventCount	INT	Event Count
SeqCounter	INT	Sequence Counter
AIDataPt	INT	DNP Analog Input Data Point

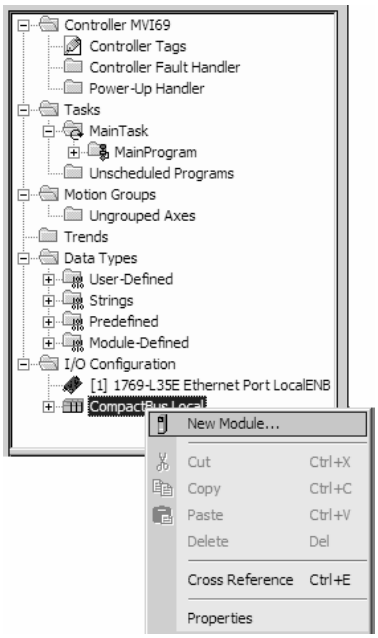
Tag Name	Data Type	Description
AIvalue	INT	DNP Analog Input Value
MonDay	INT	Month/Day
HourMin	INT	Hour/Minute
SecMilsec	INT	Second Millisecond
Year	INT	Year
Event2	INT[6]	Event #2
Event3	INT[6]	Event #3
Event4	INT[6]	Event #4
Event5	INT[6]	Event #5
Event6	INT[6]	Event #6
Event7	INT[6]	Event #7
Event8	INT[6]	Event #8
Event9	INT[6]	Event #9

3.3 Adding the Module to an Existing CompactLogix Project

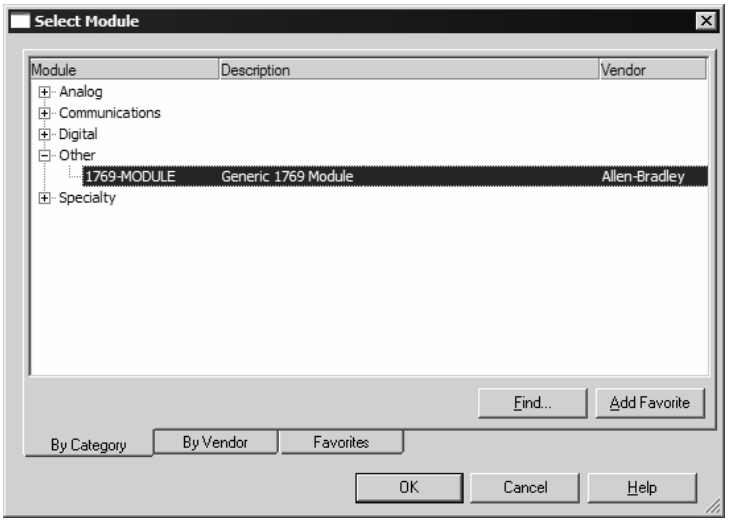
Important: The following steps describe how to install and configure the MVI69-DNP module with RSLogix 5000 version 15 or older. If you are using RSLogix 5000 version 16, please refer to Sample Add-On Instruction Import Procedure.
Important: The MVI69-DNP module has a power supply distance rating of 2 (L43 and L45 installations on first 2 slots of 1769 bus)

If you are installing and configuring the module with a CompactLogix processor, follow these steps. If you are using a MicroLogix processor, refer to the next section.

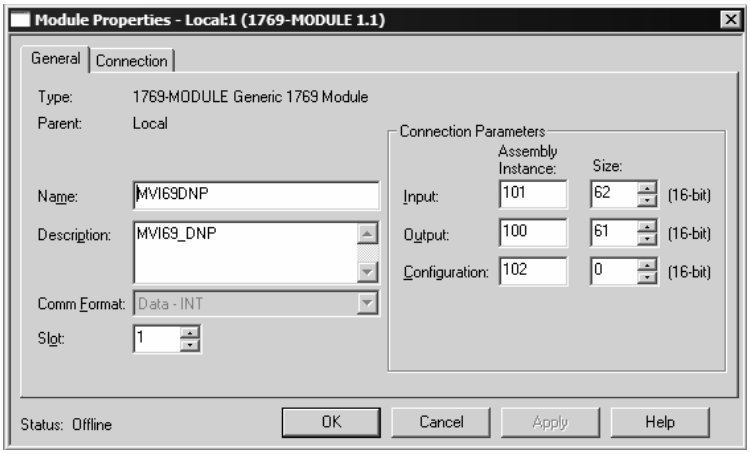
1 **Add the** MVI69-DNP **module to the project.** Right-click the mouse button on the I/O Configuration option in the Controller Organization window to display a pop-up menu. Select the New Module option from the I/O Configuration menu.



This action opens the following dialog box:

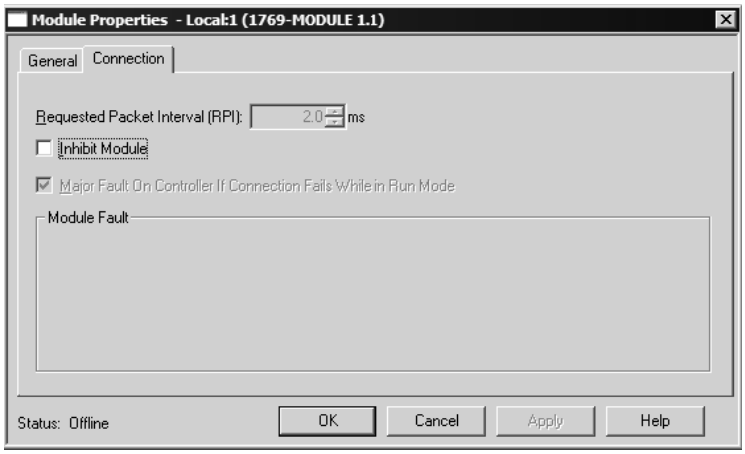


2 Select the 1769-Module (Generic 1769 Module) from the list and click OK.

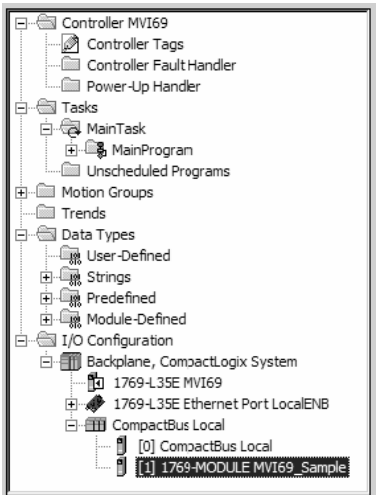


3 Enter the Name, Description and Slot options for your application, using the values in the illustration above. You must select the **Comm Format** as **Data - INT** in the dialog box, otherwise the module will not communicate over the backplane of the CompactLogix rack.

Click **OK** to continue.



4 Select the Request Packet Interval value for scanning the I/O on the module. This value represents the minimum frequency the module will handle scheduled events. This value should not be set to less than 1 millisecond. Values between 1 and 10 milliseconds should work with most applications.
5 Save the module. Click OK to dismiss the dialog box. The Controller Organization window now displays the module's presence. The following illustration shows the Controller Organization window:



6 Copy the Controller Tags from the sample program.
7 Copy the User Defined Data Types from the sample program.
8 Copy the Ladder Rungs from the sample program.
9 Save and Download the new application to the controller and place the processor in run mode.

3.4 Adding the Module to an Existing MicroLogix Project

If you are installing and configuring the module with a MicroLogix processor, follow these steps. If you are using a CompactLogix processor, refer to the previous section.

This chapter describes how to install and configure the module to work with your application. The configuration process consists of the following steps.

1 Use RSLogix to identify the module to the processor and add the module to a project.

Note: The RSLogix software must be in "offline" mode to add the module to a project.

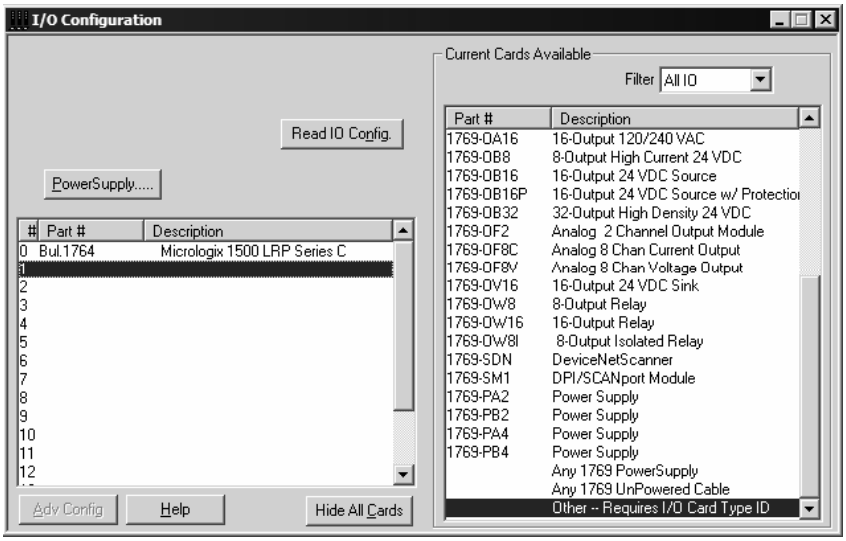
2 Modify the example ladder logic to meet the needs of your application, and copy the ladder logic to the processor. Example ladder logic files are provided on the CD-ROM.

Note: If you are installing this module in an existing application, you can copy the necessary elements from the example ladder logic into your application.

The rest of this chapter describes these steps in more detail.

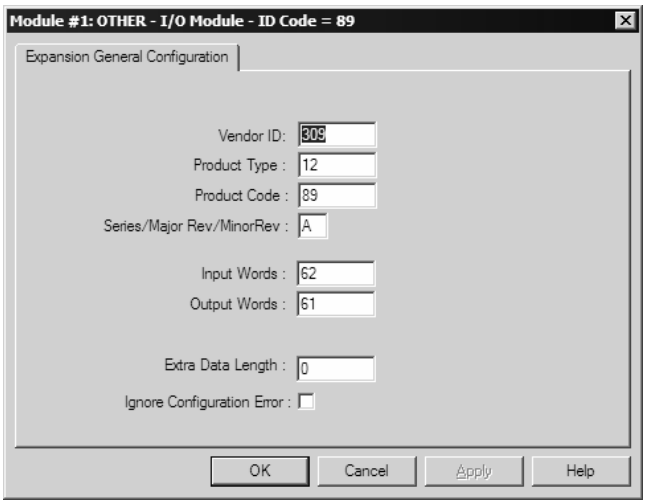
The first step in setting up the processor ladder file is to define the I/O type module to the system. Start RSLogix 500, and follow these steps:

1 In RSLogix, open your existing application, or start a new application, depending on your requirements.
2 Double-click the I/O Configuration icon located in the Controller folder in the project tree. This action opens the I/O Configuration dialog box.



3 On the I/O Configuration dialog box, select "Other - Requires I/O Card Type ID" at the bottom of the list in the right pane, and then double-click to open the Module dialog box.

4 Enter the values shown in the following illustration to define the module correctly for the MicroLogix processor, and then click OK to save your configuration.



The input words and output words parameter will depend on the Block Transfer Size parameter you specify in the configuration file. Use the values from the following table.

Block Transfer Size	Input Words	Output Words
60	62	61
120	122	121
240	242	241

5 Click **Next** to continue.

6 After completing the module setup, the I/O configuration dialog box will display the module's presence.

The last step is to add the ladder logic. If you are using the example ladder logic, adjust the ladder to fit your application. Refer to the example Ladder Logic section in this manual.

Download the new application to the controller and place the processor in run mode. If you encounter errors, refer to **Diagnostics and Troubleshooting** (page 53) for information on how to connect to the module's Config/Debug port to use its troubleshooting features.

4 Diagnostics and Troubleshooting

In This Chapter

- Reading Status Data from the Module 53
- LED Status Indicators..... 66

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 MVI69-DNP module returns a Status Data block that can be used to determine the module's operating status. This data is located in the module's database at a user set location. For a complete listing of the status data object, refer to the Reference chapter. The Configuration/Debug port provides the following functionality:

- Full view of the module's configuration data
- View of the module's status data
- Complete display of the module's internal database (registers 0 to 3999)
- Version Information
- Control over the module (warm boot, cold boot, transfer configuration)
- Facility to upload and download the module's configuration file

4.1.1 The Configuration/Debug Menu

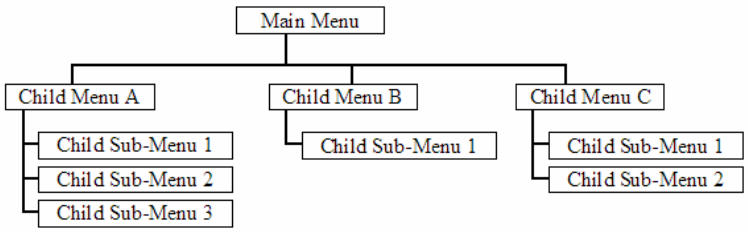
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

Because this is a text-based menu system, you enter commands by typing the command letter from your computer keyboard in the terminal application (for example, HyperTerminal). 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.

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.

4.1.2 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.

4.1.3 Required Software

In order to send and receive data over the serial port (COM port) on your computer to the module, you must use a communication program (terminal emulator).

A simple communication program called HyperTerminal is pre-installed with recent versions of Microsoft Windows operating systems. If you are connecting from a machine running DOS, you must obtain and install a compatible communication program. The following table lists communication programs that have been tested by ProSoft Technology.

DOS	ProComm, as well as several other terminal emulation programs
Windows 3.1	Terminal
Windows 95/98	HyperTerminal
Windows NT/2000/XP	HyperTerminal

The module uses the Ymodem file transfer protocol to send (upload) and receive (download) configuration files from your module. If you use a communication program that is not on the list above, please be sure that it supports Ymodem file transfers.

4.1.4 Using the Configuration/Debug Port

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

1. Connect your computer to the module's port using a null modem cable.
2. Start the communication program on your computer and configure the communication parameters with the following settings:

Baud Rate	57,600
Parity	None
Data Bits	8
Stop Bits	1
Software Handshaking	None

3. Open the connection. When you are connected, press the **[?]** key on your keyboard. If the system is set up properly, you will see a menu with the module name followed by a list of letters and the commands associated with them.

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. Verify that RSLinx is not controlling the COM port. Refer to Disabling the RSLinx Driver for the Com Port on the PC (page 90).
3. Verify that your communication software is using the correct settings for baud rate, parity and handshaking.
4. 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, you can contact ProSoft Technology Technical Support for further assistance.

4.1.5 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 DEBUG PORT HELP *******
 KEY     FUNCTION                  | KEY FUNCTION
 ------- --------------------------+---- -------------------------
 0-9,A-F Sets debug level          | Y   Class/Deadband Assignments
 L       Display error list        | U   Show DNP Databases
 P       Display setup & pointers  | <   Receive Configuration
 O       Operating parameters      | >   Send Configuration
 R       Reboot module             |
 S       Display Comm Stats        |
 W       Clear error list          | N   Display Blk X-fer Stats
 V       List COM States           | X   Master Port Commands
 T       Master Port Slave Setup   | Z   Master Port Slave Errs
 G       Version Information       | ?   Display this screen

PRODUCT = DNP5   REVISION = 2.35   OP SYS REV = 1206   PROD RUN # = 1501
```

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.

Setting the Debug Level

You can increase or decrease the level of debug messages sent from the module to the Debug Menu. The following table shows the type of debugging information for each key **[0]** to **[9]**, **[A]** to **[F]**

Key	None	DNP Statistics	Data Link Layer Messages	DPA Level Messages
0	X			
1		X		
2			X	
3		X	X	
4				
5		X		
6			X	
7		X	X	
8				X
9		X		X
A			X	X
B		X	X	X
C				X

Key	None	DNP Statistics	Data Link Layer Messages	DPA Level Messages
D		X		X
E			X	X
F		X	X	X

Viewing the Error List

Press **[L]** to display the last 60 errors for the DNP slave port. Refer to the error list section of the user manual to interpret each error recorded by the module.

If there are no errors present for the module, the message "NO ERRORS FOR SYSTEM!" is displayed.

Viewing DNP Set Up & Pointers

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

Viewing Operating Parameters

Press **[O]** to view the DNP Protocol setup information (Operating Parameters) for the module.

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 **[R]** 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.

Viewing Comm Stats

Press **[S]** to view the communication status for the DNP port.

Clearing the Error List

Press **[W]** to clear the error list. Use this command after viewing the error list (page 57) to delete the current list of errors and start a new list.

Viewing COM States

Press **[V]** to view the current state of the DNP application port and the port configuration information.

Viewing Master Port Slave Setup

Press **[T]** to view configuration information for the Master Port Slave.

Viewing Version Information

Press **[G]** 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.

Opening the Class Assignment Menu

Press **[Y]** to view the class and deadband override values for the binary, analog, float and double input DNP database.

Opening the DNP Database View Menu

Press **[U]** to open the DNP Database View 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 Database View Menu (page 61).

Receiving the Configuration File

Press **[<] (Shift Comma)** 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 (page 34).

Sending the Configuration File

Press **[>] (Shift Period)** 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 (page 34).

Viewing Block Transfer Statistics

Press **[N]** 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: Repeat this command at one-second intervals to determine the number of blocks transferred each second.

Opening the Command List Menu

Press **[X]** to open the Command List menu. Use this command to view the configured command list for the module.

```
***** DNP MASTER PORT COMMAND SELECTION MENU *****
PRODUCT = DNP     REVISION = 1.00    OP SYS REV = 0900    PROD RUN # = 1501
SELECT RANGE OF COMMANDS TO VIEW USING ONE OF THE KEYS INDICATED.
  KEY-COMMANDS    KEY-COMMANDS    KEY-COMMANDS    KEY-COMMANDS    KEY-COMMANDS
   0 -   0-19      1 -  20-39      2 -  40-59      3 -  60-79      4 -  80-99
   5 - 100-119     6 - 120-139     7 - 140-159     8 - 160-179     9 - 180-199
   A - 200-219     B - 220-239     C - 240-259     D - 260-279     E - 280-299
```

Opening the Command Error List Menu

Press **[Z]** to open the Command Error List. This list consists of multiple pages of command list error/status data. Press **[?]** to view a list of commands available on this menu.

```
***** DNP MASTER PORT SLAVE SELECTION MENU *****
PRODUCT = DNP     REVISION = 1.00    OP SYS REV = 0900    PROD RUN # = 1501
SELECT RANGE OF SLAVES TO VIEW USING ONE OF THE KEYS INDICATED.
  KEY-SLAVES    KEY-SLAVES    KEY-SLAVES    KEY-SLAVES
   0 -  0-9      1 - 10-19     2 - 20-29     3 - 30-39 _
```

4.1.6 The Class Assignment Menu

This menu allows you to view the class and deadband override values for the binary, analog, float and double input DNP database. Press **[?]** to display the commands available on this menu.

```
CLASS ASSIGNMENT MENU
 ?-Display Menu
 0-Binary Inputs
 1-Analog Inputs
 2-Float Inputs
 3-Double Inputs
 S-Show Again
 P-Previous Page
 N-Next Page
 M-Main Menu
```

The following illustration shows the output for the Analog data set (menu key **[1]**)

```
CLASS ASSIGNMENT DISPLAY 0 TO 19 (ANALOG INPUTS)
 POINT#   CLASS   DEADBAND
      0       1       1000
      1       1       1000
      2       1       1000
      3       1       1000
      4       1       1000
      5       1       1000
      6       1       2000
      7       1       2000
      8       2       1000
      9       3       2000
     10       2       1000
     11       2       1000
     12       1       1000
     13       1       1000
     14       1       1000
     15       1       1000
     16       1       1000
     17       1       1000
     18       1       1000
     19       1       1000
```

Viewing Binary Inputs

Press **[0]** to view the override values for Binary Input Data.

Viewing Analog Inputs

Press **[1]** to view the override values for Analog Input Data.

Viewing Float Inputs

Press **[2]** to view the override values for Float Input Data.

Viewing Double Inputs

Press **[3]** to view the override values for Double Input Data.

Redisplaying the Current Page

Press **[S]** to display the current page of data.

Viewing the Previous Page of Data

Press **[P]** to display the previous page of data.

Viewing the Next Page of Data

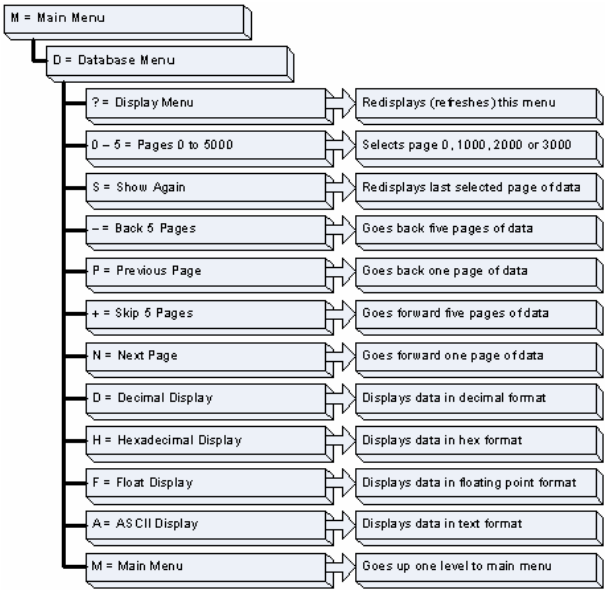
Press **[N]** to display the next page of data.

Returning to the Main Menu

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

4.1.7 DNP Database View Menu

Use this menu command to view the current contents of the selected database. Press **[?]** to view a list of commands available on this menu.



Viewing Data Type Databases

Press **[D]** from the DNP menu, then hold down the **[Shift]** key and press the ***/*** key.

```
DNP DATABASE VIEW MENU
 ?=Display Menu
 S=Show Again
 -=Back 5 Pages
 P=Previous Page
 +=Skip 5 Pages
 N=Next Page
 D=Word Decimal Display
 H=Word Hexadecimal Display
 L=Double Word Decimal Display
 X=Double Word Hexadecimal Display
 F=Float Display
 A=ASCII Display

 1=Binary Inputs
 2=Binary Outputs
 3=Counters
 4=Analog Inputs
 5=Analog Outputs
 6=Frozen Counters

 M=Main Menu
```

Use the number keys 1 to 6 to select the display of the data type you wish to view. For example, if the '1' key is pressed, the following is displayed:

```
DNP BINARY INPUT DATABASE DISPLAY 0 TO 1 (DECIMAL)
      0     0
```

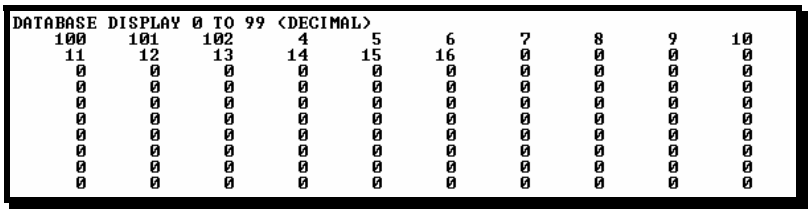
Viewing Register Pages

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



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.

Viewing Data in Double Word Decimal Format

Press **[L]** to display the data on the current page in Double Word Decimal format. This is useful for regions of the database that contain Double Word Decimal data.

Viewing Data in Double Word Decimal Format

Press **[X]** to display the data on the current page in Double Word Hexadecimal format. This is useful for regions of the database that contain Double Word Hexadecimal data.

Viewing DNP Binary Inputs

Press **[1]** to view a list of DNP Binary Inputs.

Viewing DNP Binary Outputs

Press **[2]** to view a list of DNP Binary Outputs.

Viewing DNP Counters

Press **[3]** to view a list of DNP Counters.

Viewing DNP Analog Inputs

Press **[4]** to view a list of DNP Analog Inputs.

Viewing DNP Analog Outputs

Press **[5]** to view a list of DNP Analog Outputs.

Viewing DNP Frozen Counters

Press **[6]** to view a list of DNP Frozen Counters.

Viewing DNP Float Inputs

Press **[7]** to view a list of DNP Float Inputs.

Viewing DNP Float Outputs

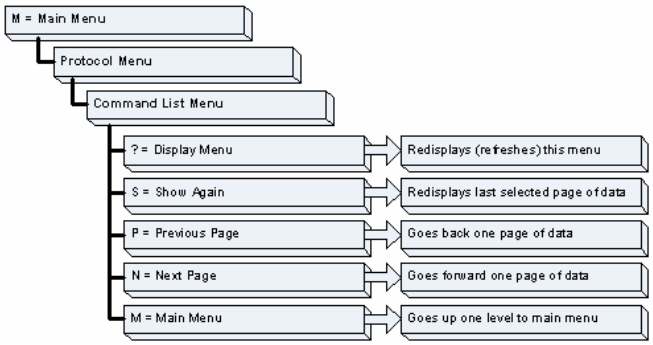
Press **[9]** to view a list of DNP Float Outputs.

Returning to the Main Menu

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

4.1.8 Master Command List Menu

Use this menu to view the command list for the module. Press **[?]** to view a list of commands available on this menu.



Redisplaying the Current Page

Press **[S]** to display the current page of data.

Viewing the Previous 50 Commands

Press **[-]** to view the previous 50 commands.

Viewing the Previous Page of Commands

Press **[P]** to display the previous page of commands.

Viewing the Next 50 Commands

Press **[+]** to view the next 50 commands from the master command list.

Viewing the Next Page of Commands

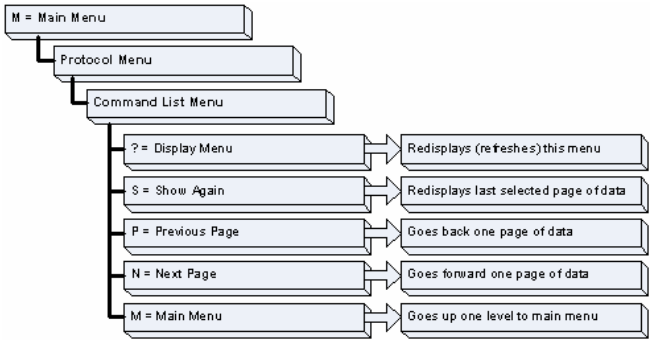
Press **[N]** to display the next page of commands.

Returning to the Main Menu

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

4.1.9 Master Command Error List Menu

Use this menu to view the command error list for the module. Press **[?]** to view a list of commands available on this menu.



Redisplaying the Current Page

Press **[S]** to display the current page of data.

Viewing the Previous 20 Commands

Press **[-]** to display data for the previous 20 commands.

Viewing the Previous Page of Commands

Press **[P]** to display the previous page of commands.

Viewing the Next 20 Commands

Press **[+]** to display data for the next 20 commands.

Viewing the Next Page of Commands

Press **[N]** to display the next page of commands.

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:

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	Data is being sent/received on this port
		Off	Waiting for data on this port
P2	Green	On	Data is being sent/received on this port
		Off	Waiting for data on this port
APP	Amber	Off	The MVI69-DNP is working normally.
		On	The MVI69-DNP module program has recognized a communication error on one of its DNP ports.
BP	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. Remove the card from the rack and re-insert the card to restart the module's program.
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 a configuration error is found for the client, the client configuration error word will have a value other than zero. The configuration error word bits have the following definitions:

Bit	Description	Value
0		0x0001
1		0x0002
2		0x0004
3		0x0008
4	Invalid retry count parameter	0x0010
5	The float flag parameter is not valid.	0x0020
6	The float start parameter is not valid.	0x0040
7	The float offset parameter is not valid.	0x0080
8		0x0100
9		0x0200
10		0x0400

Bit	Description	Value
11		0x0800
12		0x1000
13		0x2000
14		0x4000
15		0x8000

Correct any invalid data in the configuration for proper module operation. When the configuration contains a valid parameter set, all the bits in the configuration word will be clear. This does not indicate that the configuration is valid for the user application. Make sure each parameter is set correctly for the specific application.

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.

4.2.1 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 CompactLogix or MicroLogix 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.

4.2.2 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 has a power supply distance rating of 2 on Compact Logix. The module must be within 2 slots of the power supply on Compact Logix, or that the MicroLogix backplane can supply the 800ma required for the module.

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.

5 Reference

In This Chapter

- Product Specifications 69
- Functional Overview 72
- Cable Connections 89
- Error Status Table 95
- Internal Indication Word 103
- DNP V3.00 Device Profile Document 104
- DNP Subset Definition - Slave 105
- DNP Subset Definition - Master 112

5.1 Product Specifications

The MVI69 DNP 3.0 module is a single slot, backplane compatible DNP 3.0 interface solution for the CompactLogix platform. This module provides highly configurable support of both DNP 3.0 Master and Slave implementations (level 2 minimum), allowing the many SCADA and field devices supporting the DNP protocol to be integrated into the powerful CompactLogix platform.

The module supports DNP Subset Level 2 features and some of the Level 3 features allowing the many SCADA and field devices supporting the DNP protocol to be integrated into the CompactLogix platform. The module acts as an input/output module between the DNP network and the CompactLogix processor. The data transfer from the CompactLogix processor is asynchronous from the actions on the DNP network. Databases are user defined and stored in the module to hold the data required by the protocol.

5.1.1 General Specifications

- Single Slot - 1769 backplane compatible
- The module is recognized as an Input/Output module and has access to processor memory for data transfer between processor and module
- 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.
- Supports all CompactLogix processors: L20/L23/L30/L31/L32/L35, L43 and L45 (L43 and L45 supported with RSLogix 5000 v16.03 or later)
- Also supports MicroLogix 1500 LRP

5.1.2 Hardware Specifications

Specification	Description
Dimensions	Standard 1769 Single-slot module
Current Load	800 mA max@ 5 VDC
	Power supply distance rating of 2 (L43 and L45 installations on first 2 slots of 1769 bus)
Operating Temp.	0 to 60°C (32 to 140°F)
Storage Temp.	-40 to 85°C (-40 to 185°F)
Relative Humidity	5% to 95% (non-condensing)
LED Indicators	Battery and Module Status
	Application Status
	Serial Port Activity
	CFG Port Activity
CFG Port (CFG)	RJ45 (DB-9F with supplied cable)
	RS-232 only
	No hardware handshaking
App Ports (P1,P2) (Serial modules)	RS-232, RS-485 or RS-422 (jumper selectable)
	RJ45 (DB-9F with supplied cable)
	RS-232 handshaking configurable
	500V Optical isolation from backplane
Shipped with Unit	RJ45 to DB-9M cables for each port
	6-foot RS-232 configuration Cable

5.1.3 Functional Specifications

The MVI69-DNP module supports the DNP 3.0 protocol with a minimum of Level 2 functionality. DNP protocol Subset Definitions for the Master and the Slave drivers are available in the module's User Manual.

The module has two DNP protocol ports that can be user configured to operate in a Master/Slave or in a Slave/Slave redundant port configuration.

User defined internal register space is accessible to the protocol driver and to the CompactLogix processor memory.

Redundant Slave Port Operation

When configured in the Slave/Slave port configuration, the module's slave ports operate in a primary and secondary fashion. In this mode, a single host polls the module via redundant physical layer connections. Several methods are supported to automatically switch between the primary and secondary slave ports.

DNP 3.0 Slave Protocol Specifications

The DNP Slave port(s) accepts DNP commands to control and monitor data stored in the module's DNP Slave databases. If a DNP Master port is also configured, a portion of these slave databases can be derived from or can control IED devices connected to the DNP master port.

- Report-by-Exception data is logged to the module's database
- Supports unsolicited messaging
- Each DNP point type is user configurable by point
- Class assignments are completely user-definable on a Type and point basis (BI, AI, FI point types)
- The analog inputs are class and deadband configurable on a point basis for all formats (integer, float)
- Supports clock synchronization from a master or from the processor
- Support for four octet-strings are supported (object type 110) in the slave driver to return version and other module information
- Up to 400 events are stored for Analog Inputs (AI and FI) and 400 events for Binary Inputs
- In addition to the module generated events, AI and BI events can be generated in the processor and transferred to the module (useful with external timestamping hardware)
- Configurable event buffer transmission threshold based on count and/or time since last event transmission
- Collision avoidance algorithm per DNP organization for redundant port switching (redundant slave mode)
- Special modem AT command string and timing support for dialing out on redundant port (redundant slave mode)

DNP 3.0 Master Protocol Specifications

The DNP 3.0 Master port can be configured as a virtual DNP Master device that actively issues user-defined DNP commands to nodes on the network.

- The Master port supports 300 user defined commands, each one containing its own set of data link and application layer characteristics
- Master port logically supports up to 40 slave devices
- Individual command configuration includes conditional or continuous polling and Poll Delay Time
- Slave status and Command status available for transfer to the processor
- Event data received from the slave devices updates the module database (Date and Time stamping is not stored or used by module)
- Special command handling for Digital Output CROB under processor control for pulse output control
- Supports Report-by-Exception and Unsolicited Responses on a Time Interval basis or on a user determined Event Count basis. Analog and Binary input points are supported

5.2 Functional Overview

This section provides an overview of how the MVI69-DNP module transfers data using the DNP protocol. You should understand the important concepts in this chapter before you begin installing and configuring the module.

5.2.1 General Concepts

The following discussion explains several concepts that are important for understanding the operation of the MVI69-DNP module.

Module Power Up

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

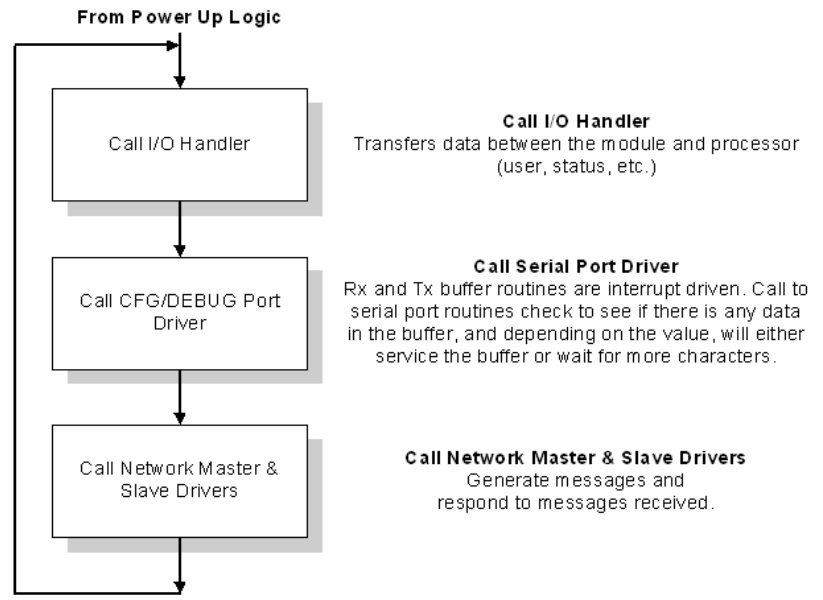
1 Initialize hardware components.
- Initialize CompactLogix or MicroLogix backplane driver.
- Test and Clear all RAM.
- Initialize the serial communication ports.
- Initialize the TCP/IP stack and Ethernet interface.

2 Read module configuration from DNP.CFG file on Compact Flash disk or Flash RAM (depending on hardware configuration).
3 Initialize the Module Register Space.
4 Enable Client and/or Server driver on selected ports, as configured.

When this initialization procedure is complete, 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 following functions:



Backplane Data Transfer

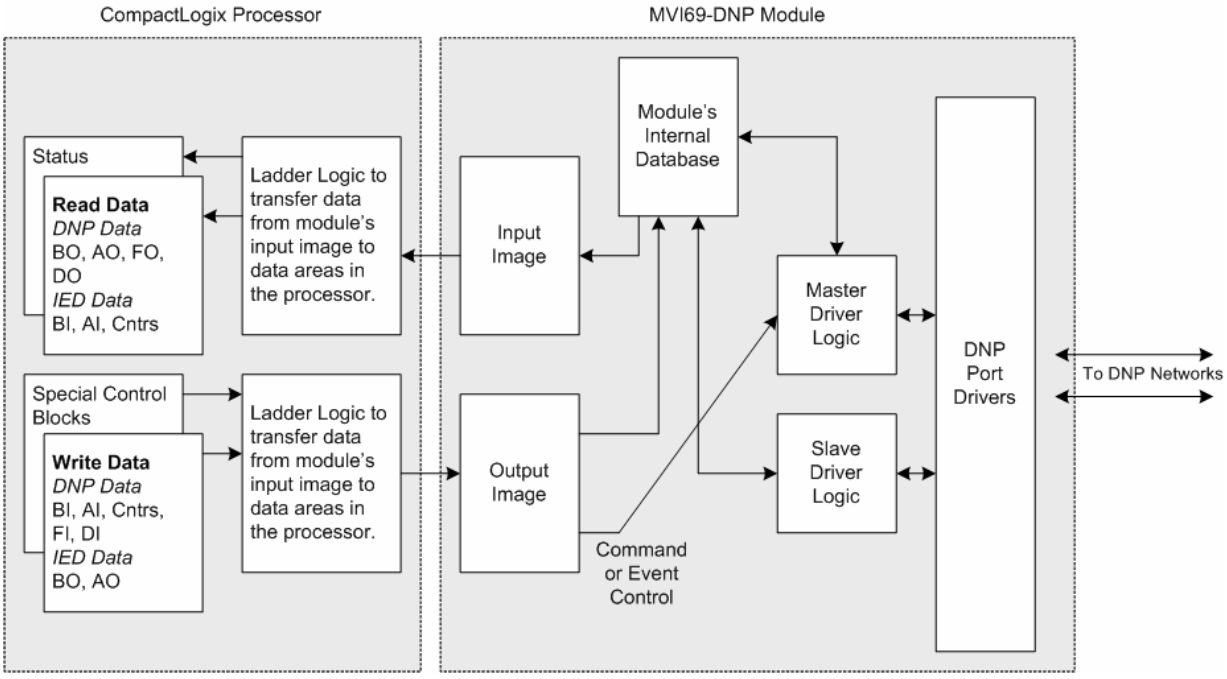
The MVI69-DNP module communicates directly over the CompactLogix or MicroLogix backplane. Data is paged between the module and the CompactLogix processor across the backplane using the module's input and output images. The update frequency of the images is determined by the scheduled scan rate defined by the user for the module and the communication load on the module. Typical updates are in the range of 2 to 10 milliseconds.

The data is paged between the processor and the module using input and output image blocks.

This bi-directional transference of data is accomplished by the module filling in data in the module's input image to send to the processor. Data in the input image is placed in the Controller Tags in the processor by the ladder logic. The input image for the module may be set to 62 words.

The processor inserts data to the module's output image to transfer to the module. The module's program extracts the data and places it in the module's internal database. The output image for the module may be set to 61 words.

The following illustration shows the data transfer method used to move data between the CompactLogix processor, the MVI69-DNP module and the DNP network.



All data transferred between the module and the processor over the backplane is through the input and output images. Ladder logic must be written in the CompactLogix processor to interface the block data with user data files. All data used by the module is stored in its internal database. The following illustration shows the layout of the database:

Data Area			Blocks
DNP DATA	BINARY INPUTS	PLC DATA	0 to 7
		IED DATA	
	BINARY OUTPUTS	PLC DATA	8 to 15
		IED DATA	
	COUNTER DATA	PLC DATA	16 to 23
		IED DATA	
	ANALOG INPUTS	PLC DATA	24 to 31
		IED DATA	
	ANALOG OUTPUTS	PLC DATA	32 to 39
		IED DATA	
	FLOAT INPUTS	PLC DATA	80 to 87
	DOUBLE INPUTS	PLC DATA	88 to 95
	FLOAT OUTPUTS	PLC DATA	96 to 103
	DOUBLE OUTPUTS	PLC DATA	104 to 111
	FROZEN COUNTER DATA		
	BINARY INPUT EVENTS		
	ANALOG INPUT EVENTS		
	FLOAT INPUT EVENTS		
	DOUBLE INPUT EVENTS		
LAST VALUE DATA	BINARY INPUTS		
	ANALOG INPUTS		
	FLOAT INPUTS		
	DOUBLE INPUTS		
	DNP BIN OUTPUTS		
	DNP ANAL OUTPUTS		
	IED BIN OUTPUTS		
	IED ANAL OUTPUTS		
IED DATA	BINARY INPUTS		40 to 47
	BINARY OUTPUTS		48 to 55
	COUNTER DATA		56 to 63
	ANALOG INPUTS		64 to 71
	ANALOG OUTPUTS		72 to 79
RBE FLAGS	BINARY INPUT		
	ANALOG INPUT		

Data contained in this database is paged through the input and output images by coordination of the CompactLogix ladder logic and the MVI69-DNP module's program. Up to 64 words of data can be transferred from the module to the processor at a time. Up to 64 words of data can be transferred from the processor to the module. The read and write block identification codes in each data block determine the function to be performed or the content of the data block. The module uses the following block numbers:

Block Number	function/Description
-1 to -2	Dummy blocks: Used when too few blocks are to be transferred.
0 to 39, 80 to 111	DNP Data blocks
40 to 79	IED Data blocks
112 to 114	Error/Status and Error List Blocks
9901	CROB Control Block for Digital Outputs
9902	Command Control Block (Add command to Command List Queue)
9903	Event Messages from Master port
9949	Slave IED unit errors on master port
9950	Command List Error data
9958	PLC Binary Input Event data
9959	PLC Analog Input Event Data
9970	Set PLC time using module's DNP time
9971	Set module's time using PLC time
9998	Warm Boot Request from PLC (Block contains no data)
9999	Cold Boot Request from PLC (Block contains no data)

Blocks 0 to 54 transfer data. Block 112 transfers error/status data. Blocks 9901 to 9999 are used for command control of the module. Each block group is described in the following topics.

Note: Block identification codes -1 and -2 are dummy blocks. The PLC should ignore any blocks with these numbers as data does not represent any valid information in the module. The DNP application will ignore any write blocks with these numbers.

5.2.2 Normal Data Transfer

Normal data transfer includes the paging of the user data found in the module's internal database in registers 0 to 4999 and the status data. These data are transferred through read (input image) and write (output image) blocks. The structure and function of each block is discussed in the following topics.

The block transfer numbers are fixed in the program for each DNP data type for this application. Block numbers are assigned by the application based on the number of points of each type. The application only allocates the number of blocks required to hold the data point count specified. For example, if 200 digital input points are required for the application, only block 0 is allocated and if 150 counter points are required, blocks 16 to 20 are defined. These are separate data areas for the DNP data used by the PLC and the data used by attached IED units that are not part of the DNP database in the module. Each block transfers 60 word-size registers.

DATA TYPE	START BLOCK #	MAX BLOCK #	MAX # OF POINTS
Digital Input	0	7	7680
Digital Output	8	15	7680
Counters	16	23	240
Analog Input	24	31	480
Analog Output	32	39	480

DATA TYPE	START BLOCK #	MAX BLOCK #	MAX # OF POINTS
IED Digital Input	40	47	7680
IED Digital Output	48	55	7680
IED Counters	56	63	240
IED Analog Input	64	71	480
IED Analog Output	72	79	480
DNP Float Input	80	87	240
DNP Float Output	88	95	240

These blocks are mapped into a pre-defined memory area in the module. Pointers into the memory area are computed by the application for each data type. The application will only copy the portion of the block that contains valid data.

Read Block

These blocks of data transfer information from the module to the CompactLogix processor. The structure of the input image used to transfer this data is shown below:

Offset	Description
0	Read Block ID
1	Write Block ID
2 to 61	Read Data

The Read Block ID is an index value used to determine the location of where the data will be placed in the CompactLogix processor controller tag array of module read data. Each transfer can move up to 60 words (block offsets 0 to 61) of data.

The Write Block ID associated with the block requests data from the CompactLogix processor. Under normal, program operation, the module sequentially sends read blocks and requests write blocks. For example, if three read and two write blocks are used with the application, the sequence will be as follows:

R1W1→R2W2→R3W1→R1W2→R2W1→R3W2→R1W1→

This sequence will continue until interrupted by other write block numbers sent by the controller or by a command request from a node on the DNP network or operator control through the module's Configuration/Debug port.

The following example shows a typical backplane communication application.

Write Block

These blocks of data transfer information from the CompactLogix or MicroLogix processor to the module. The structure of the output image used to transfer this data is shown below:

Offset	Description
0	Write Block ID
1 to 60	Write Data
61	Spare

The Write Block ID is an index value used to determine the location in the module's database where the data will be placed as defined in the table presented in the previous section. Each transfer can move up to 60 words (block offsets 1 to 60) of data.

5.2.3 Command Control Blocks

Command control blocks are special blocks used to control the module or request special data from the module. The current version of the software supports several command control blocks, each of which are discussed in the following topics.

Block 9901 - Event Command

This block issues one or more command control requests to slaves attached to the DNP master port for object 12 data. When the module receives a block 9901 identification code, it places the included commands into the command queue.

Block Format for Write

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the block identification code of 9901 for the block.
1	Command Count	This field defines the number of CROB blocks to generate. The valid range for the field is 1 to 6.
2 to 11	Command #1	Data for the command relay block (CROB) to be generated.
12 to 21	Command #2	Data for the command relay block (CROB) to be generated.
22 to 31	Command #3	Data for the command relay block (CROB) to be generated.
32 to 41	Command #4	Data for the command relay block (CROB) to be generated.
42 to 51	Command #5	Data for the command relay block (CROB) to be generated.
52 to 61	Command #6	Data for the command relay block (CROB) to be generated.

The following fields are used for each 10-word record in the command list:

Word Offset	Definitions	Description
0	Port/Flags	This field is currently ignored as all 9901 blocks are sent immediately out the master port.
1	Slave Address	This is the IED node address for the slave to consider on the network.
2	Object	Object type always 12
3	Variation	Variation always 1
4	Function	Function codes 3, 5 and 6 supported. Function code 4 is automatically sent after a successful function 3.
5	Address in Slave	Point in IED to consider with the CROB.
6	Control Code	This is a standard DNP protocol control code byte (see description below).
7	Pulse Count	This parameter specifies the number of pulses to generate for pulse output control. This parameter has a range of 0 to 255 as the value is a byte parameter in the CROB. If a value of zero is entered, the operation will not execute.
8	Pulse On Time	This parameter specifies the on-time interval for pulse control.
9	Pulse Off Time	This parameter specifies the off-time interval for pulse control.

The control code in the command is a bit coded byte value with the following definition:

Bits	Definitions	Description
0 to 3	Code	These bits determine the control operation to be performed by the command: 0=No operation, 1=Pulse on, 2=Pulse off, 3=Latch on and 4=Latch off. All other values are undefined in the DNP protocol.
4	Queue	0=Normal (execute once), 1=Requeue (place at end of queue after operation).
5	Clear	This parameter clears the queue. If the value is set to zero, the queue is not affected. If the value is set to 1, the queue will be cleared.
6 to 7	Trip/Close	These two bits select the trip or close relay. For close relay control, set the bits to 01. For trip relay control, set the bits to 10. A value of 00 for the bits is used for single point control of normal digital output points.

Block Format for Read

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the block identification code of 9901 for the block.
1	Next write block ID number	This field contains the ID number of the next write block.

Block 9902 - Command Control Block

The 9902 identification code is used by the PLC to send a list of commands to be placed in the command queue. Commands placed in the queue with this method need not have their enable bit set. The format of the block is as follows:

Block Format for Write

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9902 identifying the enable command to the module.
1	Command count	This field contains the number of commands to enable in the command list. Valid values for this field are 1 to 60.
2 to 61	Command Numbers to enable	These 60 words of data contain the command numbers in the command list to enable. The commands in the list will be placed in the command queue for immediate processing by the module. The first command in the list has an index of 0.

There is no response to this block by the module. The module will place the selected commands in the command queue. If the command references a IED unit that is not in the slave list, the command will not be placed in the command queue. Normal processing of the command list will continue after the commands specified in this block are processed.

Block Format for Read

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9902 identifying the enable command to the module.
1		
2	Number of commands added	This is the number of commands added to the command queue in the last 9902 block.

Block 9903 - Event Message Block

If the module contains a DNP master port and it is configured to pass event messages from the port to the processor, block identification code 9903 is used. When the master port receives an event message, it will place the data in the message into the event message queue. When the backplane task in the module recognizes data in this queue, it forms 9903 blocks to transfer the data to the processor. Ladder Logic extracts the event data from the 9903 block and place it in controller tags. The format of the block is as follows:

Block Format for Read

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the block identification code of 9903 for the block.
1	Block ID	This is the next block requested by the module.
2	Event Count	This field contains the number of events present in the block. Values of 1 to 4 are valid.

Word Offset in Block	Data Field(s)	Description
3 to 14	Event 1	Event message
15 to 26	Event 2	Event message
27 to 38	Event 3	Event message
39 to 50	Event 4	Event message
51 to 61	Spare	Not used

The format of each 12-word data region in the block is as follows:

Word Offset	Definitions	Description
0	Device Index	This field contains the module's device index for the IED the message was received from (0 to 39).
1	IED Address	This field contains the IED database index for the point. If set to -1, then not in database.
2	DNP Address	This field contains the DNP database index for the point. If set to -1, then not in database.
3	Slave Address	This field contains the remote slave address for the IED unit from which the message was received.
4	Point Number	This field contains the point number in the remote IED unit for the event message.
5	Object	This field contains the object code for the point and event.
6	Variation	This field contains the variation code for the point and event.
7	Low Time	This field contains the least-significant word of the 48-bit DNP time for the event.
8 to 9	High Time	This field contains the most-significant double word of the 48-bit time for the event.
10 to 11	Value	This field contains the a double word value for the point associated with the event message.

Block 9949 - Read Slave Communication Error Table

If the processor sends a block 9949, the module will respond with a slave communication error listing. The following example shows the format of the block:

Block Format for Write

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9949 identifying the block type to the module.
1	Number of slaves	This field contains the number of slaves to report in the response message. The value has a range of 1 to 7.
2	Start Slave Index	This parameter sets the index in the slave array where to start. The first slave in the array has a value of 0. The last index in the array has a value of MaxSlaves -1.
3 to 61	Spare	Not Used

The module responds to a valid request with a block 9949 containing the requested slave information. The following example shows the format of the block:

Block Format for Read

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9949 identifying the block type to the PLC.
1	Block ID	This is the next block requested by the module.
2	Slave Count	This field contains the number of slave records contained in the block that must be processed by the PLC. This field will have a value of 1 to 7.
3	Slave Start Index	This field contains the index in the slave array for the first record in the file. This field will have a value of 0 to MaxSlaves-1.
4 to 11	Slave Data #1	This is the slave data for the first slave in the block. The slave index for the data is the Slave Start Index given in word 3.
12 to 19	Slave Data #2	This is the slave data for the second slave in the block.
20 to 27	Slave Data #3	This is the slave data for the third slave in the block.
28 to 35	Slave Data #4	This is the slave data for the third slave in the block.
36 to 43	Slave Data #5	This is the slave data for the third slave in the block.
44 to 51	Slave Data #6	This is the slave data for the third slave in the block.
52 to 59	Slave Data #7	This is the slave data for the third slave in the block.
60 to 61	Spare	Not Used

Block 9950 - Read Command Error List

If the processor sends a block number of 9950 to the module, the application responds with a command error list. Each command in the system has a data word set aside for its last error code. The value is set by the DNP master port command list task and the values correspond to the error section of this manual. The format of the block is shown below:

Block Format for Write

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9950 identifying the block type to the module.
1	Number of Commands to report	This field contains the number of commands to report in the response message. The value has a range of 1 to 58.
2	Start Index of First Command	This parameter sets the index in the command list where to start. The first command in the list has a value of 0. The last index in the list has a value of MaxCommands -1.
3 to 61	Spare	Not Used

The module responds to a valid request with a block containing the requested error information. The format of the block is shown in the following example:

Block Format for Read

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9950 identifying the block type to the PLC.
1	Block ID	This is the next block requested by the module.
2	Number of Commands reported	This field contains the number of commands contained in the block that must be processed by the PLC. This field will have a value of 1 to 58.
3	Start Index of First Command	This field contains the index in the command list for the first value in the file. This field will have a value of 0 to MaxCommands-1.
4 to 61	Command List Errors	Each word of this area contains the last error value recorded for the command. The command index of the first value (offset 4) is specified in word 3 of the block. The number of valid command errors in the block is set in word 2 of the block. Refer to the command error list to interpret the error codes reported.

Block 9958 - Processor Binary Input Event

If the processor 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 message is shown in the following table.

Block Format for Write

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9958 identifying the event block to the module.
1	Event Count	This field contains the number of events contained in the block. Valid values for this field are 1 to 11.
2	Sequence Counter	This field holds the sequence counter for each 9958 block transfer. This synchronizes and confirms receipt of the block by the module.
3	DNP Binary Input Data point	This is the data point in the DNP binary input database represented by the event.
4	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.
5	Hour/Minute	Formatted: bits 0 to 5 = Minutes, bits 8 to 12 = Hour. All other bits are ignored.
6	Sec/Millisecond	Formatted: bits 0 to 9 = Milliseconds, bits 10 to 15 = Seconds.
7	Year	This is the four digit year for the event.
8 to 12		Five words of data for Event #2.

Word Offset in Block	Data Field(s)	Description
13 to 17		Five words of data for Event #3.
18 to 22		Five words of data for Event #4.
23 to 27		Five words of data for Event #5.
28 to 32		Five words of data for Event #6.
33 to 37		Five words of data for Event #7.
38 to 42		Five words of data for Event #8.
43 to 47		Five words of data for Event #9.
48 to 52		Five words of data for Event #10.
53 to 57		Five words of data for Event #11.
58 to 61	Spare	Not Used

To ensure the receipt of this block of information, the module returns a block 9958 with the sequence counter set to the value of the last successful block 9958 received.

Block Format for Read

Word Offset in Block	Data Field(s)	Description
0	Block ID	Identification code for block set to 9958.
1	Block ID	Block identification code for request from PLC by the module.
2	Event Count	This field contains the number of events processed by the module.
3	Sequence Counter	This field contains the sequence counter of the last successful block 9958 received.
4 to 61	Spare	Not used

Block 9959 - Processor Analog Input Event

If the processor 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 in the following table.

Block Format for Write

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9959 identifying the event block to the module.
1	Event Count	This field contains the number of events contained in the block. Valid values for this field are 1 to 9.
2	Sequence Counter	This field holds the sequence counter for each 9959 block transfer. This synchronizes and confirms receipt of the block by the module.
3	DNP Analog Input Data point	This is the data point in the DNP analog input database represented by the event.
4	Analog Input Value	This is the new analog input value represented in the event.

Word Offset in Block	Data Field(s)	Description
5	Month/Day	Formatted: bits 0 to 4 = Day, bits 8 to 11 = Month. All other bits are ignored.
6	Hour/Minute	Formatted: bits 0 to 5 = Minutes, bits 8 to 12 = Hour. All other bits are ignored.
7	Sec/Millisecond	Formatted: bits 0 to 9 = Milliseconds, bits 10 to 15 = Seconds.
8	Year	Four digit year value for event.
9 to 14		Six words of data for Event #2.
15 to 20		Six words of data for Event #3.
21 to 26		Six words of data for Event #4.
27 to 32		Six words of data for Event #5.
33 to 38		Six words of data for Event #6.
39 to 44		Six words of data for Event #7.
45 to 50		Six words of data for Event #8.
51 to 56		Six words of data for Event #9.
57 to 61	Spare	Not Used

To ensure the receipt of this block of information, the module returns a block 9959 with the sequence counter set to the value of the last successful block 9959 received.

Block Format for Read

Word Offset in Block	Data Field(s)	Description
0	Block ID	Identification code for block set to 9959.
1	Block ID	Block identification code for request from PLC by the module.
2	Event Count	This field contains the number of events processed by the module.
3	Sequence Counter	This field contains the sequence counter of the last successful block 9959 received.
4 to 61	Spare	Not used

Block 9970 - Set Processor Time Using Module Time

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

Block Format for Write

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9970 identifying the block type to the module.
1 to 61	Not Used	Not Used

The module will respond to a valid block 9970 request with a block containing the requested date and time. The following example shows the format of this block.

Block Format for Read

Word Offset in Block	Data Field(s)	Description
0	Block Read ID	This field contains the block identification code of 9970 for the block.
1	Block Write ID	This is the next block requested by the module.
2	Year	This field contains the four-digit year to be used with the new time value.
3	Month	This field contains the month value for the new time. Valid entry for this field is in the range of 1 to 12.
4	Day	This field contains the day value for the new time. Valid entry for this field is in the range of 1 to 31.
5	Hour	This field contains the hour value for the new time. Valid entry for this field is in the range of 0 to 23.
6	Minute	This field contains the minute value for the new time. Valid entry for this field is in the range of 0 to 59.
7	Seconds	This field contains the second value for the new time. Valid entry for this field is in the range of 0 to 59.
8	Milliseconds	This field contains the millisecond value for the new time. Valid entry for this field is in the range of 0 to 999.
9	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.
10 to 61	Not Used	Not Used

Block 9971 - Set Module's Time Using the Processor's Time

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

Block Format for Write

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the block identification code of 9971 for the block.
1	Year	This field contains the four-digit year to be used with the new time value.
2	Month	This field contains the month value for the new time. Valid entry for this field is in the range of 1 to 12.
3	Day	This field contains the day value for the new time. Valid entry for this field is in the range of 1 to 31.
4	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 Block	Data Field(s)	Description
5	Minute	This field contains the minute value for the new time. Valid entry for this field is in the range of 0 to 59.
6	Seconds	This field contains the second value for the new time. Valid entry for this field is in the range of 0 to 59.
7	Milliseconds	This field contains the millisecond value for the new time. Valid entry for this field is in the range of 0 to 999.
8 to 61	Not Used	Not Used

Warm Boot

This block is sent from the CompactLogix or MicroLogix processor to the module (output image) when the module is required to perform a warm-boot (software reset) operation. This block is commonly sent to the module any time configuration data modifications are made in the controller tags data area. This will force the module to read the new configuration information and to restart. The structure of the control block is shown below:

Offset	Description	Length
0	9998	1
1 to 63	Spare	63

Cold Boot

This block is sent from the CompactLogix processor to the module (output image) when the module is required to perform the cold boot (hardware reset) operation. This block is sent to the module when a hardware problem is detected by the ladder logic that requires a hardware reset. The structure of the control block is shown in the following table.

Offset	Description	Length
0	9999	1
1 to 63	Spare	63

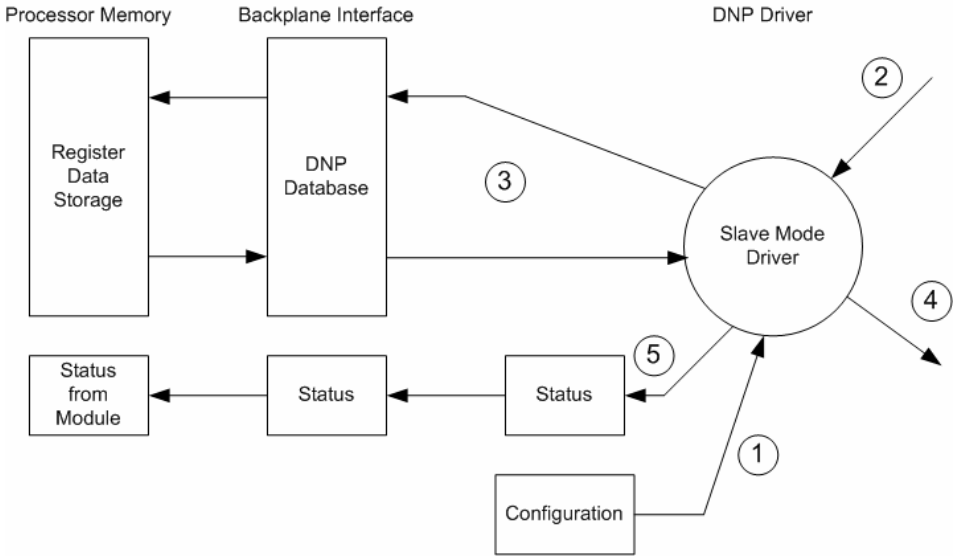
5.2.4 Data Flow between MVI69-DNP Module and CompactLogix Processor

The following topics describe the flow of data between the two pieces of hardware (CompactLogix processor and MVI69-DNP module) and other nodes on the DNP network under the module's different operating modes. Each port on the module is configured to emulate a DNP master device or a DNP slave device. The operation of each port is dependent on this configuration.

The following topics discuss the operation of these drivers.

Slave Driver

The Slave Driver mode allows the MVI69-DNP 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 slave port driver receives the configuration information from the configuration file. This information configures the serial port and define the slave node characteristics. Additionally, the configuration information contains data that can be used to offset data in the database to addresses requested in messages received from master units.
2	A host device (DNP Master Unit) issues a read or write command to the module's node address. The port driver qualifies the message before accepting it into the module.
3	After the module accepts the command, the data is immediately transferred to or from the internal database in the module. If the command is a read command, 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 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.

Review the Installing and Configuring the Module section for a complete list of parameters that must be defined for a slave port. The response messages from the slave driver include an IIN (internal indication word) defined in the Reference chapter - Internal Indication Word.

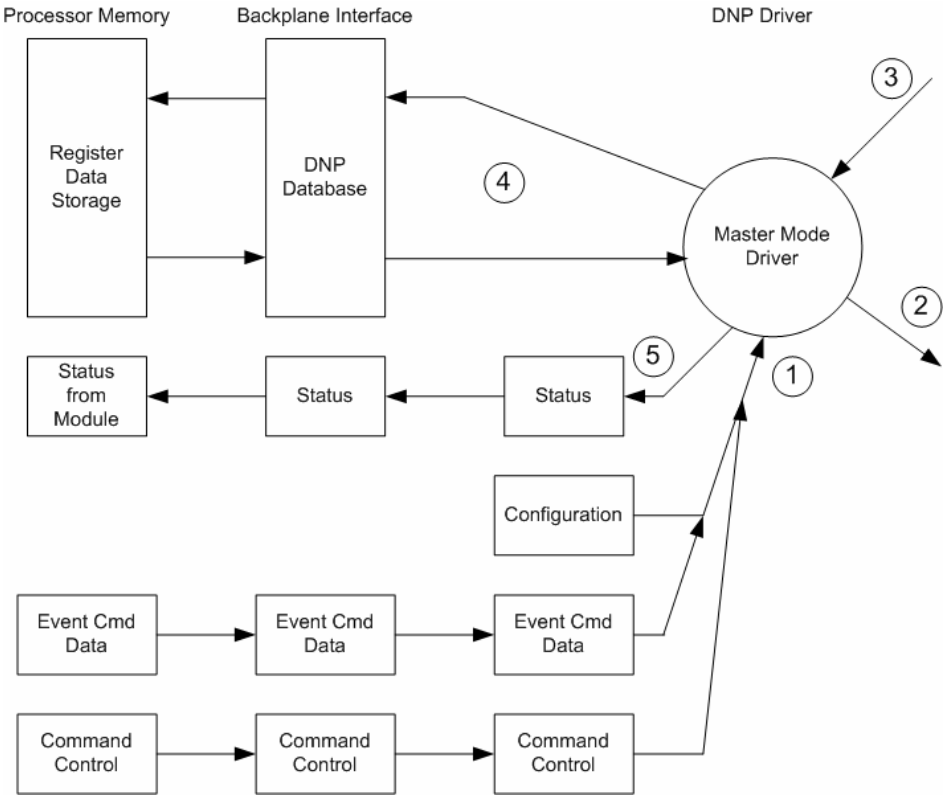
The slave driver supports Object 110 (octet string data). Four points are pre-assigned values as shown in the following table.

OCTET STRING POINT LIST

Point #	Description
0	Module Name as assigned in configuration file.
1	Product Name
2	Version Information in format: wwww xxxx yyyy zzzz Where wwww is product code, xxxx is the revision, yyyy is the operating system number and zzzz is the run number.
3	Manufacturer name for module.

Master Driver Mode

In the Master mode, the MVI69-DNP module is responsible for issuing read or write commands to slave devices on the DNP network. These commands are user-configured in the module via the Master Command List received from the processor or issued directly from the processor (command control). Command status is returned to the processor for each individual command in the command list status block. The following flow chart and associated table describe the flow of data into and out of the module.



Step	Description
1	The Master driver obtains configuration data from the configuration file. The configuration data obtained includes the Master Slave and Command Lists. These values are used by the Master driver to determine the type of commands to be issued to other nodes on the DNP network.
2	After configuration, the Master driver begins transmitting read and/or write commands to the other nodes on the network. If writing data to another node, the data for the write command is obtained from the module's internal database to build the command.
3	Presuming successful processing by the node specified in the command, a response message is received into the Master driver for processing.
4	Data received from the node on the network is passed into the module's internal database, assuming a read command.
5	Status data is returned to the CompactLogix processor for each command in the Master Command List.

Refer to the Installing and Configuring the Module section for a complete description of the parameters required to define the virtual DNP Master port.

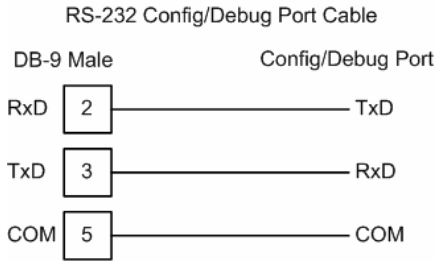
5.3 Cable Connections

The application ports on the MVI69-DNP module support RS-232, RS-422, and RS-485 interfaces. Please inspect the module to ensure that the jumpers are set correctly to correspond with the type of interface you are using.

Note: When using RS-232 with radio modem applications, some radios or modems require hardware handshaking (control and monitoring of modem signal lines). Enable this in the configuration of the module by setting the UseCTS parameter to 1.

5.3.1 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:

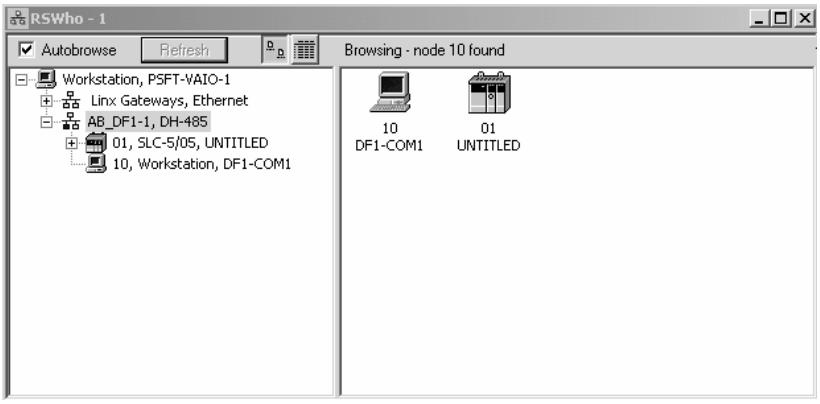


Disabling the RSLinx Driver for the Com Port on the PC

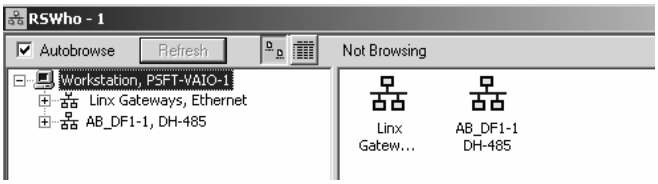
The communication port driver in RSLinx can occasionally prevent other applications from using the PC's COM port. If you are not able to connect to the module's configuration/debug port using ProSoft Configuration Builder (PCB), HyperTerminal or another terminal emulator, follow these steps to disable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

1 Open RSLinx and go to Communications>RSWho

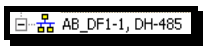
2 Make sure that you are not actively browsing using the driver that you wish to stop. The following shows an actively browsed network:



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:

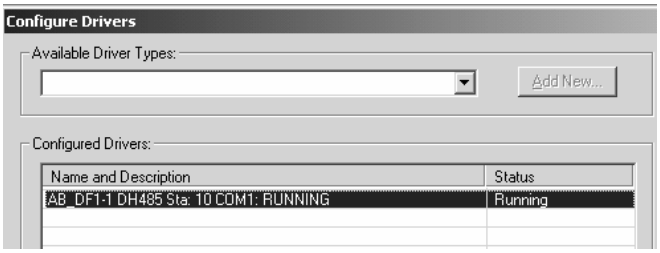


Branches are displayed or hidden by clicking on the ⊞ or the ⊟ icons.



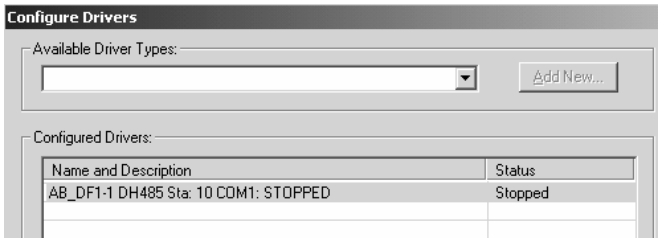
4 When you have verified that the driver is not being browsed, go to

Communications>Configure Drivers
You may see something like this:



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:

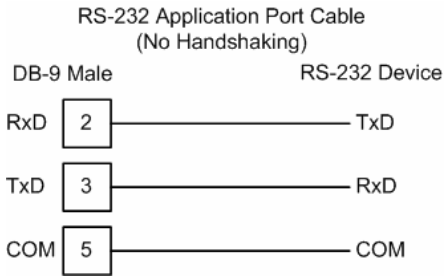


6 Upon seeing this, you may now use that com port to connect to the debug port of the module.

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.

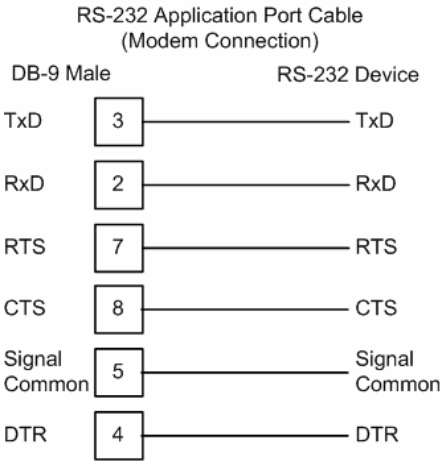
5.3.2 RS-232

When the RS-232 interface is selected, the use of hardware handshaking (control and monitoring of modem signal lines) is user definable. If no hardware handshaking will be used, the cable to connect to the port is as shown below:



RS-232: Modem Connection

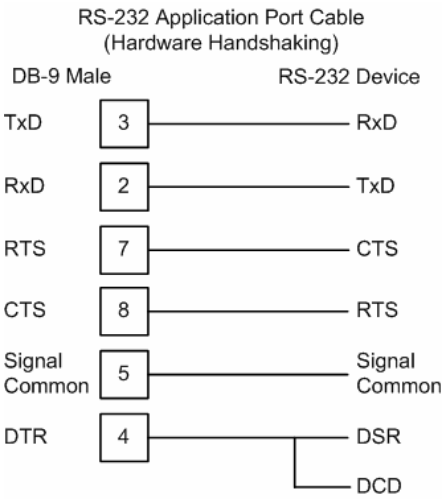
This type of connection is required between the module and a modem or other communication device.



The "Use CTS Line" parameter for the port configuration should be set to 'Y' for most modem applications.

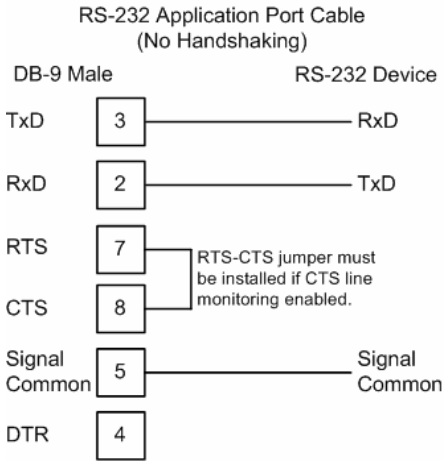
RS-232: Null Modem Connection (Hardware Handshaking)

This type of connection is used when the device connected to the module requires hardware handshaking (control and monitoring of modem signal lines).



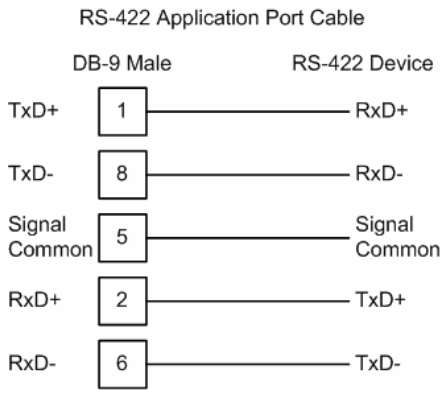
RS-232: Null Modem Connection (No Hardware Handshaking)

This type of connection can be used to connect the module to a computer or field device communication port.



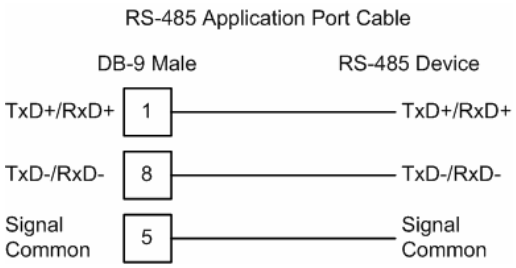
Note: If the port is configured with the "Use CTS Line" set to 'Y', then a jumper is required between the RTS and the CTS line on the module connection.

5.3.3 RS-422



5.3.4 RS-485

The RS-485 interface requires a single two or three wire cable. The Common connection is optional and dependent on the RS-485 network. The cable required for this interface is shown below:

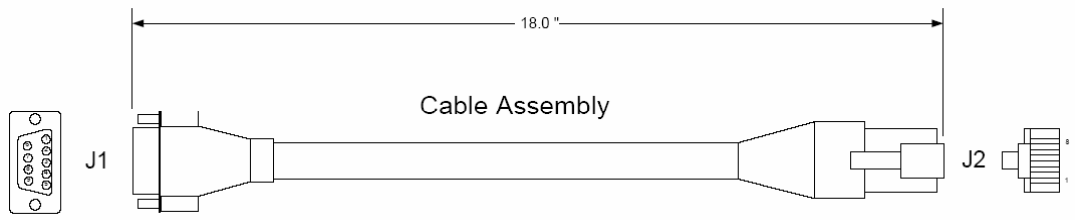


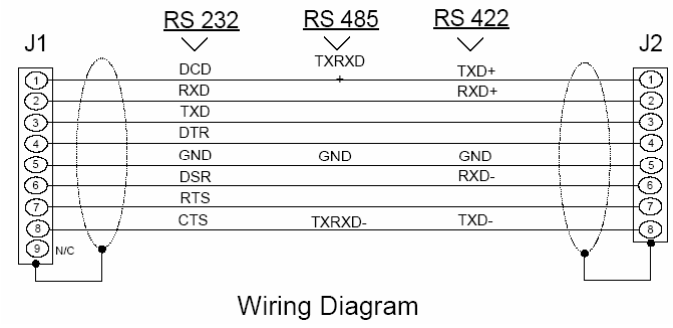
Note: Terminating resistors are generally not required on the RS-485 network, unless you are experiencing communication problems that can be attributed to signal echoes or reflections. In this case, install a 120 ohm terminating resistor on the RS-485 line.

RS-485 and RS-422 Tip

If communication in the RS-422/RS-485 mode does not work at first, despite all attempts, try switching termination polarities. Some manufacturers interpret +/- and A/B polarities differently.

5.3.5 DB9 to RJ45 Adaptor (Cable 14)





Wiring Diagram

5.4 Error Status Table

The program maintains an error status table. You can use the error/status data to determine the "health" of the module.

The data in the block is structured as follows:

Word	Block Offset	Variable Name	Description
0	2	Current DNP Slave Port status	This value represents the current value of the error code for the port. This value will only be valid if the port is configured as a slave. The possible values are described in the application documentation.
1	3	DNP Slave Port last transmitted error code	This value represents the last error code transmitted to the master by this slave port.
2	4	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.
3	5	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.
4	6	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.
5	7	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.
6	8	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.
7	9	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.
8	10	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.
9	11	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.
10	12	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.
11	13	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.

Word	Block Offset	Variable Name	Description
12	14	DNP Slave Binary Input Event count	This value contains the total number of binary input events which have occurred.
13	15	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.
14	16	DNP Slave Analog Input Event count	This value contains the total number of analog input events which have occurred.
15	17	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.
16	18	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.
17	19	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.
18	20	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.
19	21	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.
20	22	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.
21	23	Total blocks transferred	Total BTR/BTW or side-connect interface transfers attempted by the module.
22	24	Successful blocks transferred	This value represents the total number of transfer operations between the CompactLogix processor and module that are successful.
23	25	Total errors in block transfer	Total number of transfers that resulted in an error condition.
24	26	Total BTR or write errors	Total number of BTR or write transfers that resulted in an error.
25	27	Total BTW or read errors	Total number of BTW or read transfers that resulted in an error.
26	28	Block number error	Number of BTW requests that resulted in an incorrect BTW identification code.
27	29	Continuous block error counter	Count of sequential data transfer errors. When this value exceeds that specified for the data transfer operation, the error flag below will be set.
28	30	Reserved	Not used
29	31	Configuration Type	This is a coded field that defines the configuration of the module. The codes are as follows: 0=Single Slave Configuration, 1=Dual Slave Configuration, 2=Slave/Master Configuration

Word	Block Offset	Variable Name	Description
30 to 31	32 to 33	Product Name (ASCII)	These two words contain the product name of the module in ASCII format.
32 to 33	34 to 35	Revision (ASCII)	These two words contain the product revision level of the firmware in ASCII format.
34 to 35	36 to 37	Operating System Revision (ASCII)	These two words contain the module's internal operating system revision level in ASCII format.
36 to 37	38 to 39	Production Run Number (ASCII)	These two words contain the production 'batch' number for the particular chip in the module in ASCII format.
38	40	DNP Master Port Slave Count	This is the total number of slaves configured for the DNP Master port. This may not represent the number of active slaves as it includes slaves that are not enabled.
39	41	DNP Master Port Command Count	This is the total number of commands configured for the DNP Master port. This may not represent the number of active commands as it includes commands that are disabled.
40	42	DNP Master Port Device Memory Block Count	This value represents the number of memory allocation blocks for slave devices. This number should be one greater than the number of slave devices. The extra device is held for the broadcast device.
41	43	DNP Master Port Frame Block Count	This value represents the number of physical layer frame memory allocation blocks used by the program.
42	44	DNP Master Port Data Link Receive Block Count	This value represents the number of receive data link layer memory blocks allocated.
43	45	DNP Master Port Data Link Transmit Block Count	This value represents the number of transmit data link layer memory blocks allocated.
44	46	DNP Master Port Application Layer Receive Block Count	This value represents the number of application layer receive memory blocks allocated.
45	47	DNP Master Port Application Layer Receive Block Count	This value represents the number of application layer transmit memory blocks allocated.
46	48	DNP Master Port Device Memory Allocation Error Count	This value represents the number of memory allocation errors for device blocks.
47	49	DNP Master Port Physical Layer Memory Allocation Error Count	This value represents the number of memory allocation errors for physical layer frame blocks.
48	50	DNP Master Port Data Link Layer Receive Memory Allocation Error Count	This value represents the number of memory allocation errors for data link layer receive blocks.
49	51	DNP Master Port Data Link Layer Transmit Memory Allocation Error Count	This value represents the number of memory allocation errors for data link layer transmit blocks.

Word	Block Offset	Variable Name	Description
50	52	DNP Master Port Application Layer Receive Memory Allocation Error Count	This value represents the number of memory allocation errors for application layer receive blocks.
51	53	DNP Master Port Application Layer Transmit Memory Allocation Error Count	This value represents the number of memory allocation errors for application layer transmit blocks.
52	54	DNP Master 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.
53	55	DNP Master 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.
54	56	DNP Master Bad CRC Error Count (Physical Layer Error)	This value counts the number of times a bad CRC value is received in a message.
55	57	Scan Counter LSB	Program scan counter
56	58	Scan Counter MSB	
57	59	Free Memory LSB	Free memory in module
58	60	Free Memory MSB	
59	61	DNP Slave Port Transmit State	Value of the DNP Slave state machine for transmit.
60	62	DNP Float Event Count	Total number of events generated for analog floating-point input data points.
61	63	DNP Double Event Count	Total number of events generated for analog double, floating-point input data points.
62	64	Event Message Queue Count	Number of event messages waiting to send to processor.
63	65	Event Message Queue Overflow	Flag to indicate if the event message queue has overflowed. If more than 200 event messages are received on the master port and they are not sent to the processor, this flag will be set (1). The flag will clear after the messages are sent to the processor.
64 to 77	66 to 79	Reserved	Future Use
78	80	Error_List[0]	First value in error list
79	81	Error_List[1]	Second value in error list
-	-	-	-
137	139	Error_List[59]	Last value in error list

5.4.1 Module Error Codes

If the module's program encounters an error during execution, it will log the error to the error list. This list is transferred to the CompactLogix processor using block identification code 100 (see section above) in at offsets 62 to 119. This data is also available for viewing on the debug monitor port. The following tables list the error codes generated by the program with their associated description. Use the errors to help define where problems exist in the system.

Slave Port Communication Errors

Error Code	Name	Description
0	OK	The module is operating correctly and there are no errors.
10	DNP synchronization error (Physical Layer Error)	Extra bytes are received before the start bytes (0x05 and 0x64).
11	DNP overrun error (Physical Layer Error)	Mainline Data Link Layer routine could not read data received on DNP port before it was overwritten.
12	DNP length error (Physical Layer Error)	Length of message does not match length value in message.
13	DNP bad CRC error (Data Link Layer Error)	Computed CRC value for message does not match that received in message.
14	DNP user data overflow error (Transport Layer Error)	Application layer received a message fragment buffer which is too small.
15	DNP sequence error (Transport Layer Error)	Sequence numbers of multi-frame request fragments do not increment correctly.
16	DNP address error (Transport Layer Error)	Source addresses contained in multi-frame request fragments do not match.
17	DNP bad function code error (Application Layer Error)	Function code received from DNP master is not supported for selected object/variation.
18	DNP object unknown error (Application Layer Error)	Slave does not have the specified objects or there are no objects assigned to the requested class.
19	DNP out of range error (Application Layer Error)	Qualifier, range or data fields are not valid or out of range for the selected object/variation.
20	DNP message overflow error (Application Layer Error)	Application response buffer overflow condition. The response message from the slave is too long to transmit.
21	DNP master multi-frame message error (Application Layer Error)	Received a multi-frame message from the DNP master. This application does not support multi-frame messages from the master.

System Configuration Errors

Error Code	Name	Description
100	Too many binary input points	Too many binary input points are configured for the module. Maximum value is 7680
101	Too many binary output points	Too many binary output points are configured for the module. Maximum value is 7680
102	Too many counter points	Too many counter points are configured for the module. Maximum value is 240
103	Too many analog input points	Too many analog input points are configured for the module. Maximum value is 480
104	Too many analog input points	Too many analog output points are configured for the module. Maximum value is 480
105	Too many binary input events	Too many binary input events are configured for the module. Maximum value is 400.
106	Too many analog input events	Too many analog input events are configured for the module. Maximum value is 400.
107	Invalid analog input deadband	Deadband value for analog input events is out of range. Value must be in the range of 0 to 32767.
108	Not enough memory	There is not enough memory in the module to configure the module as specified.
109	Invalid block transfer delay for blocks error/status blocks	Block transfer delay value specified is too low.
110	File count invalid	The file count must be in the range of 0 to 6.
111	Invalid file record size	The file record size must be in the range of 1 to 120.
112	Invalid block identification code for file	The file block transfer code must be in the range of 100 to 120.

DNP Port Configuration Errors

Error Code	Name	Description
212	Invalid DNP address	The DNP address specified in the configuration is not valid (0 to 65534).
213	Invalid DNP port baud rate	The baud rate code specified in the configuration is not valid.
219	Invalid DNP data link layer confirm mode	The data link confirmation mode code is not valid in the configuration.
220	Invalid DNP data link confirm time-out	The data link time-out period specified in the configuration is 0. It must be an integer in the range of 1 to 65535.
222	Invalid DNP select/operate arm time duration	The select/operate arm timer is set to 0. It must be an integer in the range of 1 to 65535.
223	Invalid DNP application layer confirm time-out	The application layer confirm time-out value is set to 0. It must be an integer in the range of 1 to 65535.
224	Invalid DNP write time interval	The write time interval is not in the data range in the configuration. The value must be in the range of 0 to 1440.

Error Code	Name	Description
225	Invalid DNP unsolicited response mode	The unsolicited response mode code is not valid in the configuration.
226	Invalid DNP unsolicited response minimum quantity for Class 1	The unsolicited response minimum quantity for Class 1 is not valid in the configuration. Value must be an integer in the range of 1 to 255.
227	Invalid DNP unsolicited response minimum quantity for Class 2	The unsolicited response minimum quantity for Class 2 is not valid in the configuration. Value must be an integer in the range of 1 to 255.
228	Invalid DNP unsolicited response minimum quantity for Class 3	The unsolicited response minimum quantity for Class 3 is not valid in the configuration. Value must be an integer in the range of 1 to 255.
230	Invalid DNP unsolicited response destination address	The unsolicited response destination address is not valid in the configuration. Value must be in the range of 1 to 65534.

Command Error Codes

General Command Errors

Error Code	Name	Description
1	Device not defined	The IED slave address referenced in the command is not defined in the module. Check to make sure there is an entry in the slave table for each slave device referenced in the command list.
2	Invalid command	This command is not valid. Check to make sure the slave address parameter is greater than or equal to zero and that the point count is not set to zero.
3	Object not supported	The data object in the command is not supported by the module. Refer to the DNP subset for the Master Port.
4	Command function not supported	The function specified in the command is not supported for the object type selected. Refer to the DNP subset for the Master Port.
10	Invalid binary input poll command	This binary input object command is not valid.
11	Invalid binary input event poll command	This binary input event object poll command is not valid.
20	Invalid binary output command function	This binary output command function is not valid.
30	Invalid counter poll command function	The counter object poll command contains an invalid function code.
31	Invalid counter poll command	This counter object poll command is not valid.
40	Invalid frozen counter poll command	This frozen counter object poll command is not valid.
50	Invalid analog input poll command	This analog input poll command is not valid.
51	Invalid analog input event poll command	This analog input event poll command is not valid.

Error Code	Name	Description
60	Invalid analog output poll command function	This analog output poll command contains an invalid function code.
61	Invalid analog output poll command	This analog output poll command is not valid.
70	Invalid time/date poll command	This time/date object poll command is not valid.
80	Invalid event poll command	This event poll command is not valid.

Application Layer Errors

Error Code	Name	Description
1000	Device index invalid	The device index in the request or response message is not found in the slave list.
1001	Duplicate request in application layer queue	The newly submitted message to the application layer already exists in the queue. The message is ignored.
1002	COM port device removed from system	The communication port for the message has been uninstalled on the system. This error should never occur as the communication ports are only uninstalled when the module's program is terminated.
1003	Sequence number error	The application sequence number in the response message does not match that based on the last request message. This indicates application layer messages are received out of order.
1004	Response to select before operate does not match	The select response message received from the slave module is not that expected from the last select request. This indicates a synchronization problem between the master and slave devices.
1005	Response does not contain date/time object	The response message from the slave device does not contain a date/time object. The master expects this object for the response message.
1006	Time-out condition on response	The slave device did not respond to the last request message from the master within the time-out set for the IED device. The application layer time-out value is specified for each IED unit in the slave configuration table in the module. This table is established each time the module performs the restart operation.
1007	Function code in application layer message not supported	The function code returned in the response message is not valid for the application layer or not supported by the module.
1008	Read operation not supported for object/variation	The application layer response message contains an object that does not support the read function.
1009	Operate function not supported for the object/variation	The application layer response message contains an object that does not support the operate function.
1010	Write operation not supported for the object/variation	The application layer response message contains an object that does not support the write function.

5.5 Internal Indication 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 master station know broadcast 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 master. The master should write the date and time when this 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 requested operation is already executing. The slave will never set this bit.
5	Bad configuration. The slave configuration is invalid and should be re-configured. If the configuration is invalid, the slave will set the invalid parameters to default values and continue to run. Check error log using debug port.
6	Reserved, always 0.
7	Reserved, always 0.

5.6 DNP V3.00 Device Profile Document

Vendor Name:	ProSoft Technology, Inc.
Device Name:	MVI69-DNP (Revision 1.00)

Highest DNP Level Supported:	Device Function:
For Request: L2	Slave & Master
For Responses: L2	

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

Definition of selected IIN bits: Configuration Error - User specified point or event count is too high for application (can only correct by changing configuration in PLC).

Support for a redundant slave port on the module which may be attached to a dial-up modem is provided. Auto switching is provided by the module to switch between the primary and secondary ports.

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

Events generated by IED units attached to a master port may pass their events directly to the slave port. These events may not occur in the correct time sequence. They are placed in the event buffer as the module receives them. This provides the greatest time resolution for remote events.

Module will not generate events until Restart IIN bit is cleared by DNP master except for events passed through module from attached IED units.

Maximum Data Link Frame Size (octets):	Maximum Application Fragment Size (octets):
Transmitted: 292	Transmitted: 2048
Received: 292	Received: 2048
Maximum Data Link Re-tries:	Maximum Application Layer Retries:
Configurable from 0 - 255	None

Requires Data Link Layer Confirmation:

Configurable at module start-up (never, sometimes, & always)

Requires Application Layer Confirmation:

When reporting Event Data as a slave unit

Time-outs while waiting for:

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

Sends/Executes Control Operations:

WRITE Binary Outputs	: Never
SELECT/OPERATE	: Always
DIRECT OPERATE	: Always
DIRECT OPERATE-NO ACK	: Always
Count > 1	: Always (1 to 65535)
Pulse On	: Always
Pulse Off	: Always
Latch On	: Always
Latch Off	: Always

Queue : Never	
Clear Queue : Never	
Reports Binary Input Change Events when no specific variation requested: Only time-tagged 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. Default Counter Object/Variation: Object : 20 Variation : 5	Reports time-tagged Binary Input Change Events when no specific variation requested: Binary Input Change with Time Sends Static Data in Unsolicited Responses: Never Counters Roll Over at: 32 Bits
Sends Multi-Fragment Responses: Yes	

5.7 DNP Subset Definition - Slave

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

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

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

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	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 respond with variation 4 data
	1	32-Bit Analog Input	1	06	129, 130	00, 01	40	Slave will return this variation (Note: Data will only be 16-bit)
	2	16-Bit Analog Input	1	06	129, 130	00, 01	24	Slave will return this variation
	3	32-Bit Analog Input Without Flag	1	06	129, 130	00, 01	32	Slave will return this variation (Note: Data will only be 16-bit)
	4	16-Bit Analog Input Without Flag	1	06	129, 130	00, 01	16	Slave will return this variation
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

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 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
	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

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
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 Date With Interval					80	Slave will return Unknown Object to this request
51	0	Time and Date CTO: All Variations						Slave will return Unknown Object to this request
	1	Time and Date CTO			129, 130	07, With Quant=1	48	Slave will return Unknown Object to this request
	2	Unsynchronized Time and Date CTO			129, 130	07, With Quant=1	48	Slave will return Unknown Object to this request
52	0	Time Delay: All Variations						
	1	Time Delay Coarse			129	07, With Quant=1	16	Slave will never return this variation
	2	Time Delay Fine			129	07, With Quant=1	16	Slave will return this variation to functions 0D, 0E, and 17
60	0	Not Defined						Not Defined in DNP
	1	Class 0 Data	1	06				Slave will respond to this variation with all static data
	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						
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

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	1	Storage Object						
82	0	Not Defined						Not Defined in DNP
	1	Device Profile						
83	0	Not Defined						Not Defined in DNP
	1	Private Registration Object						
	2	Private Registration Objection Descriptor						
90	0	Not Defined						Not Defined in DNP
	1	Application Identifier						
100	0							
	1	Short Floating Point					48	
	2	Long Floating Point					80	
	3	Extended Floating Point					88	
101	0							
	1	Small Packed Binary-Coded Decimal					16	
	2	Medium Packed Binary-Coded Decimal					32	
	3	Large Packed Binary-Coded Decimal					64	
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

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
			23					Slave supports the Delay Measurement & Time Synchronization Function and will return Obj 52, Var 2, Qual 7, Cnt 1

5.8 DNP Subset Definition - Master

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	Master will generate this variation
	1	Binary Input	1	06	129, 130	00, 01	1	Master will generate and process this variation
	2	Binary Input with Status	1	06	129, 130	00, 01	8	Master will generate and process this variation
2	0	Binary Input Change: All Variations	1	06, 07, 08			56	Master will generate this variation
	1	Binary Input Change Without Time	1	06, 07, 08	129, 130	17, 28	8	Master will generate and process this variation
	2	Binary Input Change With Time	1	06, 07, 08	129, 130	17, 28	56	Master will generate and process this variation
	3	Binary Input Change With Relative Time	1	06, 07, 08	129, 130	17, 28	24	Master will generate and process this variation
10	0	Binary Output: All Variations	1	06			8	Master does not use this object type and will not generate a message or process this type
	1	Binary Output					1	
	2	Binary Output Status			129, 130	00, 01	8	
12	0	Control Block: All Variations					88	
	1	Control Relay Output Block	3, 4, 5, 6	17, 28	129	Echo of request	88	Master will generate this variation and parse the response
	2	Pattern Control Block					88	
	3	Pattern Mask					16	
20	0	Binary Counter: All Variations	1, 7, 8, 9, 10	06			32	Master will generate this variation
	1	32-Bit Binary Counter			129, 130	00, 01	40	Master will process this variation
	2	16-Bit Binary Counter			129, 130	00, 01	24	Master will process this variation
	3	32-Bit Delta Counter			129, 130	00, 01	40	Master will process this variation

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	4	16-Bit Delta Counter			129, 130	00, 01	24	Master will process this variation
	5	32-Bit Binary Counter Without Flag	1, 7, 8, 9, 10	06	129, 130	00, 01	32	Master will generate and process this variation
	6	16-Bit Binary Counter Without Flag	1, 7, 8, 9, 10	06	129, 130	00, 01	16	Master will generate and process this variation
	7	32-Bit Delta Counter Without Flag			129, 130	00, 01	32	Master will process this variation
	8	16-Bit Delta Counter Without Flag			129, 130	00, 01	16	Master will process this variation
21	0	Frozen Counter: All Variations	1	06			32	Master will generate this variation
	1	32-Bit Frozen Counter			129, 130	00, 01	40	Master will process this variation
	2	16-Bit Frozen Counter			129, 130	00, 01	24	Master will process this variation
	3	32-Bit Frozen Delta Counter					40	
	4	16-Bit Frozen Delta Counter					24	
	5	32-Bit Frozen Counter With Time Of Freeze					88	
	6	16-Bit Frozen Counter With Time Of Freeze					72	
	7	32-Bit Frozen Delta Counter With Time Of Freeze					88	
	8	16-Bit Frozen Delta Counter With Time Of Freeze					72	
	9	32-Bit Frozen Counter Without Flag	1	06	129, 130	00, 01	32	Master will generate and process this variation
	10	16-Bit Frozen Counter Without Flag	1	06	129, 130	00, 01	16	Master will generate and process this variation
	11	32-Bit Frozen Delta Counter Without Flag					32	
	12	16-Bit Frozen Delta Counter Without Flag					16	
22	0	Counter Change Event: All Variations	1	06, 07, 08				Master will not generate a request for this variation
	1	32-Bit Counter Change Event Without Time			129, 130	17, 28	40	Master will process this variation

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	2	16-Bit Counter Change Event Without Time			129, 130	17, 28	24	Master will process this variation
	3	32-Bit Delta Counter Change Event Without Time					40	
	4	16-Bit Delta Counter Change Event Without Time					24	
	5	32-Bit Counter Change Event With Time					88	
	6	16-Bit Counter Change Event With Time					72	
	7	32-Bit Delta Counter Change Event With Time					88	
	8	16-Bit Delta Counter Change Event With Time					72	
23	0	Frozen Counter Event: All Variations						
	1	32-Bit Frozen Counter Event Without Time					40	
	2	16-Bit Frozen Counter Event Without Time					24	
	3	32-Bit Frozen Delta Counter Event Without Time					40	
	4	16-Bit Frozen Delta Counter Event Without Time					24	
	5	32-Bit Frozen Counter Event With Time					88	
	6	16-Bit Frozen Counter Event With Time					72	
	7	32-Bit Frozen Delta Counter Event With Time					88	
	8	16-Bit Frozen Delta Counter Event With Time					72	
30	0	Analog Input: All Variations	1	06			16	Master will generate this variation

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	1	32-Bit Analog Input	1	06	129, 130	00, 01	40	Master will generate and process this variation (Data returned will be only 16 bit)
	2	16-Bit Analog Input	1	06	129, 130	00, 01	24	Master will generate and process this variation
	3	32-Bit Analog Input Without Flag	1	06	129, 130	00, 01	32	Master will generate and process this variation (Data returned will be only 16 bit)
	4	16-Bit Analog Input Without Flag	1	06	129, 130	00, 01	16	Master will generate and process this variation
31	0	Frozen Analog Input: All Variations						
	1	32-Bit Frozen Analog Input					40	
	2	16-Bit Frozen Analog Input					24	
	3	32-Bit Frozen Analog Input With Time To Freeze					88	
	4	16-Bit Frozen Analog Input With Time To Freeze					72	
	5	32-Bit Frozen Analog Input Without Flag					32	
	6	16-Bit Frozen Analog Input Without Flag					16	
32	0	Analog Change Event: All Variations	1	06, 07, 08			24	Master will generate this variation
	1	32-Bit Analog Change Event Without Time	1	06, 07, 08	129, 130	17, 28	40	Master will generate and process this variation
	2	16-Bit Analog Change Event Without Time	1	06, 07, 08	129, 130	17, 28	24	Master will generate and process this variation
	3	32-Bit Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	88	Master will generate and process this variation
	4	16-Bit Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	72	Master will generate and process this variation
33	0	Frozen Analog Event: All Variations						
	1	32-Bit Frozen Analog Event Without Time					40	
	2	16-Bit Frozen Analog Event Without Time					24	

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	3	32-Bit Frozen Analog Event With Time					88	
	4	16-Bit Frozen Analog Event With Time					72	
40	0	Analog Output Status: All Variations	1	06			24	Master does not use this object type and will not generate a message or process this type
	1	32-Bit Analog Output Status					40	
	2	16-Bit Analog Output Status			129, 130	00, 01	24	
41	0	Analog Output Block: All Variations					24	
	1	32-Bit Analog Output Block					40	
	2	16-Bit Analog Output Block	3, 4, 5, 6	17, 28	129	Echo of Request	24	Master will generate this variation and parse the response
50	0	Time and Date: All Variations					48	
	1	Time and Date	2	07, With Quant=1			48	Master will generate this variation
	2	Time and Date With Interval					80	
51	0	Time and Date CTO: All Variations						
	1	Time and Date CTO			129, 130	07, With Quant=1	48	Master will process this variation
	2	Unsynchronized Time and Date CTO			129, 130	07, With Quant=1	48	Master will process this variation
52	0	Time Delay: All Variations						
	1	Time Delay Coarse			129	07, With Quant=1	16	Master will not process this variation
	2	Time Delay Fine			129	07, With Quant=1	16	Master will not process this variation
60	0	Not Defined						Not Defined in DNP
	1	Class 0 Data	1	06				Master will generate this variation
	2	Class 1 Data	1	06, 07, 08				Master will generate this variation
	3	Class 2 Data	1	06, 07, 08				Master will generate this variation
	4	Class 3 Data	1	06, 07, 08				Master will generate this variation
70	0	Not Defined						

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	1	File Identifier						
80	0	Not Defined						
	1	Internal Indications	2	00, Index=7			24	The Master will generate this variation
81	0	Not Defined						
	1	Storage Object						
82	0	Not Defined						
	1	Device Profile						
83	0	Not Defined						Not Defined in DNP
	1	Private Registration Object						
	2	Private Registration Objection Descriptor						
90	0	Not Defined						Not Defined in DNP
	1	Application Identifier						
100	0							
	1	Short Floating Point					48	
	2	Long Floating Point					80	
	3	Extended Floating Point					88	
101	0							
	1	Small Packed Binary-Coded Decimal					16	
	2	Medium Packed Binary-Coded Decimal					32	
	3	Large Packed Binary-Coded Decimal					64	
No Object			13					Master supports the Cold Restart Function
			14					Master supports the Warm Restart Function
			20					Master supports the Enable Unsolicited Function
			21					Master supports the Disable Unsolicited Function

6 Support, Service & Warranty

In This Chapter

- How to Contact Us: Technical Support 119
- Return Material Authorization (RMA) Policies and Conditions 120
- LIMITED WARRANTY 122

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 119). 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 122)". 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 122) 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

A

Adding the Module to an Existing CompactLogix Project • 47
Adding the Module to an Existing MicroLogix Project • 50
Address In Slave • 33
All Product Returns: • 120
Allocation of Risks • 125
Application Layer Errors • 102

B

Backplane Data Transfer • 73
Battery Life Advisory • 2
Block 9901 - Event Command • 77
Block 9902 - Command Control Block • 79
Block 9903 - Event Message Block • 79
Block 9949 - Read Slave Communication Error Table • 80
Block 9950 - Read Command Error List • 81
Block 9958 - Processor Binary Input Event • 82
Block 9959 - Processor Analog Input Event • 83
Block 9970 - Set Processor Time Using Module Time • 84
Block 9971 - Set Module's Time Using the Processor's Time • 85
Block Format for Read • 78, 79, 81, 82, 83, 84, 85
Block Format for Write • 77, 79, 80, 81, 82, 83, 84, 85

C

Cable Connections • 89
Clearing a Fault Condition • 67
Clearing the Error List • 57
Cold Boot • 86
Command Control Blocks • 77
Command Error Codes • 101
Command List • 29
Configuration File • 21
Configuring RSLinx • 17
Configuring the MVI69-DNP Module • 21
Connect your PC to the Module • 19
Connect your PC to the Processor • 15
Controlling Law and Severability • 125

D

Data Flow between MVI69-DNP Module and CompactLogix Processor • 87
DB9 to RJ45 Adaptor (Cable 14) • 94
Diagnostics and Troubleshooting • 7, 51, 53
Disabling the RSLinx Driver for the Com Port on the PC • 55, 90
Disclaimer of all Other Warranties • 124
Disclaimer Regarding High Risk Activities • 123
Displaying the Current Page of Registers Again • 62
DNP Database View Menu • 58, 61
DNP DB Address • 33
DNP Port Configuration Errors • 100
DNP Subset Definition - Master • 112
DNP Subset Definition - Slave • 105
DNP V3.00 Device Profile Document • 104
DNP_AI_Event Object • 46
DNP_BI_Event Object • 46
DNPBackplane Object • 44
DNPCROB Object • 45
DNPData Object • 44
DNPEventMsg Object • 45
DNPModuleDef Object • 41
DNPSlave_Err Object • 46
DNPSlvStat Object • 42
Download the Sample Program to the Processor • 16

E

Editing the Configuration File • 22
Error Status Table • 95

F

First Byte • 103
Function • 33
Functional Overview • 7, 72
Functional Specifications • 70

G

General Command Errors • 101
General Concepts • 72
General Specifications • 69
Guide to the MVI69-DNP User Manual • 7

H

Hardware Specifications • 70
How to Contact Us
Technical Support • 119, 120

I

IED DB Address • 33
Install the Module in the Rack • 12
Intellectual Property Indemnity • 123
Internal Indication Word • 103

K

Keystrokes • 54

L

Ladder Logic • 41
LED Status Indicators • 7, 66
Limitation of Remedies ** • 124
LIMITED WARRANTY • 121, 122

M

Main Logic Loop • 72
Main Menu • 56
Master Command Error List Menu • 65

Master Command List Menu • 64
Master Driver Mode • 88
Module Data Objects • 41
Module Error Codes • 99
Module Power Up • 72
Moving Back Through 5 Pages of Registers • 62
MVI69-DNP Communication Module Configuration • 23

N

Navigation • 54
No Other Warranties • 125
Normal Data Transfer • 75

O

Object • 33
OCTET STRING POINT LIST • 88
Opening the Class Assignment Menu • 58
Opening the Command Error List Menu • 59
Opening the Command List Menu • 59
Opening the DNP Database View Menu • 58

P

Package Contents • 10
Pinouts • 89, 94
Please Read This Notice • 2
Point Count • 33
Poll Interval • 34
Procedures for Return of Units Out of Warranty: • 121
Procedures for Return of Units Under Warranty: • 121
Product Specifications • 7, 69
ProSoft® Product Documentation • 3
Purchasing Warranty Extension: • 121

R

Read Block • 76
Reading Status Data from the Module • 53
Receiving the Configuration File • 58
Redisplaying the Current Page • 60, 64, 65
Reference • 7, 69
Required Hardware • 54
Required Software • 55
Return Material Authorization (RMA) Policies and Conditions • 120
Returning to the Main Menu • 60, 64, 65
RS-232 • 91
  Modem Connection • 92
  Null Modem Connection (Hardware Handshaking) • 92
  Null Modem Connection (No Hardware Handshaking) • 93
RS-232 Configuration/Debug Port • 89
RS-422 • 93
RS-485 • 94
RS-485 and RS-422 Tip • 94

S

Second Byte: • 103
Sending the Configuration File • 58
Setting Jumpers • 11
Setting the Debug Level • 56
Skipping 500 Registers of Data • 62
Slave Address • 33
Slave Driver • 87
Slave List • 29
Slave Port Communication Errors • 99
Special Objects • 45
Start Here • 7, 9
Support, Service & Warranty • 7, 119
System Configuration Errors • 100
System Requirements • 9

T

The Class Assignment Menu • 59
The Configuration/Debug Menu • 34, 53
Time Limit for Bringing Suit • 125
Transferring the Configuration File to the Module • 36
Transferring the Configuration File to Your PC • 34
Troubleshooting • 68

U

Uploading and Downloading the Configuration File • 21, 34, 58
Using the Configuration/Debug Port • 55

V

Variation • 33
Viewing Analog Inputs • 60
Viewing Binary Inputs • 60
Viewing Block Transfer Statistics • 58
Viewing COM States • 57
Viewing Comm Stats • 57
Viewing Data in ASCII (Text) Format • 63
Viewing Data in Decimal Format • 62
Viewing Data in Double Word Decimal Format • 63
Viewing Data in Floating Point Format • 63
Viewing Data in Hexadecimal Format • 62
Viewing Data Type Databases • 61
Viewing DNP Analog Inputs • 63
Viewing DNP Analog Outputs • 63
Viewing DNP Binary Inputs • 63
Viewing DNP Binary Outputs • 63
Viewing DNP Counters • 63
Viewing DNP Float Inputs • 63
Viewing DNP Float Outputs • 63
Viewing DNP Frozen Counters • 63
Viewing DNP Set Up & Pointers • 57
Viewing Double Inputs • 60
Viewing Float Inputs • 60
Viewing Master Port Slave Setup • 58
Viewing Operating Parameters • 57
Viewing Register Pages • 62
Viewing the Error List • 57
Viewing the Next 100 Registers of Data • 62
Viewing the Next 20 Commands • 65
Viewing the Next 50 Commands • 64
Viewing the Next Page of Commands • 64, 65
Viewing the Next Page of Data • 60
Viewing the Previous 100 Registers of Data • 62
Viewing the Previous 20 Commands • 65

Viewing the Previous 50 Commands • 64
Viewing the Previous Page of Commands • 64, 65
Viewing the Previous Page of Data • 60
Viewing Version Information • 58

W

Warm Boot • 86
Warm Booting the Module • 57
What Is Covered By This Warranty • 122, 124
What Is Not Covered By This Warranty • 122
Write Block • 77

Y

Your Feedback Please • 2