![ProSoft Technology logo]

Where Automation Connects.

MVI56E-DNPNET

ControlLogix® Platform

DNP3 Ethernet

July 29, 2024

USER MANUAL

Your Feedback Please

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

How to Contact Us

ProSoft Technology, Inc.
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
www.prosoft-technology.com
support@prosoft-technology.com

MVI56E-DNPNET User Manual
For Public Use.

July 29, 2024

ProSoft Technology®, is a registered copyright of ProSoft Technology, Inc. All other brand or product names are or may be trademarks of, and are used to identify products and services of, their respective owners.

Content Disclaimer

This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither ProSoft Technology nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. Information in this document including illustrations, specifications and dimensions may contain technical inaccuracies or typographical errors. ProSoft Technology makes no warranty or representation as to its accuracy and assumes no liability for and reserves the right to correct such inaccuracies or errors at any time without notice. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of ProSoft Technology. All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components. When devices are used for applications with technical safety requirements, the relevant instructions must be followed. Failure to use ProSoft Technology software or approved software with our hardware products may result in injury, harm, or improper operating results. Failure to observe this information can result in injury or equipment damage.

Copyright © 2024 ProSoft Technology, Inc. All Rights Reserved.

For professional users in the European Union
If you wish to discard electrical and electronic equipment (EEE), please contact your dealer or supplier for further information.

Warning – Cancer and Reproductive Harm – www.P65Warnings.ca.gov

Open Source Information

Open Source Software used in the product

The product contains, among other things, Open Source Software files, as defined below, developed by third parties and licensed under an Open Source Software license. These Open Source Software files are protected by copyright. Your right to use the Open Source Software is governed by the relevant applicable Open Source Software license conditions. Your compliance with those license conditions will entitle you to use the Open Source Software as foreseen in the relevant license. In the event of conflicts between other ProSoft Technology, Inc. license conditions applicable to the product and the Open Source Software license conditions, the Open Source Software conditions shall prevail. The Open Source Software is provided royalty-free (i.e. no fees are charged for exercising the licensed rights). Open Source Software contained in this product and the respective Open Source Software licenses are stated in the module webpage, in the link Open Source.

If Open Source Software contained in this product is licensed under GNU General Public License (GPL), GNU Lesser General Public License (LGPL), Mozilla Public License (MPL) or any other Open Source Software license, which requires that source code is to be made available and such source code is not already delivered together with the product, you can order the corresponding source code of the Open Source Software from ProSoft Technology, Inc. - against payment of the shipping and handling charges - for a period of at least 3 years since purchase of the product. Please send your specific request, within 3 years of the purchase date of this product, together with the name and serial number of the product found on the product label to:

ProSoft Technology, Inc.
Director of Engineering
9201 Camino Media, Suite 200
Bakersfield, CA 93311
USA

Warranty regarding further use of the Open Source Software

ProSoft Technology, Inc. provides no warranty for the Open Source Software contained in this product, if such Open Source Software is used in any manner other than intended by ProSoft Technology, Inc. The licenses listed below define the warranty, if any, from the authors or licensors of the Open Source Software. ProSoft Technology, Inc. specifically disclaims any warranty for defects caused by altering any Open Source Software or the product's configuration. Any warranty claims against ProSoft Technology, Inc. in the event that the Open Source Software contained in this product infringes the intellectual property rights of a third party are excluded. The following disclaimer applies to the GPL and LGPL components in relation to the rights holders:

"This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License and the GNU Lesser General Public License for more details."

For the remaining open source components, the liability exclusions of the rights holders in the respective license texts apply. Technical support, if any, will only be provided for unmodified software.

Important Safety Information

North America Warnings

A   This Equipment is Suitable For Use in Class I, Division 2, Groups A, B, C, D or Non-Hazardous Locations Only.

B   Warning – Explosion Hazard – Substitution of Any Components May Impair Suitability for Class I, Division 2.

C   Warning – Explosion Hazard – Do Not Disconnect Equipment Unless Power Has Been Switched Off Or The Area is Known To Be Non-Hazardous.

D   The subject devices are powered by a Switch Model Power Supply (SMPS) that has regulated output voltage of 5 VDC.

ATEX/IECEx Warnings and Conditions of Safe Usage:

Power, Input, and Output (I/O) wiring must be in accordance with the authority having jurisdiction.

A   Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or wiring modules.

B   Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.

C   These products are intended to be mounted in an ATEX/IECEx Certified, tool-secured, IP54 enclosure. The devices shall provide external means to prevent the rated voltage being exceeded by transient disturbances of more than 40%. This device must be used only with ATEX certified backplanes.

D   Before operating the reset switch, be sure the area is known to be non-hazardous.

E   If the equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

Agency Approvals and Certifications

Please visit our website: www.prosoft-technology.com

China RoHS Hazardous Material Declaration Table

产品中有害物质的名称及含量
Name and content of hazardous substances in product

部件名称 Component Name	有害物质					
	铅 Lead (Pb)	汞 Mercury (Hg)	镉 Cadmium (Cd)	六价铬 Hexavalent Chromium (Cr(VI))	多溴联苯 Polybrominated Biphenyls (PBB)	多溴二苯醚 Polybrominated Diphenyl Ethers (PBDE)
印刷电路板组件 Printed Circuit Board Assemblies	X	O	O	O	O	O
金属部件 Metal Components	X	O	O	O	O	O
电池 Battery	O	O	O	O	O	O
塑料部件 Plastic Components	O	O	O	O	O	O

本表格依据SJ/T 11364的规定编制。This table is made per guidance of SJ/T 11364
X: 表示该有害物质至少在该部件的某一均质材料中的含量超出GB/T 26572规定的限量要求。
（企业可在此处，根据实际情况对上表中打"X"的技术原因进行进一步说明。）

Contents

1 Start Here

To get the most benefit from this User Manual, you should have the following skills:

- **Rockwell Automation® RSLogix™ software:** launch the program, configure ladder logic, and transfer the ladder logic to the processor
- **Microsoft Windows®:** install and launch programs, execute menu commands, navigate dialog boxes, and enter data
- **Hardware installation and wiring:** install the module, and safely connect DNP3 Ethernet and ControlLogix devices to a power source and to the MVI56E-DNPNET's application port(s)

1.1 System Requirements

The MVI56E-DNPNET module requires the following minimum hardware and software components:

- Rockwell Automation® ControlLogix® processor (firmware version 10 or higher) with compatible limited voltage power supply and one free slot in the rack for the MVI56E-DNPNET module. The module requires 800 mA of available 5 VDC and 3 mA of available 24 VDC power.

- Rockwell Automation RSLogix™ 5000 programming software
  - o Version 16 or higher required for Add-On Instruction
- Rockwell Automation RSLinx® communication software version 2.51 or higher
- ProSoft Configuration Builder (PCB)
- ProSoft Discovery Service (PDS)

Note: The Hardware and Operating System requirements in this list are the minimum recommended to install and run software provided by ProSoft Technology®. Other third party applications may have different minimum requirements. Refer to the documentation for any third party applications for system requirements.

Note: You can install the module in a local or remote rack. For remote rack installation, the module requires EtherNet/IP or ControlNet communication with the processor.

1.2 Package Contents

Qty.	Part Name	Part Number	Part Description
1	MVI56E-DNPNET	MVI56E-DNPNET	DNPNET Ethernet Client/Server Communication Module

1.3 Setting Jumpers

The Setup Jumper acts as "write protection" for the module's firmware. In "write protected" mode, the Setup pins are not connected, and the module's firmware cannot be overwritten. The module is shipped with the Setup jumper OFF. Do not jumper the Setup pins together unless you are directed to do so by ProSoft Technical Support (or you want to update the module firmware).

The following illustration shows the MVI56E-DNPNET jumper configuration with the Setup Jumper OFF.
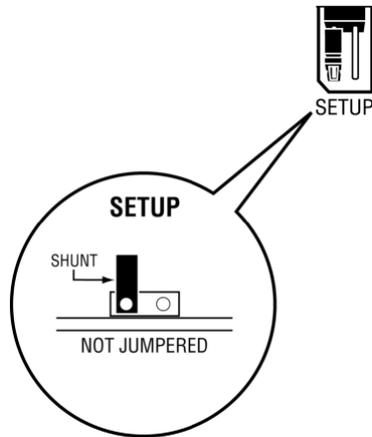
Note: If you are installing the module in a remote rack, you may prefer to leave the Setup pins jumpered. That way, you can update the module's firmware without requiring physical access to the module.

1.4   Installing the Module in the Rack

Make sure your ControlLogix processor and power supply are installed and configured, before installing the MVI56E-DNPNET. Refer to your Rockwell Automation product documentation for installation instructions.

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

After you have checked the placement of the jumpers, insert the MVI56E-DNPNET into the ControlLogix chassis. Use the same technique recommended by Rockwell Automation to remove and install ControlLogix modules.

You can install or remove ControlLogix system components while chassis power is applied and the system is operating. However, please note the following warning.

Warning: When you insert or remove the module while backplane power is on, an electrical arc can occur. An electrical arc can cause personal injury or property damage by sending an erroneous signal to the system's actuators. This can cause unintended machine motion or loss of process control. Electrical arcs may also cause an explosion when they happen in a hazardous environment. Verify that power is removed or the area is non-hazardous before proceeding.

Repeated electrical arcing causes excessive wear to contacts on both the module and its mating connector. Worn contacts may create electrical resistance that can affect module operation.

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

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

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

Note: When using the XT version (if applicable), you must use the 1756-A5XT or 1756-A7LXT chassis to uphold the XT specifications. In these chassis, modules are spaced further apart than in standard ControlLogix chassis. Blank spacers are inserted between active modules.

1.5 Creating a New RSLogix 5000 Project

1 Open the **FILE** menu, and then choose **NEW**.
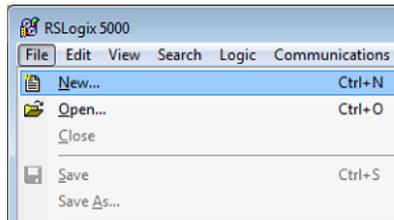
2 Select your ControlLogix controller model.
3 Select the **REVISION** of your controller. Depending on the revision, there may be some small differences in the appearance of dialog boxes from the ones shown.
4 Enter a name for your controller, such as *My_Controller*.
5 Select your ControlLogix chassis type.
6 Select **SLOT 0** for the controller.
7 Click **OK**.
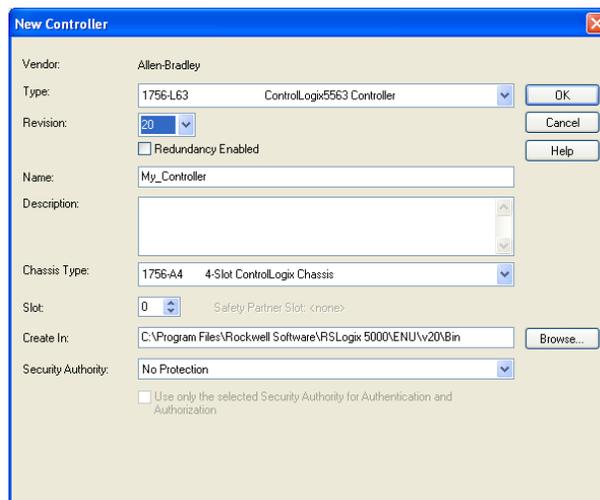
1.5.1 Before You Import the Add-On Instruction

One Add-On Instruction is provided for the MVI56E-DNPNET module. It is required for setting up the module.

Download the files from www.prosoft-technology.com. Save them to a convenient location in your PC, such as *Desktop* or *My Documents*.

File Name	Description
Example: MVI56(E)DNPNET_AddOn_Rung_vX.L5X	L5X file containing Add-On Instruction, user defined data types, controller tags and ladder logic required to configure the MVI56E-DNPNET module

1.5.2 Creating the Module

Add the MVI56E-DNPNET module to the project.

1 In the *Controller Organizer* window, right-click **I/O CONFIGURATION** or the backplane and then choose **NEW MODULE...**
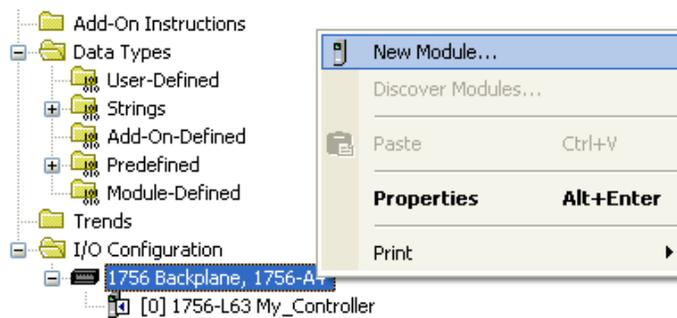
This action opens the **SELECT MODULE** dialog box. Enter *generic* in the text box and select the **GENERIC 1756 MODULE**. If you're using a controller revision of 15 or less, expand **OTHER** in the **SELECT MODULE** dialog box, and then select the **GENERIC 1756 MODULE**.
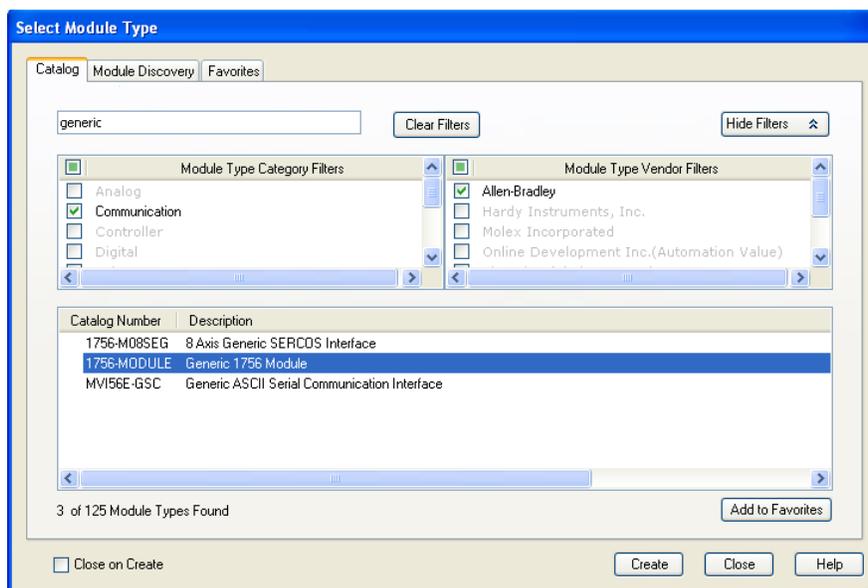
2 Click **CREATE.** This action opens the **NEW MODULE** dialog box.
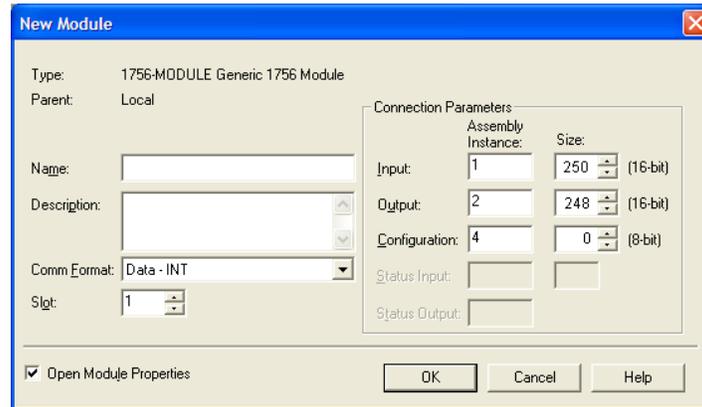
3 In the **NEW MODULE** dialog box, enter the following values.

Parameter	Value
NAME	DNPNET
DESCRIPTION	Enter a description for the module. Example: DNPNET Ethernet Client/Server Communication Module
COMM FORMAT	Select **DATA-INT**
SLOT	Enter the slot number in the rack where the MVI56E-DNPNET module is located
INPUT ASSEMBLY INSTANCE	1
INPUT SIZE	250
OUTPUT ASSEMBLY INSTANCE	2
OUTPUT SIZE	248
CONFIGURATION ASSEMBLY INSTANCE	4
CONFIGURATION SIZE	0

4 Click **OK** to continue.
5 Edit the *Module Properties*.
6 Select the **REQUESTED PACKET INTERVA**l value for scanning the I/O on the module. This value represents the minimum frequency at which the module will handle scheduled events. This value should not be set to less than 1 millisecond. The default value is 5 milliseconds. Values between 1 and 10 milliseconds should work with most applications.

---
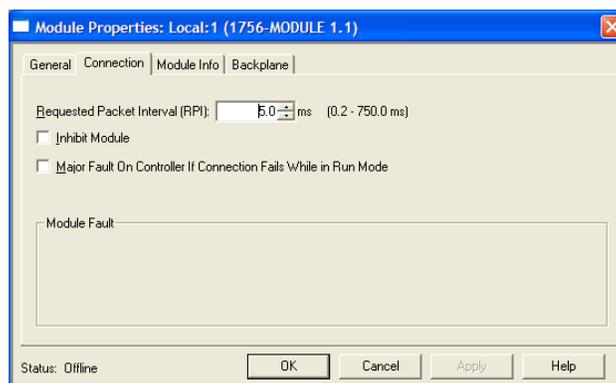
7   Click **OK** to save the module and close the dialog box. Notice that the module now appears in the *Controller Organizer* window.
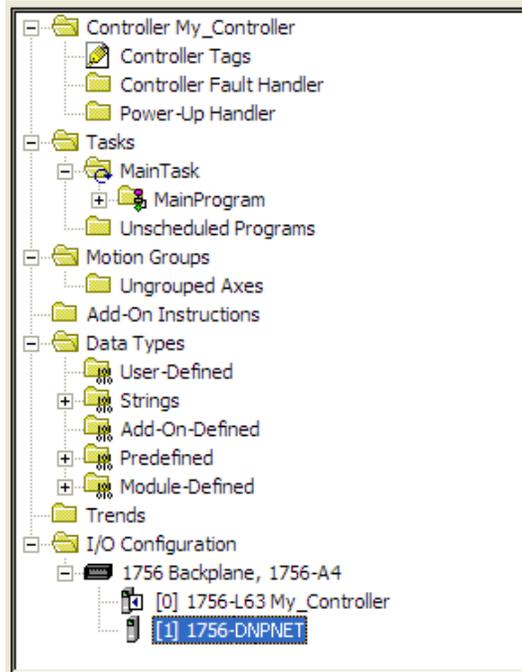
1.5.3 Importing the Add-On Instruction

1   In the **CONTROLLER ORGANIZATION** window, expand the **TASKS** folder and subfolders until you reach the **MAINPROGRAM** folder.
2   In the **MAINPROGRAM** folder, double-click to open the **MAINROUTINE** ladder.
3   Select an empty rung in the routine, and then click the right mouse button to open a shortcut menu. On the shortcut menu, choose **IMPORT RUNGS…**
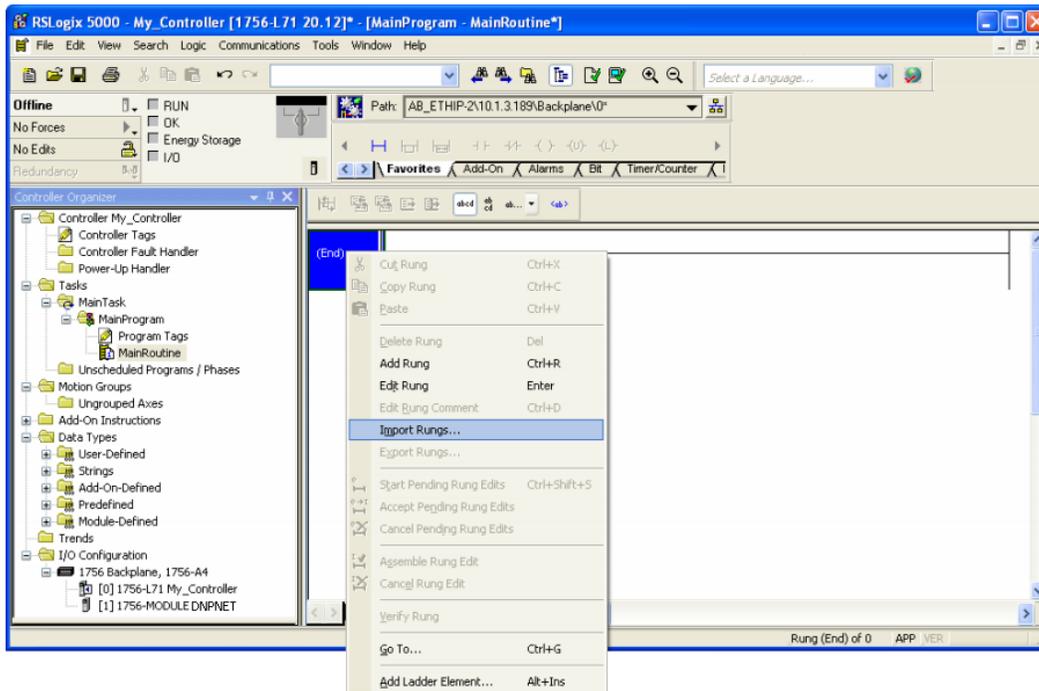
4   Navigate to the location on your PC where you saved the Add-On Instruction (for example, *My Documents* or *Desktop*). Select the **MVI56(E)DNPNET_ADDON_RUNG_VX.L5X** file.
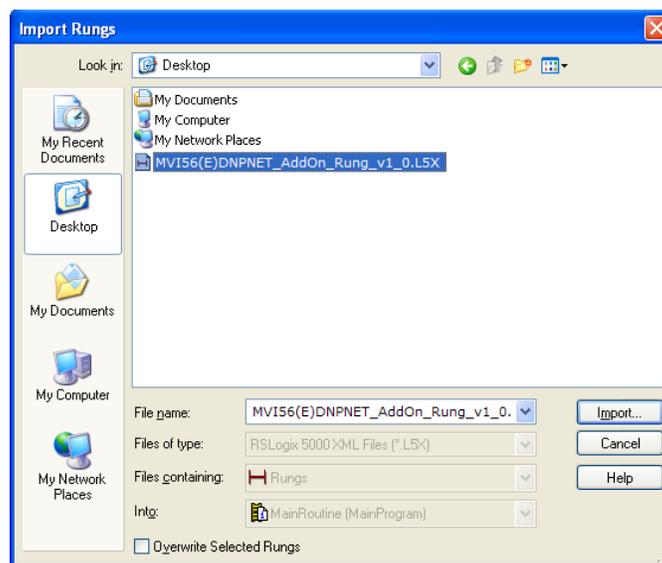
This action opens the **IMPORT CONFIGURATION** dialog box. Click **TAGS** under **MAINROUTINE** to show the controller tags that will be created.

Note that if you are using a controller revision number of 16 or less, the **IMPORT CONFIGURATION** dialog box does not show the **IMPORT CONTENT** tree.

Note that if you are using a controller revision number of 16 or less, the **IMPORT CONFIGURATION** dialog box does not show the **IMPORT CONTENT** tree.
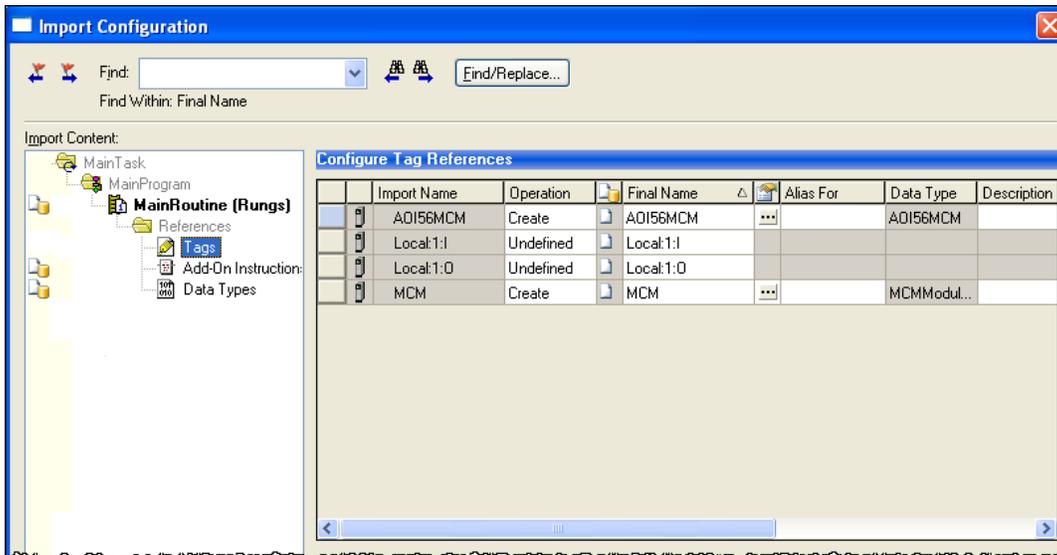
5 If you are using the module in a different slot (or remote rack), edit the connection input and output variables that define the path to the module. Edit the text in the **FINAL NAME** column **(NAME** column for controller revision 16 or less). For example, if your module is located in slot 3, change Local:1:I in the above picture to Local:3:I. Do the same for Local:1:O. If your module is located in Slot 1 of the local rack, this step is not required.

6 Click **OK** to confirm the import. RSLogix 5000 indicates that the import is in progress:
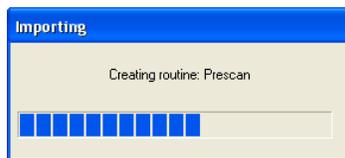
When the import is completed, the new rung with the Add-On Instruction will be visible as shown in the following illustration.
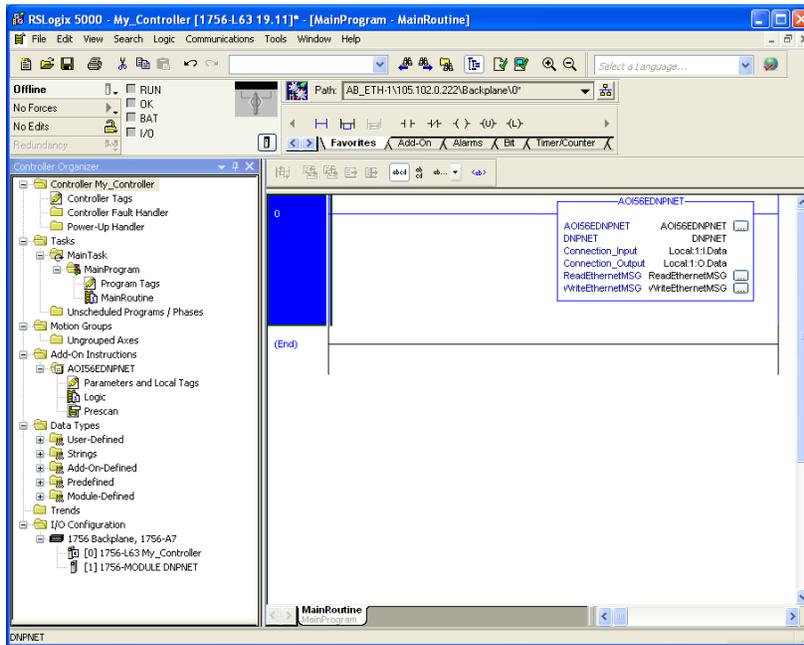
The procedure also imports new User Defined Data Types, Controller Tags, and the Add-On instruction for your project.

7 Save the application and then download the sample ladder logic into the processor.

1.5.4 Adding Multiple Modules (Optional)

Important: If your application requires more than one MVI56E-DNPNET module in the same project, follow the steps below.

1   In the **I/O CONFIGURATION** folder, click the right mouse button to open a shortcut menu, and then choose **NEW MODULE**.
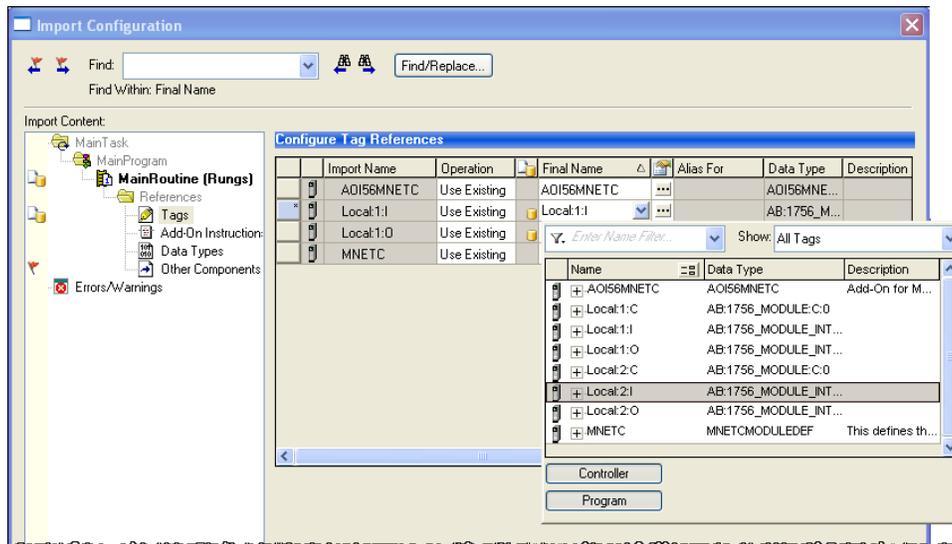
2   Select **1756-MODULE**. If you're using a controller revision of 16 or less, expand **OTHER** in the **SELECT MODULE** dialog box, and then select the **1756-MODULE**.
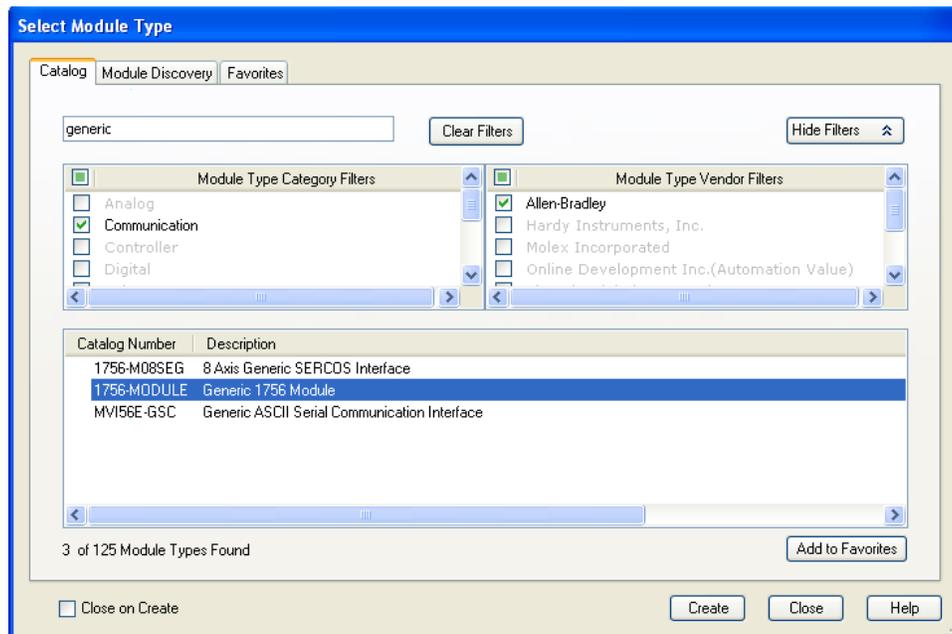
3 Fill the module properties as follows:

Parameter	Value
NAME	Enter a module identification string. Example: DNPNET_2.
DESCRIPTION	Enter a description for the module. Example: DNPNET Ethernet Client/Server Communication Module.
COMM FORMAT	Select DATA-INT.
SLOT	Enter the slot number in the rack where the MVI56E-DNPNET module is located.
INPUT ASSEMBLY INSTANCE	1
INPUT SIZE	250
OUTPUT ASSEMBLY INSTANCE	2
OUTPUT SIZE	248
CONFIGURATION ASSEMBLY INSTANCE	4
CONFIGURATION SIZE	0

4 Click **OK** to confirm. The new module is now visible:
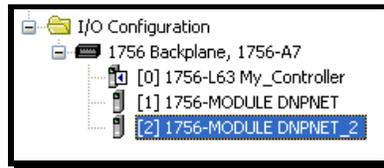
5 Expand the **TASKS** folder, and then expand the **MAINTASK** folder.
6 In the **MAINPROGRAM** folder, double-click to open the **MAINROUTINE** ladder.
7 Select an empty rung in the routine, and then click the right mouse button to open a shortcut menu. On the shortcut menu, choose **IMPORT RUNGS…**
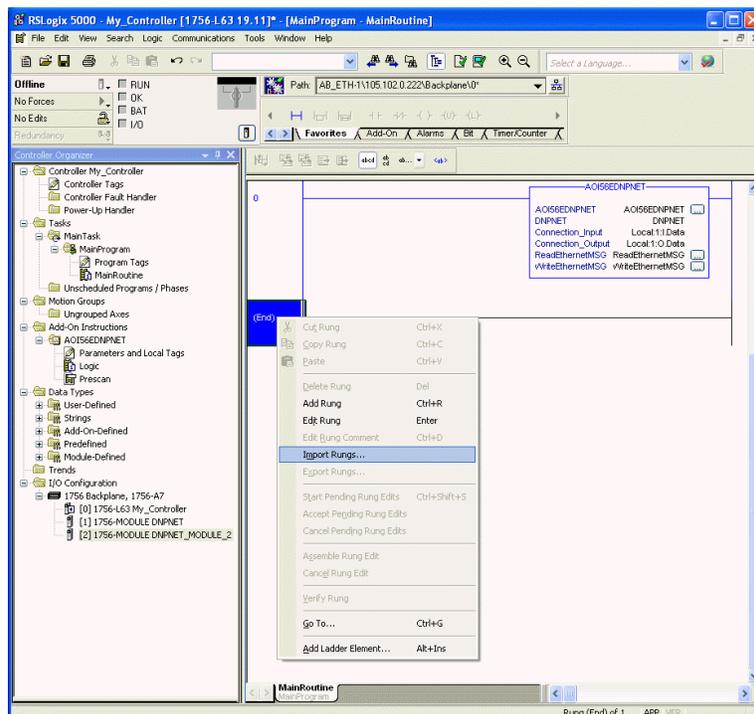
8   Select the **MVI56(E)DNPNET_ADDON_RUNG_VX.L5X** file, and then click **IMPORT.**
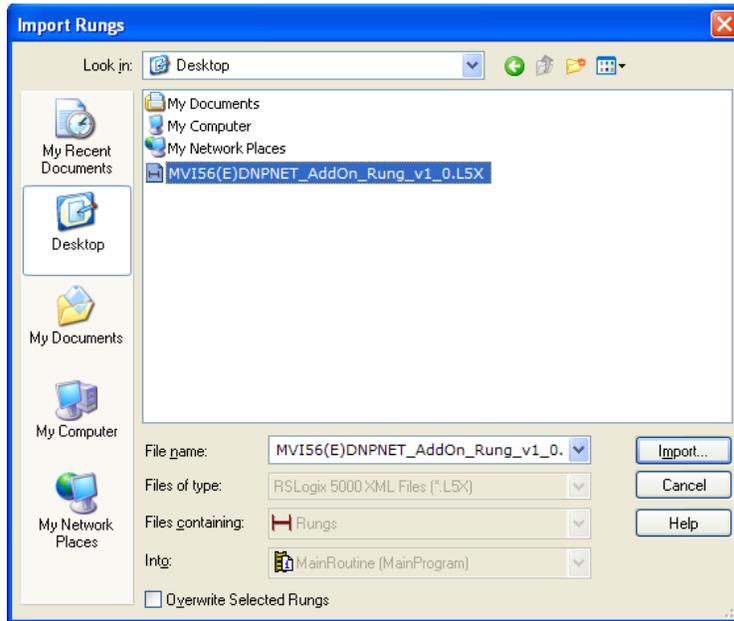
9   This action opens the **IMPORT CONFIGURATION** window. Click **TAGS** under **MAINROUTINE** to show the tags that will be imported.
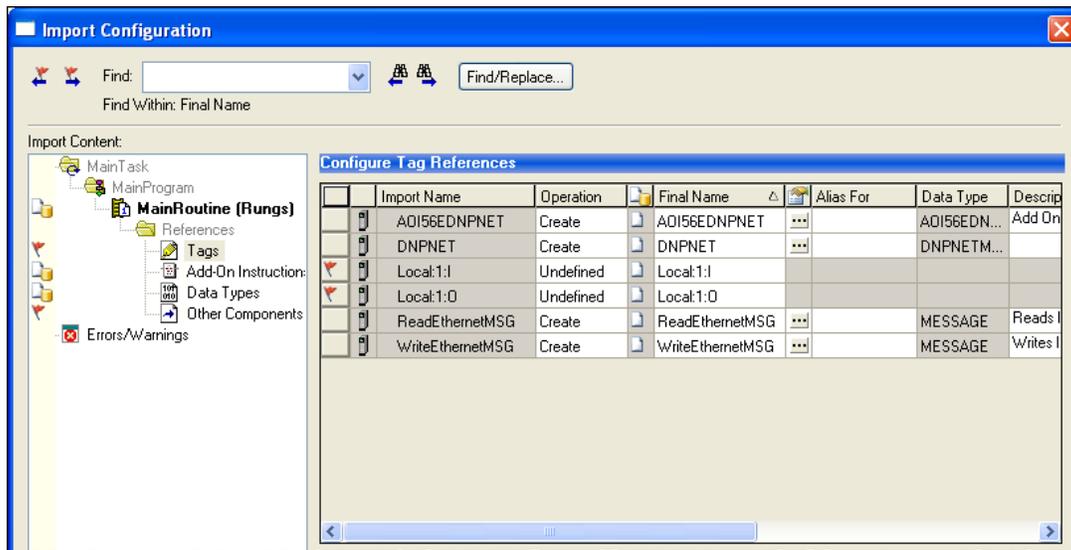
10 Associate the I/O connection variables to the correct module in the corresponding slot number. The default values are Local:1:I and Local:1:O and must be edited if the card is placed in a slot location other than slot 1 (Local:1:x means the card is located in slot 1). Since the second card is placed in slot 2, edit the **FINAL NAME** to Local:2:I and Local:2:O.

11 Also, append '2' at the end of the default tags **DNPNET**, **AOI56DNPNET,** **READETHERNETMSG**, and **WRITEETHERNETMSG** to avoid conflict with existing tags as shown below..
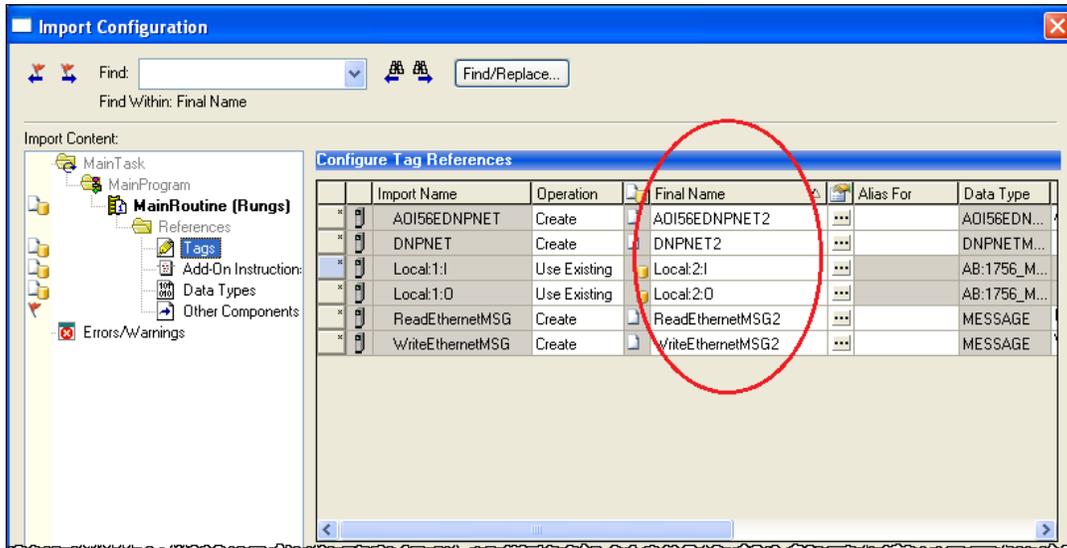
12 Click **OK** to confirm.
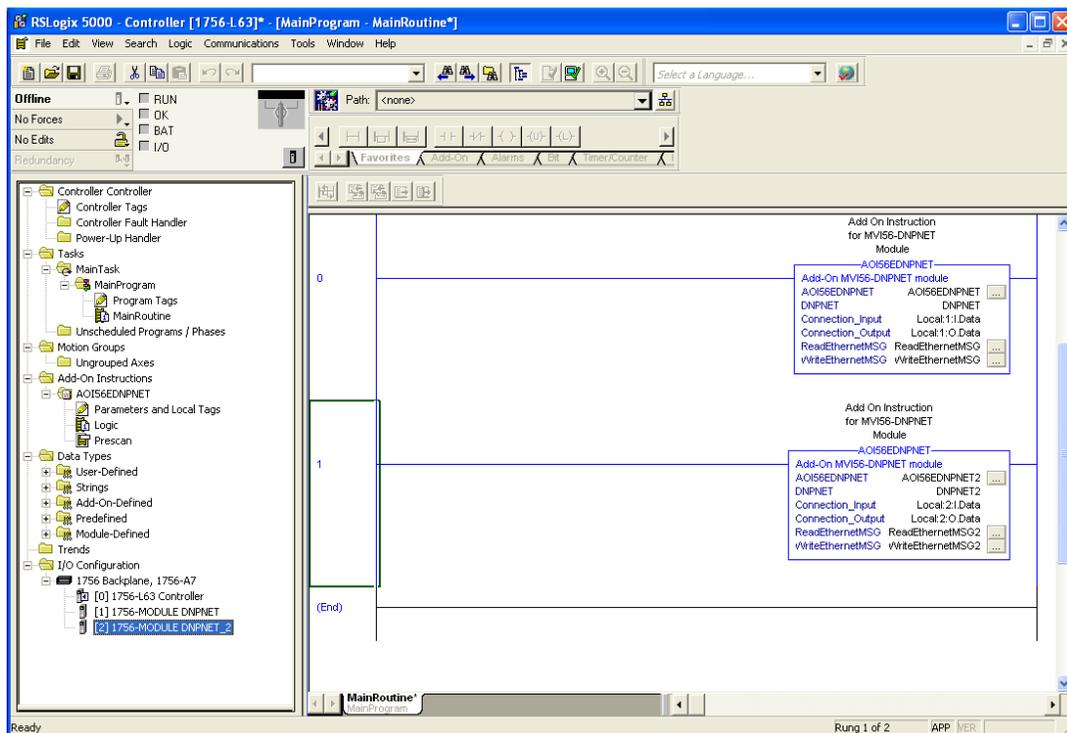
The setup procedure is now complete. Save the project and download the application to your ControlLogix processor.

1.6 Downloading the Sample Program to the Processor

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

1 If you are not already online with the processor, in RSLogix 5000 open the *Communications* menu, and then choose **DOWNLOAD**. RSLogix 5000 will establish communication with the processor. You do not have to download through the processor's serial port, as shown here. You may download through any available network connection.

2 When communication is established, RSLogix 5000 opens a confirmation dialog box. Click the **DOWNLOAD** button to transfer the sample program to the processor.

3 RSLogix 5000 will compile the program and transfer it to the processor. This process may take a few minutes.

4 When the download is complete, RSLogix 5000 will open another confirmation dialog box. If the key switch is in the REM position, click **OK** to switch the processor from PROGRAM mode to RUN mode.
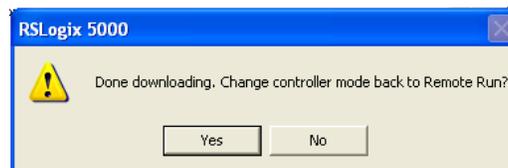
Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

2   MVI56E-DNPNET Configuration

The DNP3 Ethernet configuration resides in the DNPNET controller tags. This section covers the tag structure and descriptions of the DNPNET Client and Server parameters.

2.1   Assigning a Permanent IP Address

The module's IP address, subnet mask, and gateway parameters are configured in the DNPNET.CONFIG.IP_Settings controller tags only.

Any changes to the DNPNET.CONFIG.IP_Settings.IP, Netmask, or Gateway tags require the trigger of the DNPNET.CONFIG.IP_Settings.Write tag. This sends the parameters to the module.

You can also read the current IP settings from the module by triggering the DNPNET.CONFIG.IP_Settings.Read tag. They will populate in the DNPNET.CONFIG.IP_Settings.IP, Netmask, and Gateway tags.

2.2   DNPNET Module User-Defined Data Types

The sample ladder logic relies heavily on the use of User-Defined Data Types (UDTs) to help group and structure the wide variety and volume of data and control features the module offers. Lower-order UDT structures are often embedded in higher-order structures to help further organize data into more easily understood data collections.

All data and control parameters related to the MVI56E-DNPNET are contained in User-defined Data Types (UDTs). The *DNPNETMODULEDEF* UDT is the primary, top level data structure in which all other lower-order data types are grouped and organized. All groups branch down from this UDT.

To utilize all the features and functions of the module, an instance of each data type is required. This is accomplished by declaring controller tag variables using these data types in the Controller Tags Edit Tags dialog box.

Some UDTs hold process or status data (*Module Data Objects*). This data can be monitored and manipulated by the application-specific ladder logic program. Other UDTs are used to store and organize the parameters needed for special functions and control features (*Special Data Objects*). These data types will be discussed in more detail in succeeding topics.

2.3   DNPNET Controller Tags Definitions

2.3.1 DNPNET Controller Tag Overview

Name	Description
DNPNET.CONFIG	Configuration information
DNPNET.DATA	DNPNET input and output data transferred between the processor and the module
DNPNET.STATUS	Status information
DNPNET.CONTROL	Governs the data movement between the PLC rack and the module
DNPNET.UTIL	Generic tags used for internal ladder processing (DO NOT MODIFY)

```
─ DNPNET
    ─ DNPNET.CONFIG
        + DNPNET.CONFIG.DNP_Module_Name
        + DNPNET.CONFIG.DNP3_Server
        + DNPNET.CONFIG.DNP3_WhiteList
        + DNPNET.CONFIG.DNP3_Client
        + DNPNET.CONFIG.DNP_Server_Override
        + DNPNET.CONFIG.DNP_Server_List
        + DNPNET.CONFIG.DNP_Client_Commands
    + DNPNET.CONFIG.IP_Settings
```
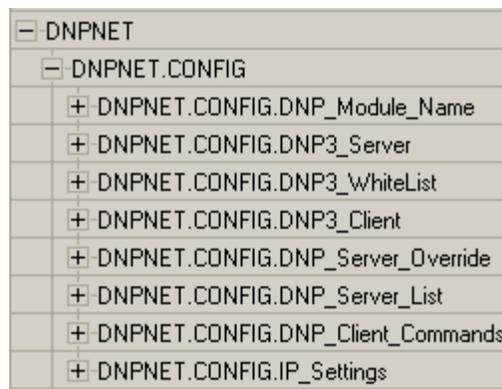
2.3.2 DNPNET.CONFIG.DNP_Module_Name

Configures the name of the MVI56E-DNPNET.

Tag Name	Range	Description
DNPNET.Config.DNP _Module_Name[0] to [39]	0 or 32 to 126	String of ASCII characters (up to 40) that gives the module a unique name. Terminate the string with a byte = 0. Module is named "MVI56E-DNPNET" by default.

2.3.3 DNPNET.CONFIG.DNP3_Server

This array configures the MVI56E-DNPNET server.

DNPNET.Config.DNP3_Server.	Range	Description
Internal_Server_ID	0 to 32767	This is the DNP address for the module. All messages with this address from the client will be processed by the module.
Use_WhiteList	0 or 1	This parameter specifies if the IP address of the host connected to the system will be validated. If the parameter is set to 0, any host may connect to the unit. If the parameter is set to 1, only hosts in the IP list will be permitted to connect to the module. All other IP addresses will be ignored by the module and the module will issue a RST to the TCP/IP connection. The IP_List is contained in DNP.Config.DNP_ENET_IP_Addresses.
Binary_Input_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.
Analog16_Input_Class	0 to 3	This parameter specifies the default class to be utilized for all the 16-bit analog input points in the DNP database that are not defined in the override list section.
Analog32_Input_Class	0 to 3	This parameter specifies the default class to be utilized for all the 32-bit 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 float input points in the DNP database that are not defined in the override list section.
Double_Class	0 to 3	This parameter specifies the default class to be utilized for all the double input points in the DNP database that are not defined in the override list section.
Analog16_Input_Deadband	0 to 32767	This parameter specifies the default deadband value assigned to all points not defined in the override list for the 16-bit analog input point type in the DNP database.
Analog32_Input_Deadband	0 to 2,147,483,647	This parameter specifies the default deadband value assigned to all points not defined in the override list for the 32-bit 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 float input point type in the DNP database.
Double_Deadband	0 to maximum double value	This parameter specifies the default deadband value assigned to all points not defined in the override list for the double input point type in the DNP database.
SelectOperate_Arm_Time	1 to 32767	Time period, in milliseconds, 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.

DNPNET.Config.DNP3_Server.	Range	Description
Write_Time_Interval	0 to 1440	Time interval, in minutes, to set the need time IIN bit (0=never), which will cause the client to write the time.
Data_Link_Confirm_Mode	0,1, or 2 (Coded Value)	IED can request acknowledgement from client station when sending data. The codes are as follows: 0=Never, 1=Sometimes, 2=Always. Recommended value = 0.
Data_Link_Confirm_Tout	1 to 32767	Time period to wait for client 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. Recommended value = 1000.
Data_Link_Max_Retry	0 to 255	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. Recommended value = 2.
App_Layer_Confirm_Tout	1 to 32767	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	0 or 1	If set to 0, the server will not send unsolicited responses. If set to 1, the server will send unsolicited responses.
Class_1_Unsol_Resp_Min	1 to 255	Minimum number of events in Class 1 required before an unsolicited response will be generated.
Class_2_Unsol_Resp_Min	1 to 255	Minimum number of events in Class 2 required before an unsolicited response will be generated.
Class_3_Unsol_Resp_Min	1 to 255	Minimum number of events in Class 3 required before an unsolicited response will be generated.
Unsol_Resp_Delay	1 to 32767	Maximum number of milliseconds to wait after an event occurs before sending an unsolicited response message. If set to 0, only use minimum number of events.
UResp_Client_Address	0 to 255	DNP destination address where unsolicited response messages are sent.
AnalogInput_Events_with_time	0 or 1	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 0, the default is set to no time data. If the parameter is set to 1, the default object will include the time of the event.
Events_Require_Time_Sync	0 or 1	This parameter is used to determine if events will be generated by the server module when its time is not synchronized from a client. If the parameter is set to 1, no events will be generated until the module's time has been synchronized. If the parameter is set to 0, events will always be generated.
Initialize_DNP_Output_Database	0 or 1	This parameter determines if the module will request data from the processor to initialize the DNP database output data areas.

DNPNET.Config.DNP3_Server.	Range	Description
PassThrough_CROB	0 or 1	This parameter determines if the module will pass all received CROB messages received through to the processor. If it is set to 0 (default), then the messages will not be sent to the processor. If the parameter is set to 1, then block 9910 will be sent to the processor with the CROB information. The database will still be controlled by the CROB message, but the ladder can control other virtual BO data in the processor using this data. This feature is useful if the controlling station sends CROB data to the server driver with very short on or off times.
Use_TripClose_Single_Point	0 or 1	This parameter determines if data associated with CROB commands operate on a single or dual point. If the value of 0 is supplied (default value), then all points will be dual-point unless neither the trip or close bit is set in the control code of the command. If either bit is set, then the CROB block will interact with the bit database as a dual-point database. If the parameter is set to 1, then all CROB blocks received will operate on the database as single bits.
Unsol_Retry_Limit	7 to 32768	Configurable unsolicited retry limit. The module sends an unsolicited message and waits for a confirmation with the Application Layer Confirm Timeout up to the limit specified until the unsolicited message is confirmed. If the amount of unsolicited messages are exceeded, the Ethernet connection will be lost. Another DNP message could wake up the connection. The allowable limits are 7 to 32768.
Use_SOE_card	0 or 1	Using Allen Bradley 1756-SOE Sequence of Events Module. 0 = No, 1 = Yes
Use_Data_from_Client_Connection	0 or 1	This parameter enables data from the MVI client connection to be shared with the MVI server database. 0 = No, 1 = Yes
Use_Double_Floats	0 or 1	This parameter enables the DNP Double Float (64bit) database. 0 = Disable, 1 = Enable
Block_Timeout_MS	11 to 5000 milliseconds	Backplane block heartbeat. If an out-of-range value is entered, the value will default to **1500**. For v1.30.063 and higher, the *Quality Flag Control* feature can be used. Please see *2.3.10 DNP3 Quality Flag Control* for more information. **Note:** If the *Block_Timeout_MS* < 10000, the *Quality Flags* are **enabled**. The module will report Quality Flags, along with DNP data. If the *Block_Timeout_MS* ≥ 10000, the *Quality Flags* are **disabled**. The module will force the Quality Flags to report only a status of **Online**, along with DNP data. The *Block_Timeout_MS* value and *Quality Flag Control* status can be checked at *PCB Diagnostics > DNP3 Server > Config*.

DNPNET.Config.DNP3_Server.	Range	Description
Server_Timeout	5 to 32767 seconds	Configurable TCP Server Connection timeout parameter. After the initial TCP connections has been made and there is no activity on the connection, the server will timeout and close the connection. If an out of range value is entered, the value will default to 5 seconds.
Reserved_0		Reserved

2.3.4 DNPNET.CONFIG.DNP3_WhiteList[x]

To avoid unknown or outside devices from trying to connect to the MVI56E-DNPNET, you can create an exclusive list of IP addresses allowed to access the device. There are a maximum of 10 IP addresses that can be configured. This tag array is used if the *DNPNET.CONFIG.DNP3_Server.Use_WhiteList* tag is set to **1**.

Tag Name	Range	Description
DNPNET.CONFIG.DNP3_WhiteList[0].a	1 to 254	IP Address Octet A
DNPNET.CONFIG.DNP3_WhiteList[0].b	0 to 254	IP Address Octet B
DNPNET.CONFIG.DNP3_WhiteList[0].c	0 to 254	IP Address Octet C
DNPNET.CONFIG.DNP3_WhiteList[0].d	1 to 254	IP Address Octet D
DNPNET.CONFIG.DNP3_WhiteList[1].a		
DNPNET.CONFIG.DNP3_WhiteList[1].b		
DNPNET.CONFIG.DNP3_WhiteList[1].c		
DNPNET.CONFIG.DNP3_WhiteList[1].d		
…	…	…
DNPNET.CONFIG.DNP3_WhiteList[9].a		
DNPNET.CONFIG.DNP3_WhiteList[9].b		
DNPNET.CONFIG.DNP3_WhiteList[9].c		
DNPNET.CONFIG.DNP3_WhiteList[9].d		

2.3.5 DNPNET.CONFIG.DNP3_Client

This array configures the MVI56E-DNPNET client.

DNPNET.CONFIG.DNP3_Client.	Range	Description
Internal_ID	0 to 32767	This is the DNP address for the module. All messages with this address from the client will be processed by the module.
Event_Messages_to_PLC	0 or 1	This parameter determines if event messages received on the client port will be sent to the processor. If this option is utilized, ladder logic must be written to handle the 9903 blocks generated by the module.
Initialize_IED_Input_Database	0 or 1	This parameter determines if the module will request data from the processor to initialize the IED database input data areas.
Only_Time_Sync_Servers_If_Synced	0 or 1	This parameter determines if the client will send a time sync message to servers when its own time has not yet been synced via PLC or a connected client. If set to 0, the client will send a time sync to connected servers even if its own clock has not been synced by the PLC or another connected client. If set to 1, the client will not send time syncs to servers until it has been synced by the PLC or a connected client.
Use_Binary_Output_status_Data	0 or 1	Enable database and functions for reading the status of Binary Outputs (object 10).
Use_Analog_Output_status_Data	0 or 1	Enable database and functions for reading the status of Analog Outputs (object 40).
Dont_Process_IIN	0 or 1	By default (0), the Client module will automatically make requests to servers that have IIN bits set for class data, need time, restart, or buffer overflow until these IIN bits have been cleared. If this parameter is set to 1, the Client will NOT automatically make these requests as a result of these IIN bits being set.

2.3.6 DNPNET.CONFIG.DNP_Server_Override

This array assigns Class and Deadband overrides to individual point types. You can adjust the size of each of the 5 point type arrays.

DNPNET.CONFIG.DNP_ Server_Override.	Range	Description
Binary_Inputs.	n/a	Number of binary input words contained in the IED database to be transferred to the PLC and obtained from the attached IED units. This array size ranges from 0 to 1000.
Point_Number	0 to Number of configured points	Point number index to be overridden
Class	1 to 3	Class number assigned to point number specified above
Analog16_Inputs.	n/a	Number of 16-bit analog input points contained in the IED database to be transferred to the PLC and obtained from the attached IED units. This array size ranges from 0 to 1000.
Point_Number	0 to Number of configured points	Point number index to be overridden
Class	1 to 3	Class number assigned to point number specified
Deadband	0 to 32767	Deadband assigned to point number specified
Analog32_Inputs.	n/a	Number of 32-bit analog input points contained in the IED database to be transferred to the PLC and obtained from the attached IED units. This array size ranges from 0 to 500.
Point_Number	0 to Number of configured points	Point number index to be overridden
Class	1 to 3	Class number assigned to point number specified
Deadband	0 to 32767	Deadband assigned to point number specified
Float_Inputs.	n/a	Number of float input points contained in the IED database to be transferred to the PLC and obtained from the attached IED units. This array size ranges from 0 to 500.
Point_Number	0 to Number of configured points	Point number index to be overridden
Class	1 to 3	Class number assigned to point number specified
Deadband	0 to 32767	Deadband assigned to point number specified
Double_Inputs.	n/a	Number of counter points contained in the IED database to be transferred to the PLC and obtained from the attached IED units. This array size ranges from 0 to 250.
Point_Number	0 to Number of configured points	Point number index to be overridden
Class	1 to 3	Class number assigned to point number specified
Deadband	0 to 32767	Deadband assigned to point number specified

The default lengths of each point type array are 10 each. Each of the tag array sizes can be adjusted (when the PLC is offline) by editing the *DNPNET_Override* User Defined Datatype (UDT). This UDT can be found in RSLogix 5000 at:
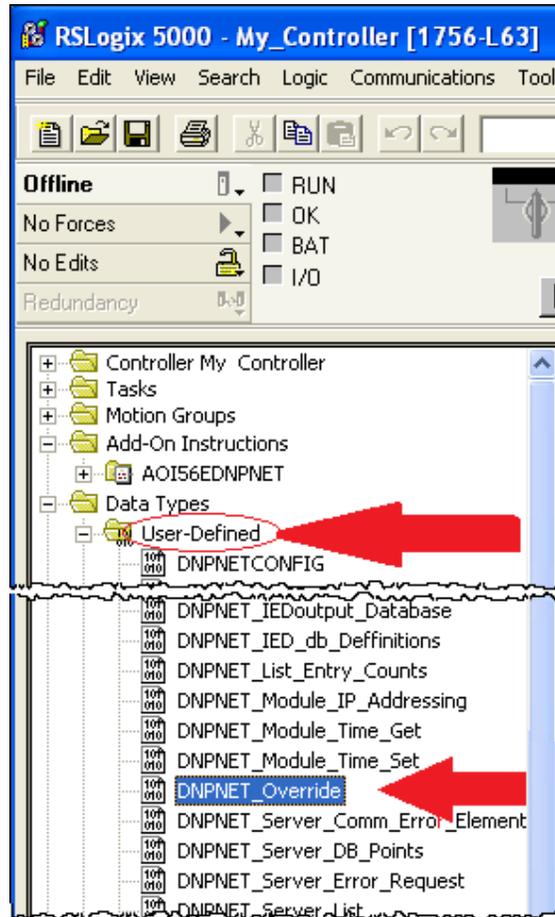
Double click the *DNPNET_Override* UDT. A window opens and displays the Members of this UDT. Notice the Members have similar names to the DNPNET tags to be edited. In the DataType column, the size of the specific tag array is determined by the value within the [ ].
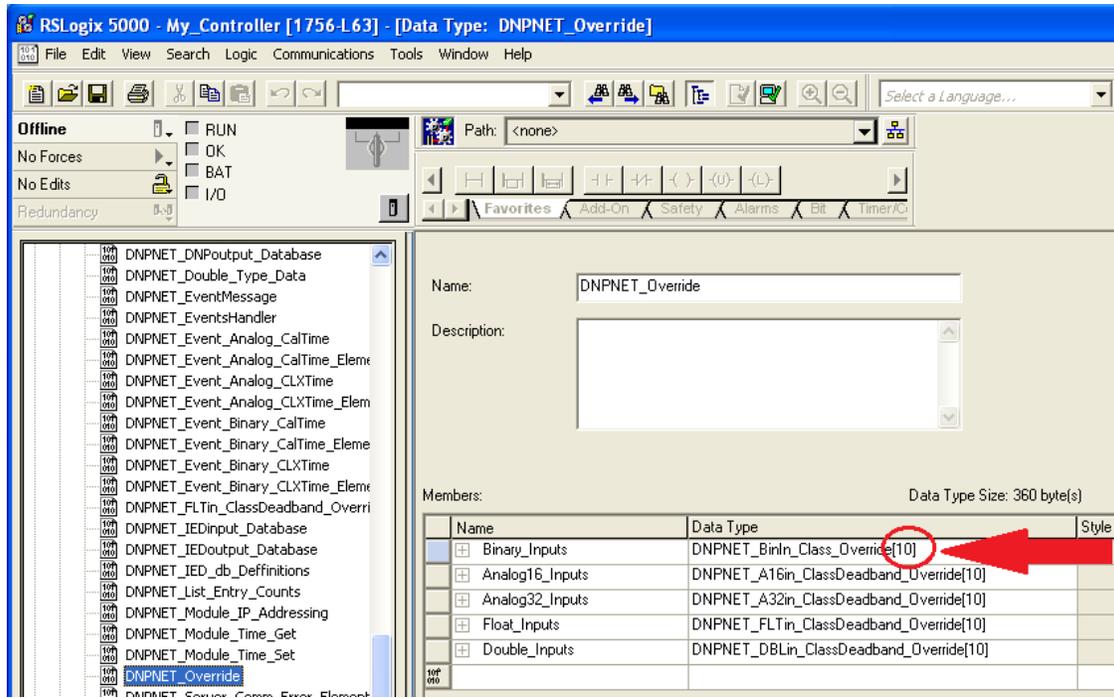
1 Manually edit these values as desired.
2 When finished, click **APPLY** and then **YES** to accept changes to the Data Type.
3 Click **OK** to close this window.
4 Save and download the RSLogix 5000 program to the processor and reboot the module to download settings to the unit.

2.3.7 DNPNET.CONFIG.DNP_Server_List[x]

This array assigns a list of server(s) the MVI56E-DNPNET client connects to. The default length of this array is five, and can by increased up to 40.

DNPNET.CONFIG.DNP_Server_List[x].	Range	Description
Address	0 to 32767	The address assigned to this server
Data_Link_Confirm_Mode	0=Never, 1=Sometimes 2=Always	IED can request acknowledgement from client station when sending data. Recommended value = 0.
Data_Link_Confirm_Timeout	1 to 32767 milliseconds	Time period to wait for client 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. Recommended value = 1000.
Data_Link_Confirm_Retries	0 to 255	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. Recommended value = 2.
Application_Layer_Response_Timeout	1 to 32767 milliseconds	Time-out period the client will wait for each response message fragment. If data link confirms are enabled, make sure the time-out period is set long enough to permit all data confirm retries.
Server_Mode	Bits 0, 1, 2, 3, 4	This word contains bits that define the server mode. The server mode defines the functionality of the server device and can be combined in any combination. Bit 0 = Enable Bit 1 = Unsolicited Message Bit 2 = Reserved Bit 3 = Auto Time Sync Bit 4 = Time Synch Variation: 　　　(0 = Var 3, 1 = Var 1)
IP_Address		IP Address of server
Port	0 to 65535	Server port number
IP_Type	0 or 1	0 = TCP Protocol, 1 = UDP Protocol
Connection_Retry_Interval	0 to 10000 milliseconds	This is how long in milliseconds the Client will wait before retrying to connect to a server that refused the connection. If this field is zero then the default of 1000 milliseconds will be set. The max is 10000 milliseconds (10 seconds). The minimum time is 1 millisecond. If this field is less than zero this server setup list entry will be completely ignored by the module.

The default Server List size is five. If there are more than five servers (up to a maximum of 40), you must increase the size of this list.

1 To edit the length of the Server List array, double-click the *DNPNETCONFIG* UDT
2 Edit the length inside the [ ] of the *DNPNET_Server_List[ ]* Data Type.
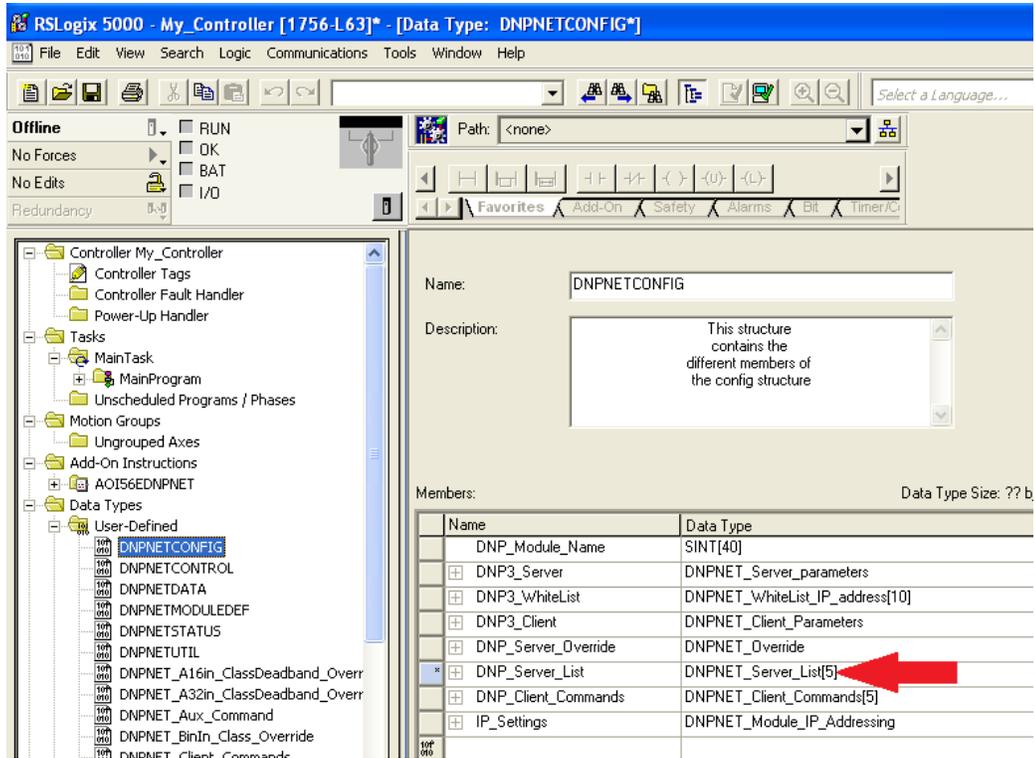
3 When finished, click **APPLY** and then click **YES** to accept changes to the Data Type.
4 Click **OK** to close this window.
5 Save and download the RSLogix 5000 program to the processor.
6 Reboot the MVI56E-DNPNET for changes to be downloaded to the module.

2.3.8 DNPNET.CONFIG.DNP_Client_Commands[x]

This array configures a list of commands the MVI56E-DNPNET client sends to DNP3 Ethernet servers. The default array size is five, and the maximum is 300.

DNPNET.CONFIG.DNP_ Client_Commands[x].	Range	Description
Port_Flags	Mapped bits [0:3]	Bits in the Port/Flags parameter are dependent on the data type. Clear 3rd bit to disable. Set 5th bit to select IED DB for write functions.
Server_Address	0 to 32767	This parameter specifies the DNP server address on the DNP network to which the command will be sent. (This is not the IP address of the server). The parameter has a range of 0 to 65535. The value of 65535 is reserved for broadcast messages. Verify that the server configuration information is set up in the module for each server defined in the command list.
Object		This parameter specifies the DNP Object type in the command. Valid Objects for the module are 1, 2, 10, 12, 20, 21, 30, 32, 40, 41, 50, 60 and 80. A value of 0 is permitted in this field for a set of special commands.
Variation		This parameter is specific to the object type selected.
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.
Point_Number_in_Server		This parameter specifies the starting point address in the remote server unit. 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.
Point_Count		This parameter defines the number of points in the IED unit that will be affected by the command. Refer to the discussion in the Command List topic, above, to interpret this parameter's meaning for the different Object types.
DNP_DB_Point_Number_ in_Client		This parameter defines the starting point address in the local Client's DNP database for the command. If the parameter has a value of -1, the DNP database is not used with the point.
IED_DB_Point_Number_ in_Client		This parameter defines the starting point address in the local Client's IED database for the command. If the parameter has a value of -1, the IED database is not used with the point.
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 this field. A value of 0 for the parameter implies that the command should be executed every scan of the list, as often as possible.

For Binary Input, Analog Input, and Counter data point types, the qualifier used in the data request is dependent on the Point Count and Address in Slave fields 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)

For Binary Output and Analog Output data point 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 or 28h (points from Address in Slave to Address in Slave + Point Count - 1)
- If Point Count <= 0, then ignore because this is illegal for outputs.

1 To edit the length of the Client Commands array, double-click the *DNPNETCONFIG* UDT.
2 Edit the length inside the [ ] of the *DNPNET_Client_Commands[ ]* Data Type.
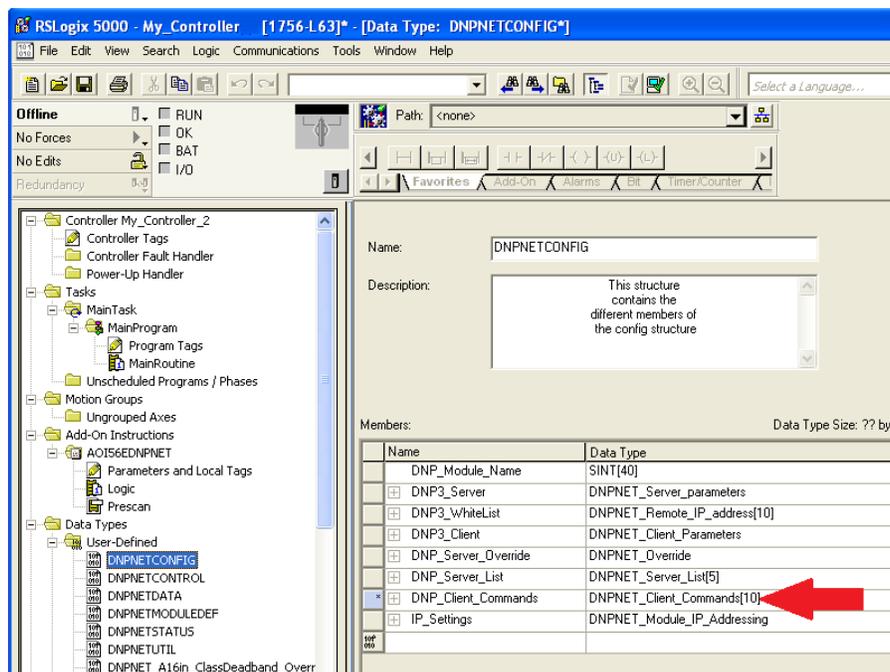
3 Click **APPLY** and then click **YES** to accept changes to the Data Type.
4 Click **OK** to close this window.
5 Save and download the RSLogix program to the processor. Reboot the MVI56E-DNPNET module to download changes to module.

2.3.9 DNPNET.CONFIG.IP_Settings[x]

Assigns the IP address configuration of the MVI56E-DNPNET.

DNPNET.CONFIG.IP_Settings.	Range	Description
Read	0 or 1	Triggers a request of the IP Address, Subnet Mask, and Gateway parameters from the module to the PLC. These parameters will display in the IP, Netmask, Gateway tags below.
Write	0 or 1	Triggers a transfer of the IP Address, Subnet Mask, and Gateway parameters from the PLC to the module. These parameters will come from the IP, Netmask, Gateway tags below.
Config.IP	0 to 255	IP Address of MVI56E-DNPNET
Config.Netmask	0 to 255	Subnet Mask of MVI56E-DNPNET
Config.Gateway	0 to 255	Gateway of MVI56E-DNPNET

After setting these parameters they must be sent to the module by triggering a download of these parameters.

2.3.10 DNP3 Quality Flag Control

Note: The *Quality Flag Control* feature is only available for v1.30.063 and higher.

DNP3 Quality Flags indicate the reliability and status of data points in a DNP3 communication system. These flags help ensure that the data being transmitted is accurate and trustworthy.

Quality Flags can be Enabled/Disabled based on the *Block_Timeout_MS* value.

- If the *Block_Timeout_MS* < 10,000, the *Quality Flags* are **enabled**. The module will report Quality Flags, along with DNP data.
- If the *Block_Timeout_MS* ≥ 10,000, the *Quality Flags* are **disabled**. The module will force the Quality Flags to report only a status of **Online**, along with DNP data.

For example, if a value of **10,000** is entered, the *Block Timeout Value* will be calculated as **0**. Since **0** is out of range, the *Block_Timeout_MS* parameter will be set to the default value of **1500** and the *Quality Flags* will be **disabled**.

The *Block_Timeout_MS* value and *Quality Flag Control* status can be checked at *PCB Diagnostics > DNP3 Server > Config*.

3 Diagnostics and Troubleshooting

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

- LED status indicators on the front of the module provide information on the module's status.
- Status data contained in the module can be viewed in *ProSoft Configuration Builder* through the Ethernet port.
- Status data values are transferred from the module to the processor.

3.1 Ethernet LED Indicators

The Ethernet LEDs indicate the module's Ethernet port status as follows:

LED	State	Description
Data	OFF	Ethernet connected at 10Mbps duplex speed
	AMBER Solid	Ethernet connected at 100Mbps duplex speed
Link	OFF	No physical network connection is detected. No Ethernet communication is possible. Check wiring and cables.
	GREEN Solid or Blinking	Physical network connection detected. This LED must be ON solid for Ethernet communication to be possible.

3.1.1 Scrolling LED Status Indicators

The scrolling LED display indicates the module's operating status as follows:

Initialization Messages

Code	Message
Boot / DDOK	Module is initializing
Ladd	Module is waiting for required module configuration data from ladder logic to configure the Modbus ports
Waiting for Processor Connection	Module did not connect to processor during initialization ■ Sample ladder logic or AOI is not loaded on processor ■ Module is located in a different slot than the one configured in the ladder logic/AOI ■ Processor is not in RUN or REM RUN mode
Last config: <date>	Indicates the last date when the module changed its IP address. You can update the module date and time through the Connect to the Module's Webpage (page 80), or with the MVI56E Add-On Instruction.

Operation Messages

After the initialization step, the following message pattern will be repeated.

```
<Backplane Status> <IP Address> <Backplane Status> <Port Status>
```

Code	Message
<Backplane Status>	OK: Module is communicating with processor ERR: Module is unable to communicate with processor. For this scenario, the <Port Status> message above is replaced with "Processor faulted or is in program mode".
<IP Address>	Module IP address
<Port Status>	OK: Port is communicating without error Client/Server Communication Errors: port is having communication errors. Refer to Diagnostics and Troubleshooting (page 38) for further information about the error.

3.1.2 Non-Scrolling LED Status Indicators

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

LED Label	Status	Indication
APP	OFF	The module is not receiving adequate power or is not securely plugged into the rack. May also be OFF during configuration download.
	GREEN	The MVI56E-DNPNET is working normally.
	RED	The most common cause is that the module has detected a communication error during operation of an application port. The following conditions may also cause a RED LED: ▪ The firmware is initializing during startup ▪ The firmware detects an on-board hardware problem during startup ▪ Failure of application port hardware during startup ▪ The module is shutting down ▪ The module is rebooting due to a ColdBoot or WarmBoot request from the ladder logic or Debug Menu
OK	OFF	The module is not receiving adequate power or is not securely plugged into the rack.
	GREEN	The module is operating normally.
	RED	The module has detected an internal error or is being initialized. If the LED remains RED for over 10 seconds, the module is not working. Remove it from the rack and re-insert it to restart its internal program.
ERR	RED	Not used.

3.2 Clearing a Fault Condition

Typically, if the OK LED on the front of the module remains RED for more than ten seconds, a hardware problem has been detected 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 Re-insert the card in the rack and turn the power back on.
5 Verify correct configuration data is being transferred to the module from the ControlLogix controller.

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 Technical Support.

3.3　Troubleshooting

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

Processor Errors

Problem Description	Steps to take
Processor Fault	Verify that the module is plugged into the slot that has been configured for the module in the I/O Configuration of RSLogix. Verify that the slot location in the rack has been configured correctly in the ladder logic.
Processor I/O LED flashes	This indicates a problem with backplane communications. A problem could exist between the processor and any installed I/O module, not just the MVI56E-DNPNET. Verify that all modules in the rack are correctly configured in the ladder logic.

Module Errors

Problem Description	Steps to take
MVI56E modules with scrolling LED display: *<Backplane Status>* condition reads ERR	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 or REM RUN mode. ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write data block transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is properly configured in the processor I/O configuration and ladder logic.
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.

3.4   Setting Up ProSoft Configuration Builder

ProSoft Configuration Builder (PCB) software provides a convenient way to configure, monitor, and troubleshoot your MVI56E-DNPNET module.

3.4.1 Installing ProSoft Configuration Builder

Use the ProSoft Configuration Builder (PCB) software to configure the module. You can find the latest version of the ProSoft Configuration Builder (PCB) on our web site: www.prosoft-technology.com. The installation filename contains the PCB version number.

Installing PCB from the ProSoft website:

1 Open a browser window and navigate to www.prosoft-technology.com.
2 Perform a search for 'pcb' in the Search bar. Click on the ProSoft Configuration Builder search result.
3 On the PCB page, click the download link for ProSoft Configuration Builder, and save the file to your Windows desktop.
4 After the download completes, double-click the file to install. If you are using Windows 7, right-click the PCB installation file and then choose **RUN AS ADMINISTRATOR**. Follow the instructions that appear on the screen.
5 If you want to find additional software specific to your MVI56E-DNPNET, enter the model number into the ProSoft website search box and press the **ENTER** key.

3.4.2 Setting Up the Project

To begin, start **PROSOFT CONFIGURATION BUILDER** (PCB).
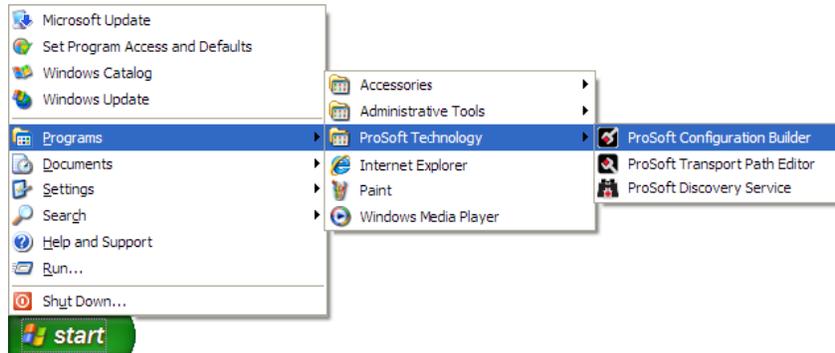
If you have used other Windows configuration tools before, you will find the screen layout familiar. *PCB*'s window consists of a tree view on the left, and an information pane and a configuration pane on the right side of the window. When you first start *PCB*, the tree view consists of folders for *Default Project* and *Default Location*, with a *Default Module* in the *Default Location* folder. The following illustration shows the *PCB* window with a new project.
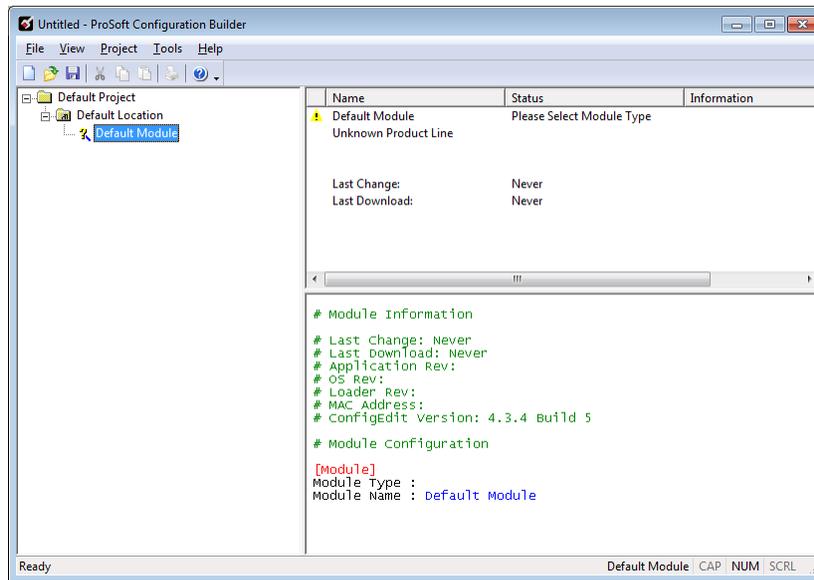
Your first task is to add the MVI56E-DNPNET module to the project.

1 Right-click **DEFAULT MODULE** in the tree view and then choose **CHOOSE MODULE TYPE**. This opens the *Choose Module Type* dialog box.
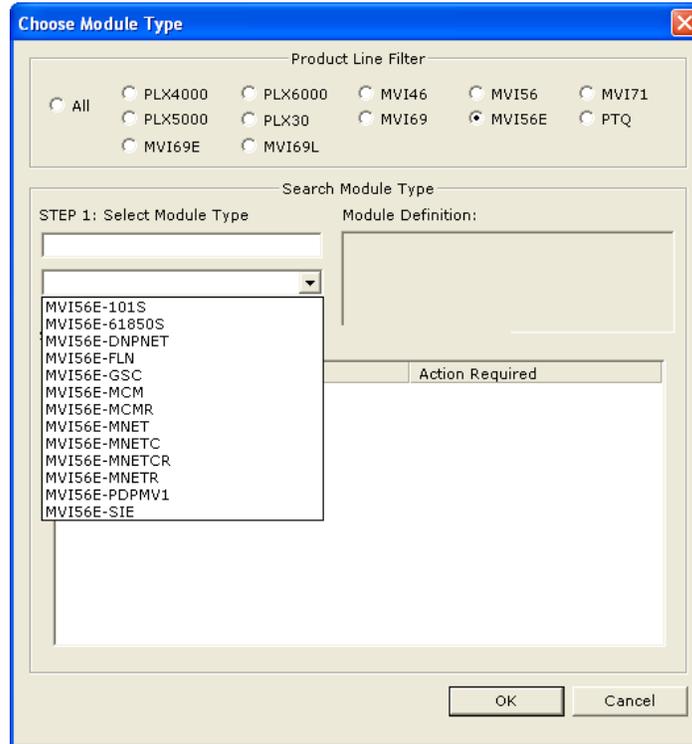
2 In the *Product Line Filter* area of the dialog box, select **MVI56E**. In the *Select Module Type* dropdown list, select **MVI56E-DNPNET**, and then click **OK** to save your settings and return to the *ProSoft Configuration Builder* window.

3.5   Connecting Your PC to the Module

3.5.1 Using CIPconnect® to Connect to the Module

You can use CIPconnect® to connect a PC to the ProSoft Technology MVI56E-DNPNET module over Ethernet using Rockwell Automation's 1756-ENBT EtherNet/IP® module. This allows you to configure the MVI56E-DNPNET network settings and view module diagnostics from a PC. RSLinx is not required when you use CIPconnect. All you need are:

- The IP addresses and slot numbers of any 1756-ENBT modules in the path
- The slot number of the MVI56E-DNPNET in the destination ControlLogix chassis (the last ENBTx and chassis in the path).

To use CIPconnect, follow these steps:

1 In the tree view in *ProSoft Configuration Builder*, right-click the **MVI56E-DNPNET** icon and then choose **DIAGNOSTICS.**
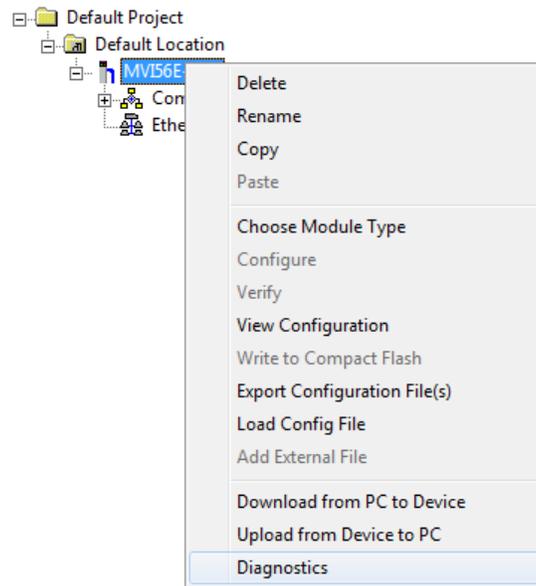
2 In the *Diagnostics* window, click the **SET UP CONNECTION** button.

3 In the *Select Connection Type* dropdown list, choose **1756-ENBT**. The default path appears in the text box, as shown in the following illustration.
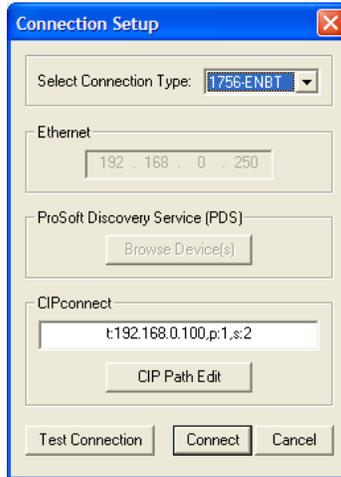
4 Click **CIP PATH EDIT** to open the *CIPconnect Path Editor* dialog box.
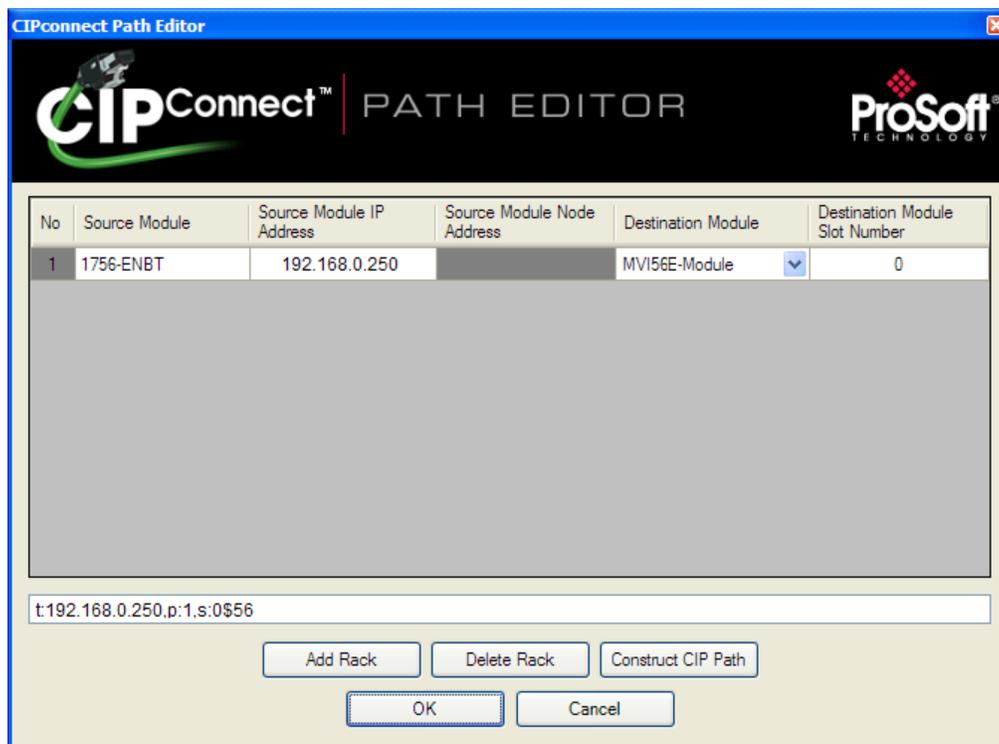
The *CIPconnect Path Editor* allows you to define the path between the PC and the MVI56E-DNPNET module. The first connection from the PC is always a 1756-ENBT (Ethernet/IP) module.

Each row corresponds to a physical rack in the CIP path.

- If the MVI56E-DNPNET module is located in the same rack as the first 1756-ENBT module, select **RACK NO. 1** and configure the associated parameters.
- If the MVI56E-DNPNET is available in a remote rack (accessible through ControlNet or Ethernet/IP), include all racks (by using the **ADD RACK** button).

Parameter	Description
Source Module	Source module type. This field is automatically selected depending on the destination module of the last rack (1756-CNB or 1756-ENBT).
Source Module IP Address	IP address of the source module (only applicable for 1756-ENBT)
Source Module Node Address	Node address of the source module (only applicable for 1756-CNB)
Destination Module	Select the destination module associated to the source module in the rack. The connection between the source and destination modules is performed through the backplane.
Destination Module Slot Number	The slot number where the destination MVI56E module is located.

To use the CIPconnect Path Editor, follow these steps:

1 Configure the path between the 1756-ENBT connected to your PC and the MVI56E-DNPNET module.
  - o If the module is located in a remote rack, add more racks to configure the full path.
  - o The path can only contain ControlNet or Ethernet/IP networks.
  - o The maximum number of supported racks is six.
2 Click **CONSTRUCT CIP PATH** to build the path in text format
3 Click **OK** to confirm the configured path.

The following examples should provide a better understanding on how to set up the path for your network.

Example 1: Local Rack Application

For this example, the MVI56E-DNPNET module is located in the same rack as the 1756-ENBT that is connected to the PC.
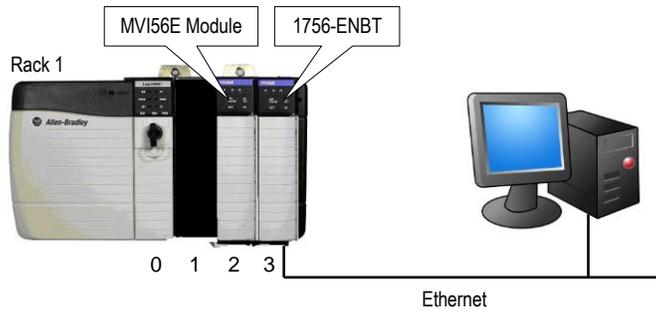
Rack 1

Slot	Module	Network Address
0	ControlLogix Processor	-
1	Any	-
2	MVI56E-DNPNET	-
3	1756-ENBT	IP=192.168.0.100

1 In the tree view in *ProSoft Configuration Builder*, right-click the **MVI56E-DNPNET** icon to open a shortcut menu.

2 On the shortcut menu, choose **DIAGNOSTICS.**

3 In the *Diagnostics* window, click the **SET UP CONNECTION** button.

4 In the *Connection Setup* dialog box, click **CIP PATH EDIT.**

5 Configure the path as shown in the following illustration, and click **CONSTRUCT CIP PATH** to build the path in text format.

Click **OK** to close the *CIPconnect Path Editor* and return to the *Connection Setup* dialog box.

6 Check the new path in the *Connection Setup* dialog box.

7 Click **TEST CONNECTION** to verify that the physical path is available. The following message should be displayed upon success.

8 Click **OK** to close the Test Connection pop-up and then click **CONNECT** to close the *Connection Set* up dialog box. The Diagnostics menu is now connected through CIPconnect.

Example 2: Remote Rack Application

For this example, the MVI56E-DNPNET module is located in a remote rack accessible through ControlNet, as shown in the following illustration.
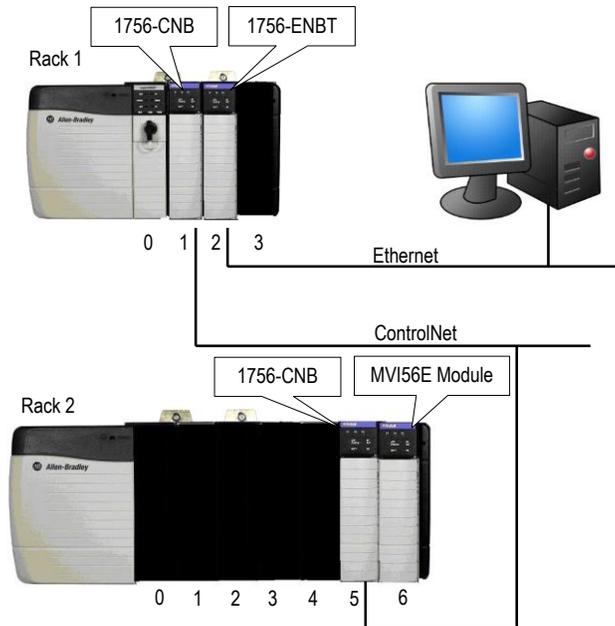
Rack 1

Slot	Module	Network Address
0	ControlLogix Processor	-
1	1756-CNB	Node = 1
2	1756-ENBT	IP=192.168.0.100
3	Any	-

Rack 2

Slot	Module	Network Address
0	Any	-
1	Any	-
2	Any	-
3	Any	-
4	Any	-
5	1756-CNB	Node = 2
6	MVI56E-DNPNET	-

1 In the tree view in *ProSoft Configuration Builder*, right-click the **MVI56E-DNPNET** icon to open a shortcut menu.

2 On the shortcut menu, choose **DIAGNOSTICS.**

3 In the *Diagnostics* window, click the **SET UP CONNECTION** button.

4 In the *Connection Setup* dialog box, click **CIP PATH EDIT.**

5 Configure the path as shown in the following illustration and click **CONSTRUCT CIP PATH** to build the path in text format.

Click **OK** to close the *CIPconnect Path Editor* and return to the *Connection Setup* dialog box.

6 Check the new path in the *Connection Setup* dialog box.

7   Click **TEST CONNECTION** to verify that the physical path is available. The following
message should be displayed upon success.

8   Click **OK** to close the Test Connection pop-up and then click **CONNECT** to close the
Connection Set up dialog box. The Diagnostics menu is now connected through
CIPconnect.

3.5.2 Using RSWho to Connect to the Module

You need to have RSLinx installed on your PC to use this feature. You also need an ENBT module set up in the rack. For information on setting up the ENBT module, see Using CIPconnect® to Connect to the Module (page 45).

1 In *ProSoft Configuration Builder*, click the **PROJECT** menu, then choose **MODULE > DOWNLOAD FROM PC TO DEVICE**.
2 In the *Download* dialog box, choose **1756 ENBT** from the *Select Connection Type* dropdown box.
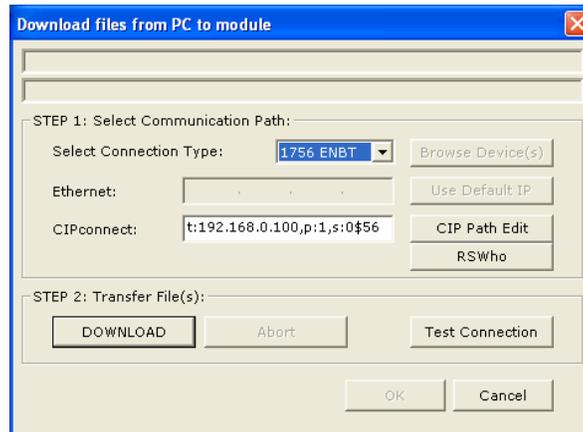
3 Click **RSWHO** to display modules on the network. The MVI56E-DNPNET module will automatically be identified on the network.
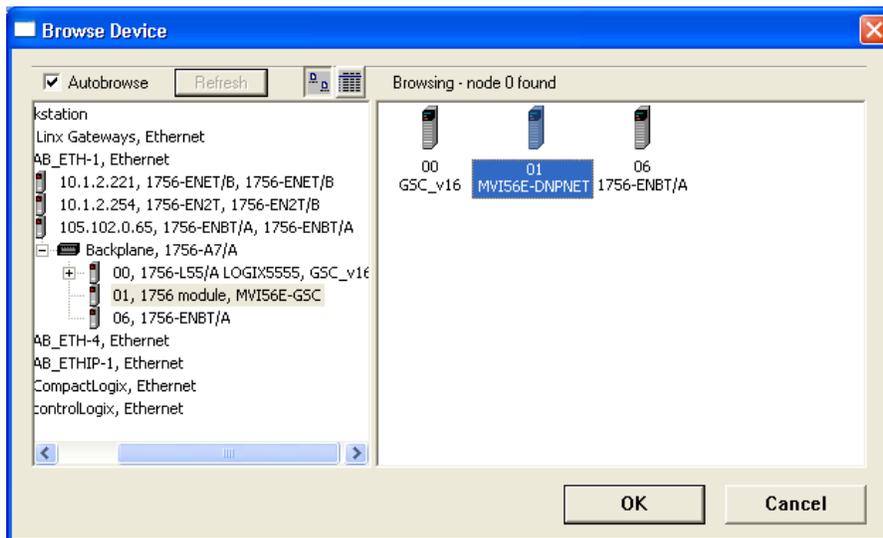
4 Select the module, and then click **OK**.
5 In the *Download* dialog box, click **DOWNLOAD**.

3.5.3 Connecting Your PC to the Module's Ethernet Port

With the module securely mounted, connect one end of the Ethernet cable to the **CONFIG (E1)** Port, and the other end to an Ethernet hub or switch accessible from the same network as your PC. Or, you can connect directly from the Ethernet Port on your PC to the **CONFIG (E1)** Port on the module.
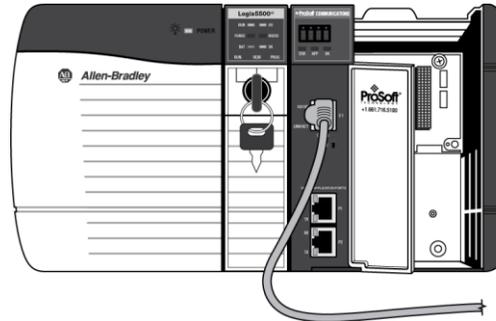
3.6 Using the Diagnostics Menu in ProSoft Configuration Builder

The *Diagnostics* menu, available through the Ethernet configuration port for this module, is arranged as a tree structure, with the *Main* menu at the top of the tree, and one or more submenus for each menu command. The first menu you see when you connect to the module is the *Main* menu.

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

To connect to the module, refer to Connecting Your PC to the Module (page 45).

1 In the tree view in *ProSoft Configuration Builder*, right-click the **MVI56E-DNPNET** icon to open a shortcut menu.

2 On the shortcut menu, choose **DIAGNOSTICS.**
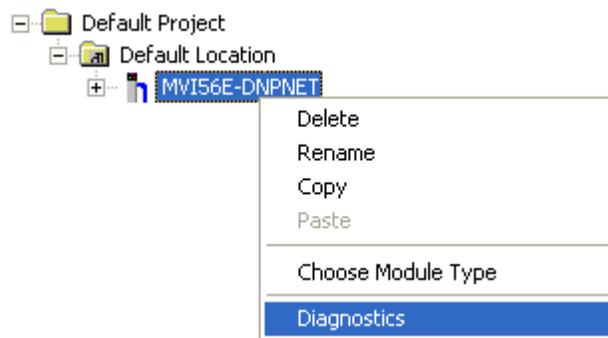
3 In the *Diagnostics* window, click the **SET UP CONNECTION** button.

Click to set up connection

4 In the *Ethernet* field of the *Connection Setup* dialog box, enter the IP address that was assigned the module in Assigning a Permanent IP Address (page 22). In the *Connection Setup* dialog box, click the **TEST CONNECTION** button to verify that the module is accessible with the current settings.
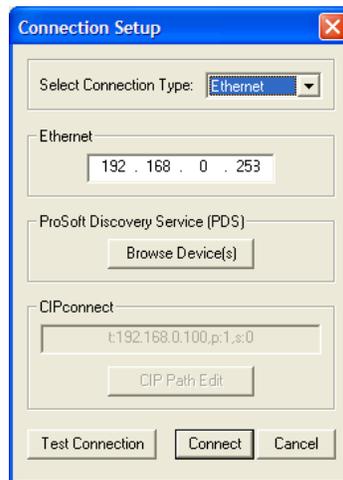
You can also use CIPconnect® to connect to the module through a 1756-ENBT card by choosing *1756-ENBT* in the **SELECT CONNECTION TYPE** list. Refer to Using CIPconnect® to Connect to the Module (page 45) for information on how to construct a CIP path.
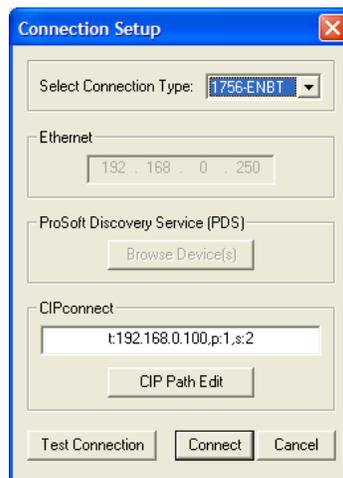
5 If the *Test Connection* is successful, click **CONNECT** to display the *Diagnostics* menu in the Diagnostics Window.
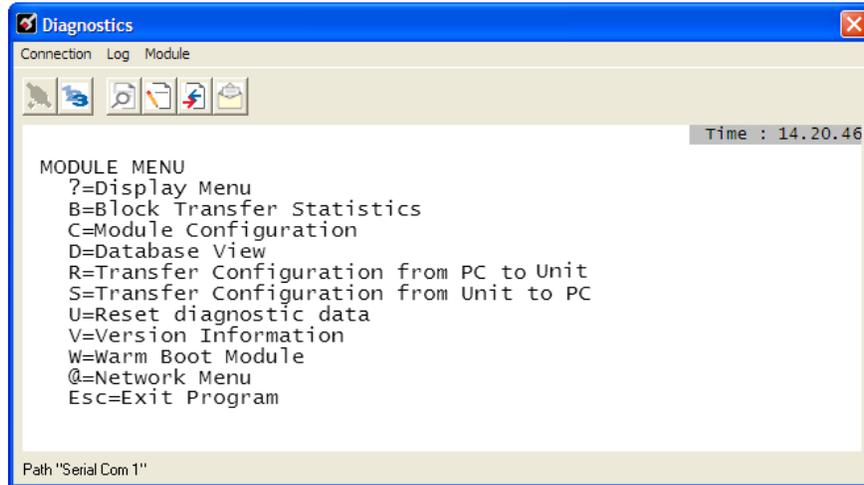
If *PCB* is unable to connect to the module:

1 Click the **BROWSE DEVICE(S)** button to open the *ProSoft Discovery Service*. Select the module, then right-click and choose **SELECT FOR PCB**.
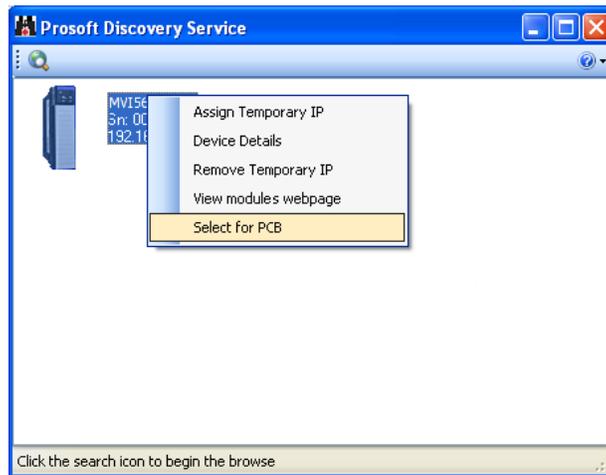
2 Close *ProSoft Discovery Service*, and click the **CONNECT** button again.
3 If these troubleshooting steps fail, verify that the Ethernet cable is connected properly between your computer and the module, either through a hub or switch or directly between your computer and the module.

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

3.6.1 The Diagnostics Menu

The Diagnostics 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.

3.6.2 Monitoring General Information

Use *MODULE > Info* to view module version information.
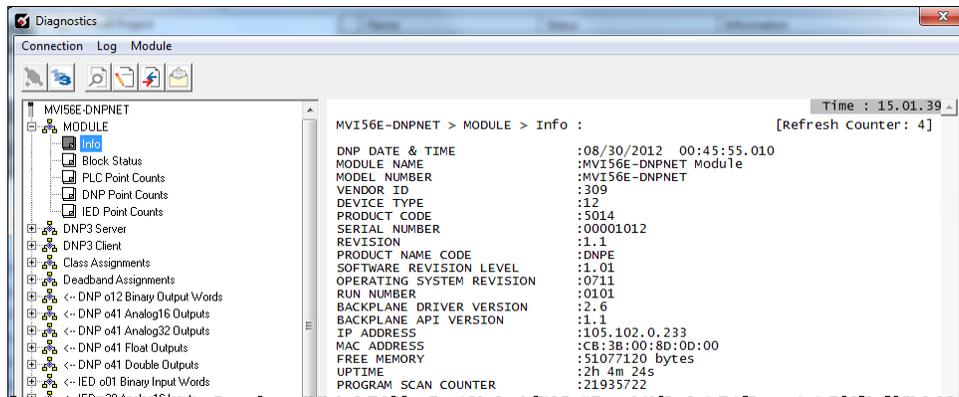
3.6.3 Monitoring Backplane Information

Use *MODULE > Block Status* menu to view the backplane status information for the MVI56E-DNPNET module.
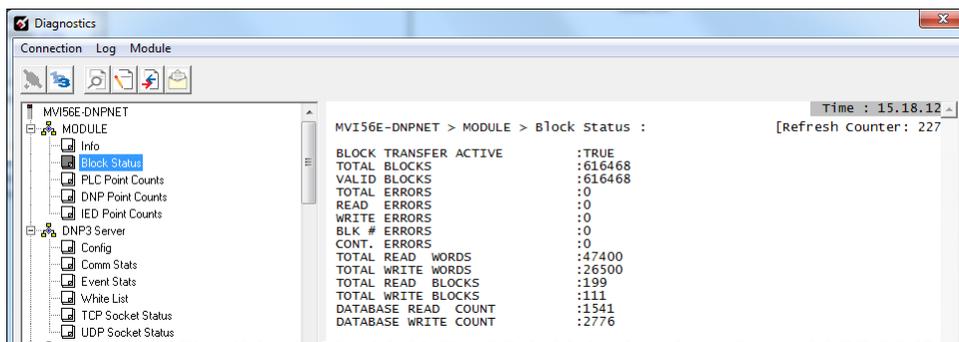
3.6.4 DNP3 Ethernet Point Count Module Information

Use *MODULE > PLC, DNP, and IED Point Counts* to view point count information for the MVI56E-DNPNET module.
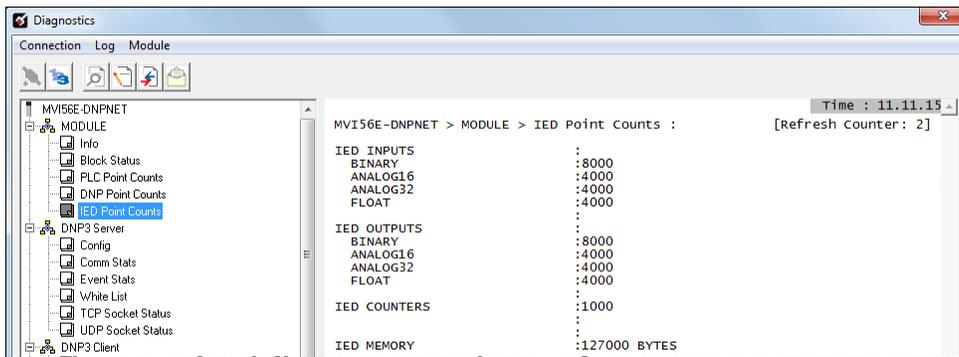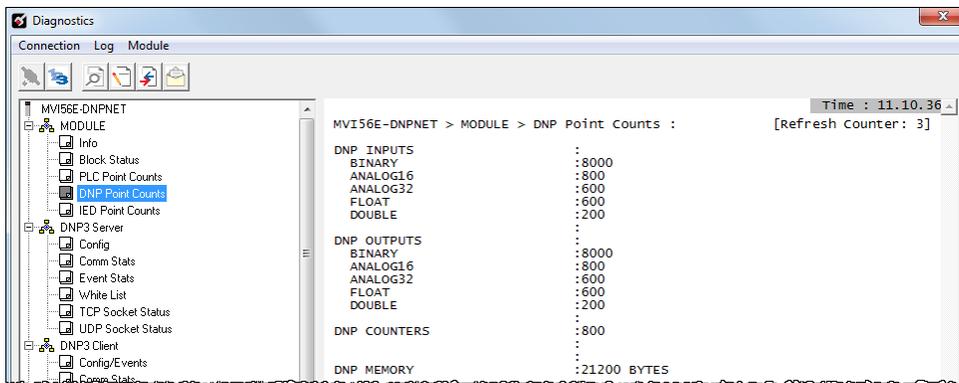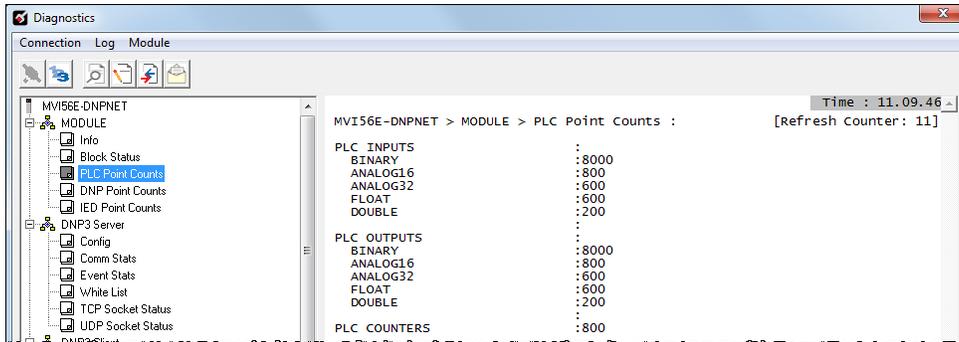
3.6.5 Monitoring MVI56E-DNPNET Server Information

Use the *DNP3 Server* menu to view the following server information for the MVI56E-DNPNET module:

* Configuration
* Communication Status
* Event Status
* Whitelist
* TCP Socket Status
* UDP Socket Status
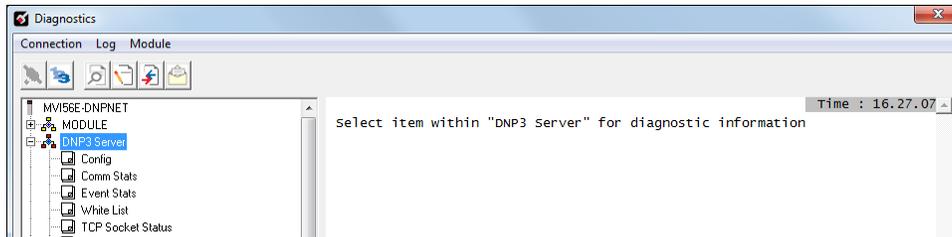
DNP3 Server - Config

This option displays the current configuration settings for the DNP3 Server.
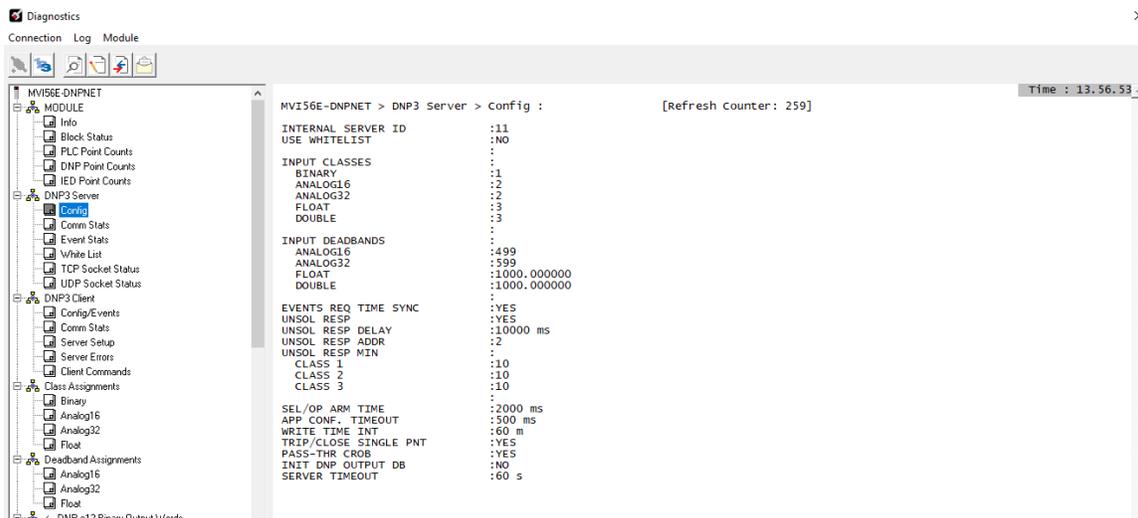
Parameter		Description
Internal Server ID		The node address of the internal server of the MVI56E-DNPNET.
Use Whitelist		**Yes** = The server will only respond to clients within the whitelist group.
		No = The server will respond to clients of any IP address.
Input Classes	Binary	The configured class type for Binary Inputs.
	Analog16	The configured class type for 16-bit Analog Inputs.
	Analog32	The configured class type for 32-bit Analog Inputs.
	Float	The configured class type for 16-bit Float Inputs.
	Double	The configured class type for Double Inputs.
Input Deadbands	Analog16	The configured deadband value assigned to all Analog16 Input points.
	Analog32	The configured deadband value assigned to all Analog32 Input points.
	Float	The configured deadband value assigned to all Float Input points.
	Double	The configured deadband value assigned to all Double Input points.
Events Req Time Sync		**Yes** = The server will not generate events if the time is not synched with the client. **No** = The server will generate events if the time is not synched with the client.
Unsol Resp		Indicates if the server will send unsolicited responses or not.
Unsol Resp Delay		Configured time to wait (ms) after an event occurs before sending an unsolicited response.
Unsol Resp Addr		Configured client address that the module will send unsolicited responses to.
Unsol Resp Min	Class 1	Configured minimum number of events in Class 1 required before an unsolicited response will be sent.
	Class 2	Configured minimum number of events in Class 2 required before an unsolicited response will be sent.
	Class 3	Configured minimum number of events in Class 3 required before an unsolicited response will be sent.
SEL/OP Arm Time		Configured time (ms) after the operate command is received that the server will perform the operate command.
App Conf. Timeout		Confirmed timeout (ms) for the application layer confirms from the DNP3 client.
Write Time INT		Configured time period (minutes) that the server will wait before it sets the Need Time IIN bit.
Trip/Close Single PNT		**Yes** = CROB commands will operate on the database as single bits. **No** = CROB commands will operate on the database as dual points.
Pass-THR CROB		**Yes** = The module will pass all received CROB messages to the processor.
Init DNP Output DB		**Yes** = The module will request data from the processor to initialize the DNP Output database.
Server Timeout		Configured TCP server connection timeout.

DNP3 Server - Comm Stats

This option displays the current DNP3 Server communication statistics.
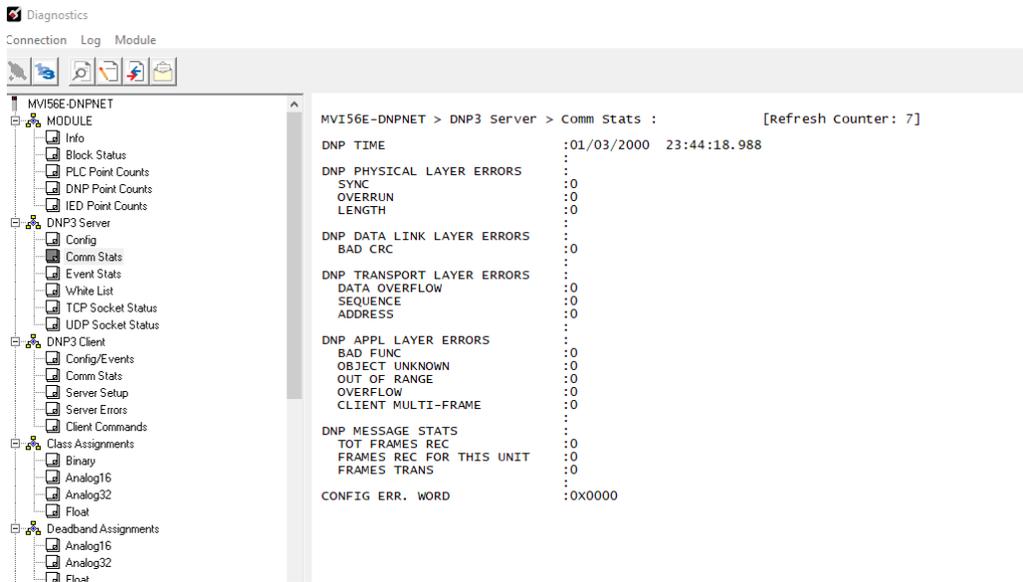
Parameter		Description
DNP Time		Current time of the module.
DNP Physical Layer Errors	Sync	Total number of DNP3 Physical Layer synchronization Errors.
	Overrun	Total number of DNP3 Physical Layer overrun Errors.
	Length	Total number of DNP3 Physical Layer length Errors.
DNP Data Link Layer Errors	Bad CRC	Total number of DNP3 Data Link Layer bad CRC Errors.
DNP Transport Layer Errors	Data Overflow	Total number of DNP3 Transport Layer user data overflow Errors.
	Sequence	Total number of DNP3 Transport Layer sequence Errors.
	Address	Total number of DNP3 Transport Layer address Errors.
DNP Application Layer Errors	Bad Func	Total number of DNP3 Application layer bad function code errors.
	Object Unknown	Total number of DNP3 Application layer object unknown errors.
	Out of Range	Total number of DNP3 Application layer 'out of range' errors.
	Client Multi-Frame	Total number of DNP3 Application layer multi-frame errors.
DNP Message Stats	Tot Frames Rec	Total number of message frames recorded for the server.
	Frames Rec for this Unit	Number of message frames received by server.
	Frames Trans	Number of message frames transmitted by for server.
Config Err Word		The configuration error word for the module. Refer to page 76 for more information.

DNP3 Server - Event Stats

This option displays the current event statistics of the DNP3 Server.
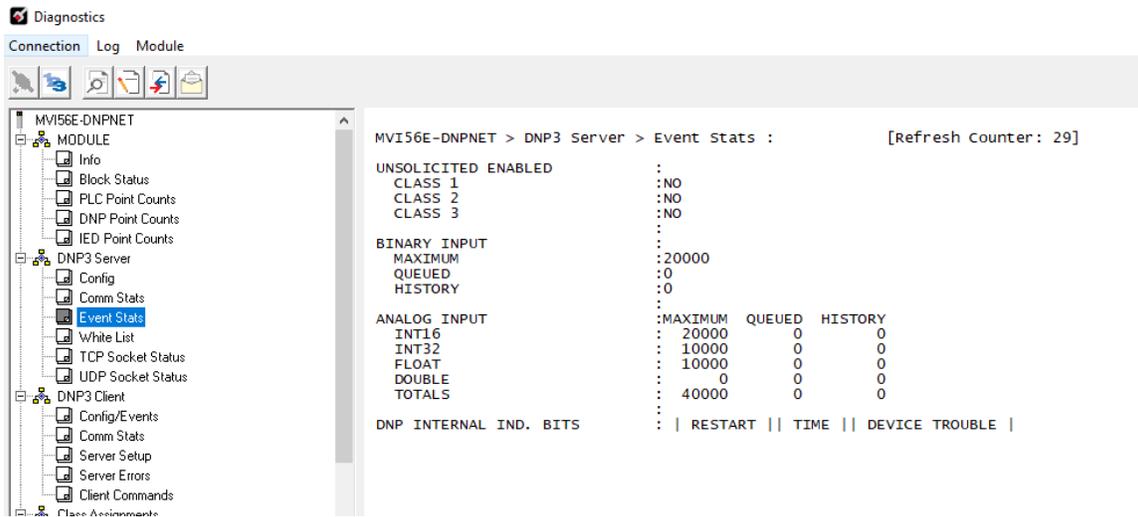
Parameter		Description
Unsolicited Enabled	Class 1	Indicates whether Class 1 unsolicited responses have been enabled or not by the client.
	Class 2	Indicates whether Class 2 unsolicited responses have been enabled or not by the client.
	Class 3	Indicates whether Class 3 unsolicited responses have been enabled or not by the client.
Binary Input	Maximum	Maximum amount of Binary Input events allowed in the buffer.
	Queued	Number of queued Binary Input events in the buffer.
	History	Number of generated Binary Input events by the DNP3 server.
Analog Input	INT16, INT32, Float, Double, Totals	*Maximum*, *Queued*, and *History* definitions are similar to the *Binary Input* definitions.
DNP Internal Ind. Bits		Displays the current DNP Internal Indication bits for the DNP3 Server.

DNP3 Server - Whitelist

This option displays the current Whitelist settings and information.
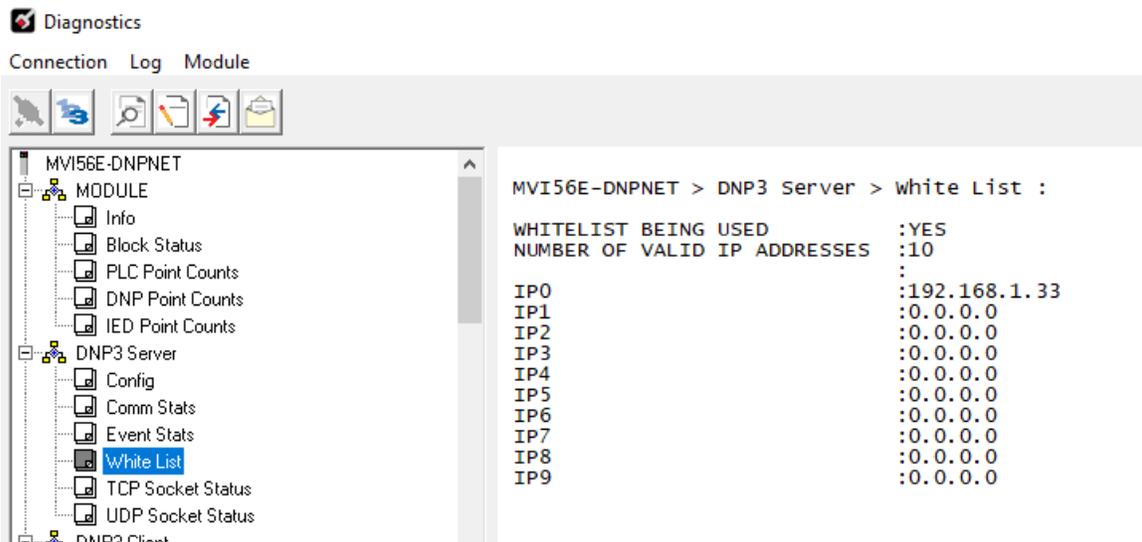
Parameter	Description
Whitelist Being Used	**Yes** = Whitelist group functionality is enabled.
	No = Whitelist group functionality is disabled.
Number of valid IP addresses	Number of valid IP addresses within the Whitelist group.
IP0 to IP9	List of IP addresses within the Whitelist group.

DNP3 Server - TCP Socket Status

This option displays the current TCP socket status of the DNP3 Server.
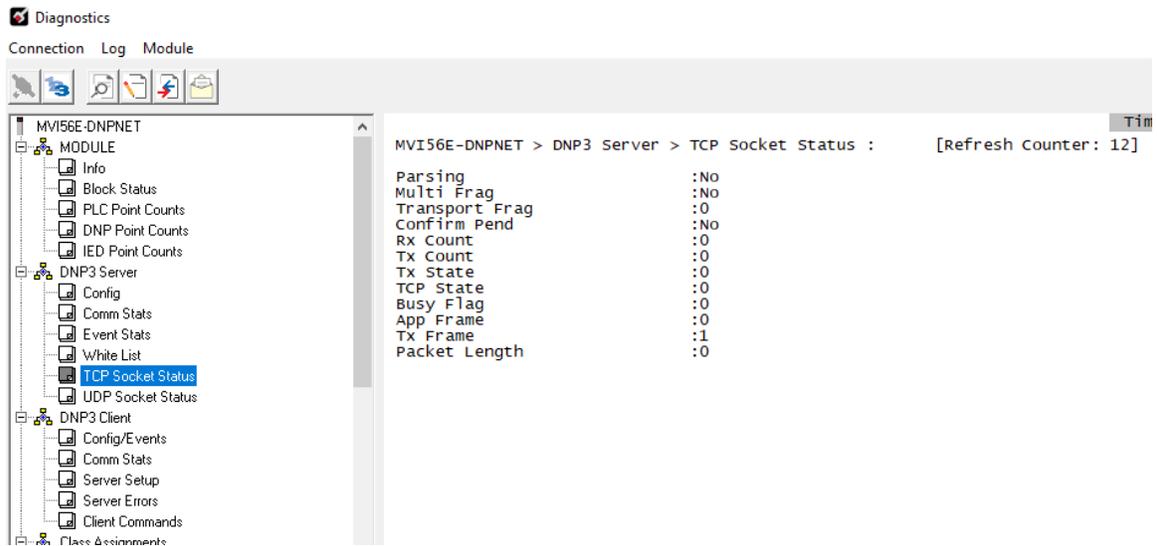
Parameter	Description
Parsing	Indication of status of TCP Parsing.
Multi Frag	Indicates if DNP3 Server is receiving a Muti-Fragment Message.
Transport Frag	Number of fragmented transport messages.
Confirm Pend	Indicates if the DNP3 server is waiting for an application confirm from the client.
Rx Count	Number of received TCP messages.
Tx Count	Number of transmitted TCP messages.
Tx State	Port 1 transmit status. **0** = Not communicating **1** = Initial direct connection **2** = Turn on wait state **3** = Turn off wait state **4** = Turn off wait state
TCP State	Indicates the TCP Communication state. **0** = Not communicating using TCP **1** = Communicating with TCP **2** = Receiving a Muti-Fragment Message
Busy Flag	Communication Busy Indicator. **0** = Not Busy **1** = TCP Busy **2** = UDP Busy
App Frame	Application Frame Fragmented Message Indicator: **0** = No **1** = Yes
Tx Frame	
Packet Length	Packet Length of current TCP Message.

DNP3 Server - UDP Socket Status

This option displays the current UDP socket status of the DNP3 Server.
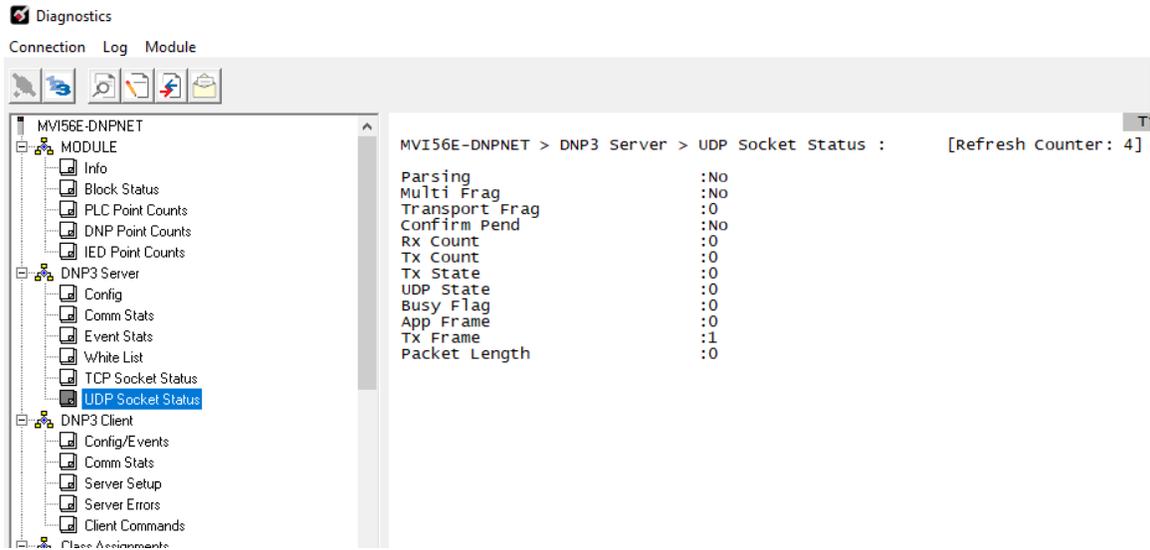
Parameter	Description
Parsing	Indication of status of UDP Parsing.
Multi Frag	Indicates if DNP3 Server is receiving a Muti-Fragment Message.
Transport Frag	Number of fragmented transport messages.
Confirm Pend	Indicates if the DNP3 server is waiting for an application confirm from the client.
Rx Count	Number of received UDP messages.
Tx Count	Number of transmitted UDP messages.
Tx State	Port 1 transmit status. **0** = Not communicating **1** = Initial direct connection **2** = Turn on wait state **3** = Turn off wait state **4** = Turn off wait state
UDP State	Indicates the UDP Communication state. **0** = Not communicating using UDP **1** = Communicating with UDP **2** = Receiving a Muti-Fragment Message
Busy Flag	Communication Busy Indicator. **0** = Not Busy **1** = TCP Busy **2** = UDP Busy
App Frame	Application Frame Fragmented Message Indicator: **0** = No **1** = Yes
Tx Frame	
Packet Length	Packet Length of current UDP Message.

3.6.6 Monitoring MVI56E-DNPNET Client Information

Use the *DNP3 Client* menu to view the following client information for the MVI56E-DNPNET module:

- Configuration and Events
- Communication Status
- Server Setup
- Server Errors
- Client Commands List
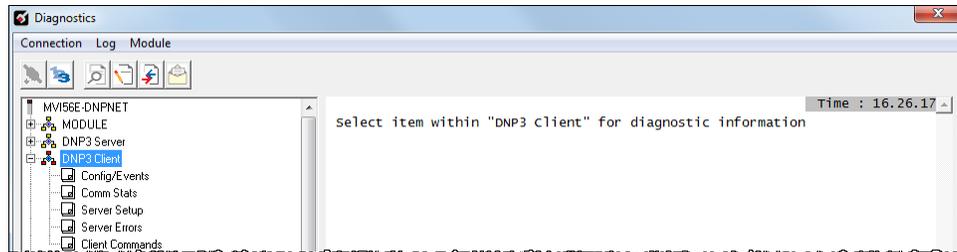
DNP Client - Config/Events

This option displays the current configuration settings for the DNP3 Client.
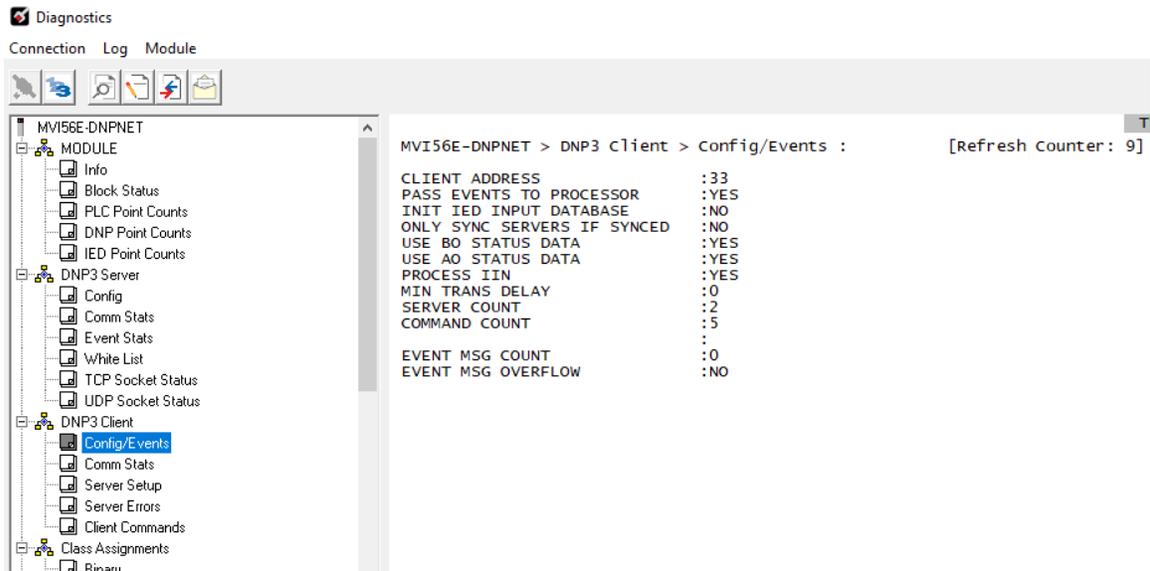
Parameter	Description
Client Address	Configured DNP3 address for the client.
Pass Events to Processor	**Yes** = All events stored within the module will be sent to the processor.
Init IED Input Database	Indicates if the module will request data from the processor to initialize the IED Input Database.
Only Sync Servers if Synced	**Yes** = The module will send a time sync message to servers when its own time has not yet been synced.
Use BO status data	Indicates if the database and functions for reading the Binary Output status (Object 10) is enabled.
Use AO status data	Indicates if the database and functions for reading the Analog Output status (Object 40) is enabled.
Process IIN	Indicates if the client will or will not send requests to servers if their IIN bits are set for class data, need time, restart, or buffer overflow. If enabled, the client will make requests to servers that have these IIN bits set until these IIN bits have been cleared.
Min Trans Delay	The minimum time (in milliseconds) between receiving last byte and transmitting first byte. Gives master time to disable transmitter on RS485 networks.
Server Count	The current connected server count.
Command Count	The current number of enabled client commands for the DNP3 client.
Event Msg Count	Total number of received event messages from DNP3 servers.
Event Msg Overflow	Indicates if the client event messages buffer has overflowed.

DNP Client - Comm Stats

This option displays the current DNP3 Client communication statistics.
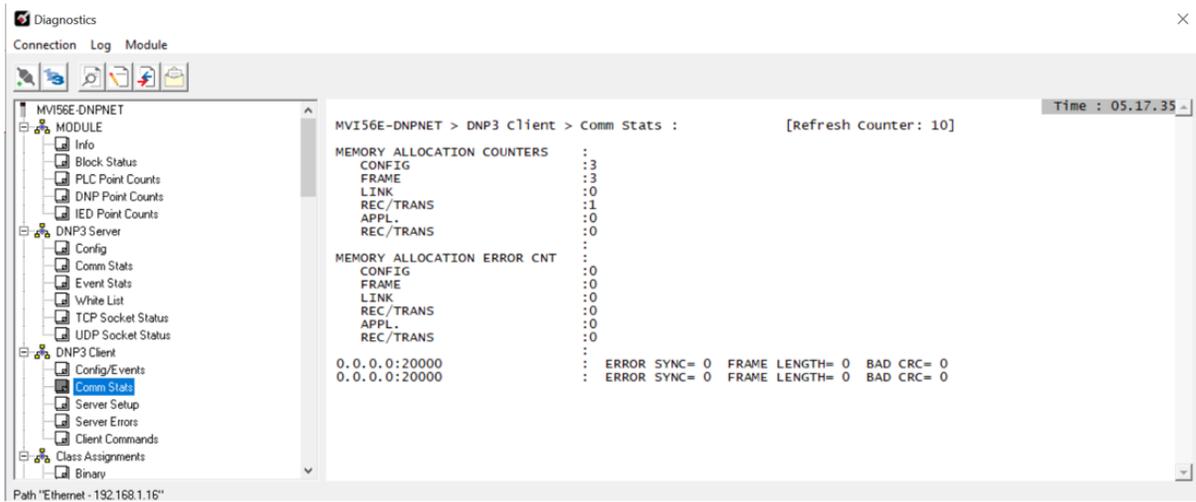
Parameter		Description
Memory Allocation Counters	Config	Number of memory blocks allocated.
	Frame	Number of frame blocks allocated.
	Link	Number of data-link layer blocks allocated.
	Rec/Trans	Number of total receive and transmit data-link layer blocks allocated.
	Appl.	Number of application layer blocks allocated.
	Rec/Trans	Number of total receive and transmit application layer blocks allocated.
Memory Allocation Error Count	Config	Number of memory block allocation errors.
	Frame	Number of frame block allocation errors.
	Link	Number of data-link layer block allocation errors.
	Rec/Trans	Number of total receive and transmit data-link layer block allocation errors.
	Appl.	Number of application layer block allocation errors.
	Rec/Trans	Number of total receive and transmit application layer block allocation errors.
IP Address : Port	Error Sync	Data packet status: **0** = Data packets are in sync. **1** = Data packets are not in sync.
	Frame Length	Length of data frame status: **0** = No error detected. **1** = The length of data is a mismatched, or an overflow has occurred.
	Bad CRC	CRC status: **0** = No error detected. **1** = The received message has a CRC mismatch.

DNP Client - Server Setup

This option displays information on the configured DNP3 Servers within the *DNP3_Server_List* controller tag.
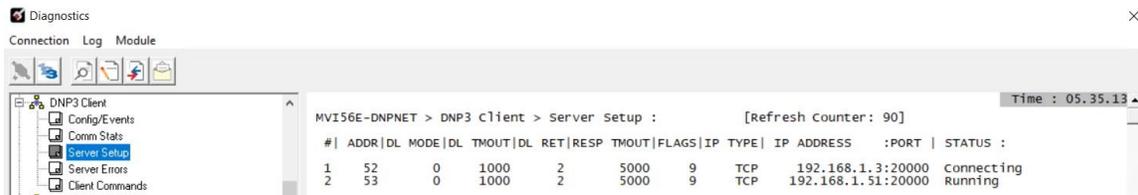
Parameter	Description
ADDR	Server Address
DL MODE	Data Link confirm mode
DL TMOUT	Data Link confirm timeout, in ms
DL RET	Data Link Retries
RESP TMOUT	Application layer response timeout, in ms
Flags	Bit values: **Bit 0** = Enabled / Polled **Bit 1** = Unsolicited **Bit 2** = USE_DM **Bit 3** = AUTO_TIME_SYNC **Bit 4** = Serial or LAN time sync
IP Type	**0** = TCP **1** = UDP
IP Address	IP address of server
Port	Port number
Status	**Connecting** = Client is trying to connect to server, connection not yet established **Running** = Client and Server are connected, connection established

DNP Client - Server Errors

This option displays the errors for the configured DNP3 Servers within the *DNP3_Server_List* controller tag.

Parameter	Description
ADDR	Configured node address for the DNP3 server.
BAD CRC	The number of bad CRC values received by the DNP3 server.
BUF OVRFLW	**Buffer Overflow** - The number of messages received that are greater than the application layer size limit for the DNP3 server.
TRANS SEQ	**Transaction Sequence Number** - The number of incorrect transport layer sequence number errors for the DNP3 server.
CONF RET	**Confirm Retries** - The number of data link layer confirm request retries for the DNP3 server.
CONF FAIL	**Confirm Failures** - The number of data link layer confirm request failures for the DNP3 server.
No RESP	**No Response** - The number of application layer 'no responses' for the DNP3 Server. This value will not increase if there is not a connection between the DNP3 Client and Server. Check STATUS under *DNP3 Client > Server Setup* for the connection status between the Client and Server.

DNP Client - Client Commands

This option displays information on all configured MVI56E-DNP client commands.
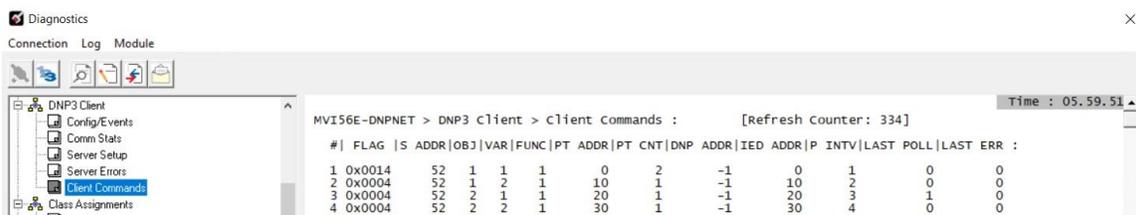
Parameter	Description
FLAG	Port flag
S ADDR	Server address
OBJ	Object utilized
VAR	Variation utilized
FUNC	Function utilized
PT ADDR	Starting point in server
PT CNT	Number of points requested in server
DNP ADDR	Starting point in DNP database
IED ADDR	Starting point in IED database
P INTV	Poll Interval, in seconds
LAST POLL	Time since command last polled, in seconds
LAST ERR	Last error code reported

3.6.7 Monitoring MVI56E-DNPNET Class Assignments Information

Use the *DNP3 Class Assignments* menu to view the following class information for the MVI56E-DNPNET module:

* Binary
* 16-bit Analog
* 32-bit Analog
* Float
* Double Float
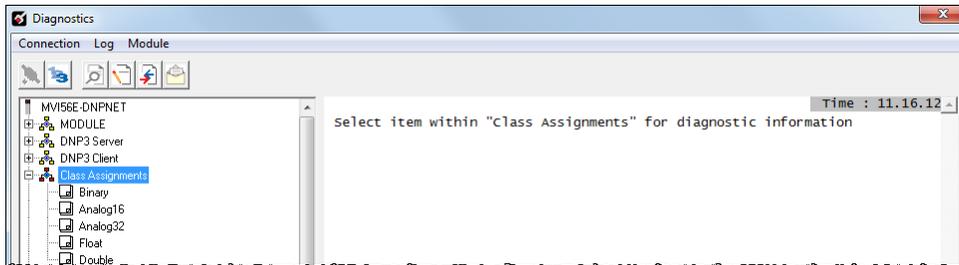
3.6.8 Monitoring MVI56E-DNPNET Deadband Assignments Information

Use the *DNP3 Deadband Assignments* menu to view the following deadband information for the MVI56E-DNPNET module:

* Binary
* 16-bit Analog
* 32-bit Analog
* Float
* Double Float
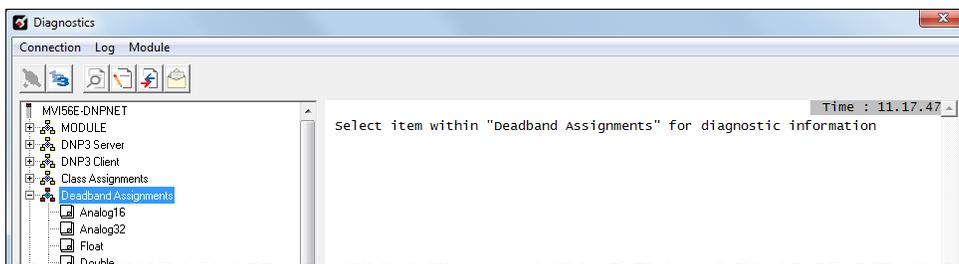
3.6.9 Monitoring DNP3 Ethernet Data Values

Use the ←*DNP* and ←*IED* menus to view the contents of the MVI56E-DNPNET module's internal database.

You can view Data values in Decimal or Hexadecimal format.
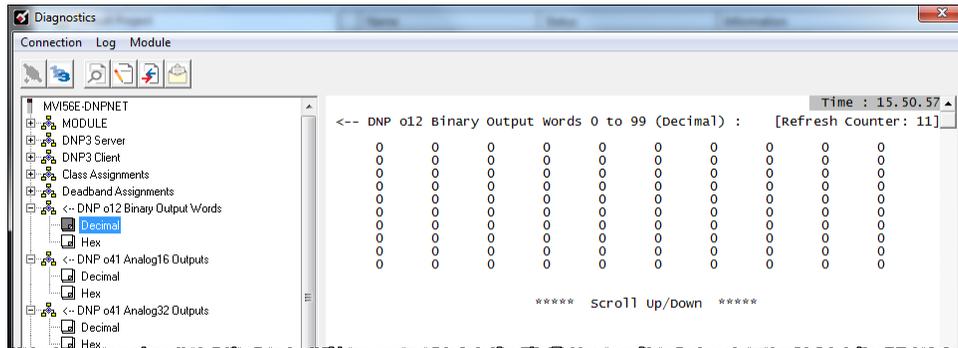
3.7   Communication Error Codes

3.7.1 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.
5	Command variation not supported	The variation specified in the command is not supported for the object type selected.
6	Object 10 not enabled	The command is not supported unless Binary Output Status is enabled
7	Object 40 not enabled	The command is not supported unless Analog Output Status is enabled
8	Invalid time/date poll command	This time/date object poll command is not valid.
9	Time/date poll command cannot execute	This time/date object poll command cannot execute because the module's clock has not been synced from a valid source (from PLC or connected Client)

3.7.2 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.

3.7.3 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 8000.
101	Too many binary output points	Too many binary output points are configured for the module. Maximum value is 8000.
102	Too many counter points	Too many counter points are configured for the module. Maximum value is 480.
103	Too many analog input points	Too many analog input points are configured for the module. Maximum value is 5000.
104	Too many analog output points	Too many analog output points are configured for the module. Maximum value is 5000.
105	Too many binary input events	Too many binary input events are configured for the module. Maximum value is 20000.
106	Too many analog input events	Too many analog input events are configured for the module. Maximum value is 20000.
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 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.
113	Too many server list entries	Too many server list entries are configured for the module. Maximum value is 40.
114	Too many client commands	Too many client commands are configured for the module. Maximum value is 300.
120	Too many PLC binary inputs	Too many PLC binary inputs are configured for the module. Maximum value is < Binary Input Count.
121	Too many PLC binary output	Too many PLC binary output are configured for the module. Maximum value is < Binary Output Count.
122	Too many PLC counters	Too many PLC counters are configured for the module. Maximum value is < DNP Counters.
123	Generic configuration error	Generic configuration error detected by the module.
124	Too many PLC analog outputs	Too many PLC analog outputs are configured for the module. Maximum value is < Analog Output Count.
140	Too many IED binary inputs	Too many IED binary inputs are configured for the module. Maximum value is 8000.
141	Too many IED binary outputs	Too many IED binary outputs are configured for the module. Maximum value is 8000.
142	Too many IED counters	Too many IED counters are configured for the module. Maximum value is 1000.
143	Too many IED analog inputs	Too many IED analog inputs are configured for the module. Maximum value is 20000.
144	Too many IED analog outputs	Too many IED analog outputs are configured for the module. Maximum value is 20000.

3.7.4 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.
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.

3.7.5 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.

3.8 Connect to the MVI56E-DNPNET Webpage

The module's internal web server provides access to module status, diagnostics, and firmware updates. If the module's IP address has already been assigned, simply enter in the IP address into a web browser. If not, follow the steps below:

1 In *ProSoft Configuration Builder*, click the **PROJECT** menu, then choose **MODULE > DOWNLOAD FROM PC TO DEVICE**. This opens the *Download* dialog box.

2 In the *Download* dialog box, choose the connection type in the *Select Connection Type* dropdown box:

- o Choose **ETHERNET** if you are connecting to the module through the Ethernet cable.
- o Choose **1756 ENBT** if you are connecting to the module through CIPconnect or RSWho.
  Refer to *Connecting Your PC to the Module* (page 45) for more information.

3 In the *Download files from PC to module* dialog box, click **BROWSE DEVICE(S)**.

4 In *ProSoft Discovery Service*, right-click the MVI56E-DNPNET icon and choose **VIEW MODULE'S WEBPAGE** from the shortcut menu.

This displays the module webpage.
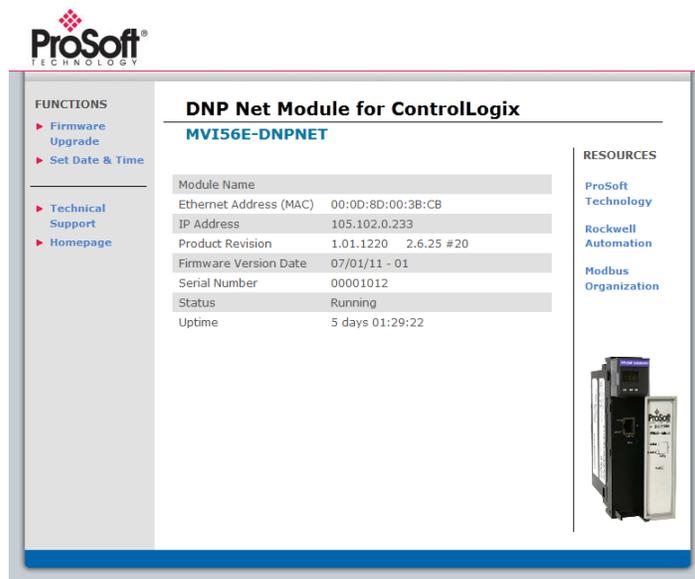
4   Reference

4.1   Product Specifications

The MVI56E-DNPNET (DNP3 Ethernet Client/Server Communication Module) allows Rockwell Automation ControlLogix I/O compatible processors to interface easily with other DNP3 Ethernet protocol compatible devices. The module supports DNP3 Ethernet Subset Level 2 features and some of the Level 3 features.

The module acts as an input/output communications module between the DNP3 Ethernet network and the ControlLogix backplane. The data transfer from the ControlLogix processor is asynchronous from the actions on the DNP3 Ethernet network. Databases are user-defined and stored in the module to hold the data required by the protocol.

This product features:

- ProSoft Configuration Builder (PCB): Microsoft Windows®-based utility software for diagnostics. Connect through the module's Ethernet port or use CIPconnect® to access troubleshooting features and functions.
- ProSoft Discovery Service (PDS): New Windows-based utility software to find and display a list of MVI56E modules on the network and to temporarily change a module's IP address to be able to connect with a module's webpage.
- CIPconnect-enabled: Allows PC-to-module diagnostics from the Ethernet network through a ControlLogix® 1756-ENxT EtherNet/IP™ module.
- LED Scrolling Diagnostic Display: 4-character, alphanumeric display, providing messages for status and alarm data, and for processor and network communication status.

4.1.1 General Specifications

- Single Slot - 1756 ControlLogix® backplane compatible
- 10/100 MB Ethernet port for network configuration and diagnostics with Auto Cable Crossover Detection
- Add-On Instruction (AOI) used for data transfers between module and processor and for module configuration
- User-definable module data memory mapping of thousands of DNP3 Ethernet points based on the various types of data:

DNP 3.0 Ethernet Data	Point Type	Range
DNP_Outputs	**Binary Outputs**	0 to 8000 points (500 16-bit words)
	Analog Outputs	
	16-bit Analog Outputs	0 to 5000 points (if all other DNP Outputs are 0)
	32-bit Analog Outputs	0 to 2500 points (if all other DNP Outputs are 0)
	Float Outputs	0 to 2500 points (if all other DNP Outputs are 0)
	Double Float Outputs	0 to 1250 points (if all other DNP Outputs are 0)
DNP_Inputs	**Binary Inputs**	0 to 8000 points ('500' 16-bit words)
	Analog Inputs	
	16-bit Analog Inputs	0 to 5000 points (if all other DNP Inputs are 0)
	32-bit Analog Inputs	0 to 2500 points (if all other DNP Inputs are 0)
	Float Inputs	0 to 2500 points (if all other DNP Inputs are 0)
	Double Float Inputs	0 to 1250 points (if all other DNP Inputs are 0)
	Counters	0 to 1000 points
IED_Outputs	**Binary Outputs**	0 to 8000 points (500 16-bit words)
	Analog Outputs	
	16-bit Analog Outputs	0 to 20000 points (if all other IED Outputs are 0)
	32-bit Analog Outputs	0 to 10000 points (if all other IED Outputs are 0)
	Float Outputs	0 to 10000 points (if all other IED Outputs are 0)
IED_Inputs	**Binary Inputs**	0 to 8000 points ('500' 16-bit words)
	Analog Inputs	
	16-bit Analog Inputs	0 to 20000 points (if all other IED Inputs are 0)
	32-bit Analog Inputs	0 to 10000 points (if all other IED Inputs are 0)
	Float Inputs	0 to 10000 points (if all other IED Inputs are 0)
	Counters	0 to 1000 points
	Binary Output Status	0 to 8000 points ('500' 16-bit words)
	16-bit Analog Output Status	0 to 20000 points
	32-bit Analog Output Status	0 to 10000 points
	Float Output Status	0 to 10000 points

4.1.2 Functional Specifications

The MVI56E-DNPNET operates on a Local or Remote rack CIPconnect® enabled for module and network configuration using 1756-ENxT module with EtherNet/IP pass-through communications.

- 4-digit LED Display for status and diagnostics information
- Error codes, network error counters, and port status data available in user data memory

Server Specifications

The DNP3 Ethernet port can accept DNP3 Ethernet commands to control and monitor data stored in the module's DNP3 Ethernet server database. If a DNP3 Ethernet Client is also configured, a portion of the server database can be derived from or can control IED devices connected to the DNP3 Ethernet Client.

- Report-by-Exception data is logged to the module's database
- Supports unsolicited messaging
- Each DNP3 Ethernet point type is user-configurable in the DNPNET Controller tags of RSLogix 5000 software
- Class assignments are user-definable on a Type and point basis (BI, AI, FI, DI point types)
- Supports clock synchronization from a remote Client or from the processor
- Up to 20,000 events are stored for Binary Inputs, Analog Inputs, Floats and Double Inputs. This varies based on point types in the table above.

Client Specifications

The DNP3 Ethernet port can be configured as a virtual DNP3 Ethernet Client device that actively issues user-defined DNP3 Ethernet commands to nodes on the network.

- The Module supports 300 user defined commands, each one containing its own set of data link and application layer characteristics
- Client logically supports up to 40 server devices
- Individual command configuration includes conditional or continuous polling and Poll Delay Time
- Server status and Command status available for transfer to the processor
- Event data received from the server devices updates the module database with the latest data values. Optionally date and time stamped data can be passed to the processor through a special block 9903 that is enabled/disabled through the parameter Pass Event Messages to PLC within the configuration of the module. When this option is used, events from the attached servers are passed to an array in the ladder logic containing the event data (server device, point index, point value) as well as the time stamp of the event from the attached server device (value is presented as the 64 bit UCT time matching the ControlLogix processor date/time format).
- Special command handling for Digital Output CROB under processor control for pulse output control

4.1.3 Hardware Specifications

Specification	Description
Backplane Current Load	800 mA @ 5 Vdc 3 mA @ 24 Vdc
Operating Temperature	0°C to 60°C (32°F to 140°F)
Storage Temperature	-40°C to 85°C (-40°F to 185°F)
Shock	30 g operational 50 g non-operational Vibration: 5 g from 10 to 150 Hz
Relative Humidity	5% to 95% (without condensation)
LED Indicators	Application Status (APP) Module Status (OK)
4-Character, Scrolling, Alpha-Numeric LED Display	Shows Module, Version, IP, Port Client/Server Setting, Port Status, and Error Information
Communication Ethernet Port	
Ethernet Port	10/100 Base-T, RJ45 Connector, for CAT5 cable Link and Activity LED indicators Auto-crossover cable detection

4.2   Functional Overview

4.2.1 MVI56E-DNPNET Backplane Data Exchange

General Concepts of MVI56E-DNPNET Data Transfer

Ladder logic is required for the MVI56E-DNPNET module to communicate along the backplane with the ControlLogix processor. The ladder logic handles the module data transfer, transfer of configuration data, 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.

For most applications, the sample Add-On Instruction (which includes the ladder logic) will work without modification.

The following topics describe several concepts that are important for understanding the operation of the MVI56E-DNPNET module.

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

- o   Initialize hardware components
- o   Initialize ControlLogix backplane driver
- o   Test and Clear all RAM

2   Reads configuration from the ControlLogix processor via ladder logic
3   Allocate and initialize Module Register space
4   Enable Client and Server Driver on Ethernet port
5   After the module has received the Module Configuration, the module will begin communicating with other nodes on the DNP3 Ethernet network, depending on the configuration.

Main Logic Loop

Upon completing the power up configuration process, the module enters an infinite loop performing the following functions:
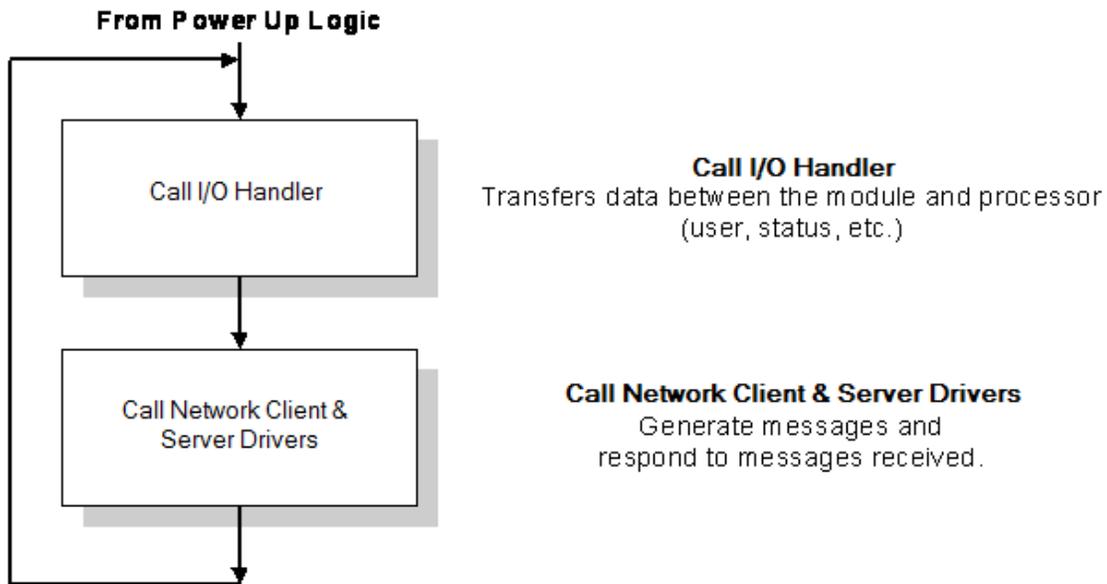
Backplane Data Transfer

The MVI56E-DNPNET module communicates directly over the ControlLogix backplane. Data is paged between the module and the ControlLogix 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 1 to 10 milliseconds per block of information.

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 is set to 250 words. This large data area permits fast throughput of data between the module and the processor.

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 is set to 248 words. This large data area permits fast throughput of data from the processor to the module.

The following illustration shows the data transfer method used to move data between the ControlLogix processor, the MVI56E-DNPNET module and the DNP3 Ethernet Network.
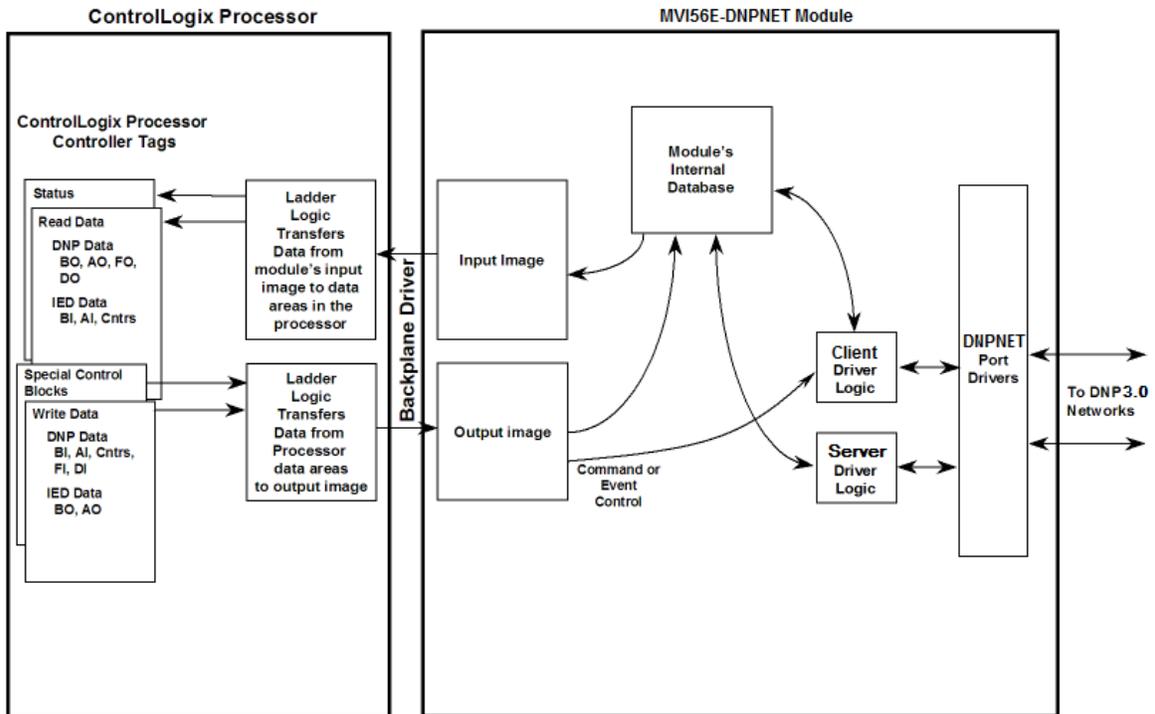
All data transferred between the module and the processor over the backplane is through the input and output images. Ladder logic is needed in the ControlLogix processor to interface the input and output image data with data defined in the Controller Tags. All data used by the module is stored in its internal databases. These databases are defined as a virtual DNPNET data tables with addresses from 0 to the maximum number of points for each data type.

Data Flow Between the DNP3 Ethernet network, MVI56E-DNPNET Module, and ControlLogix Processor

The following topics describe the flow of data between the two pieces of hardware (ControlLogix processor, and the MVI56E-DNPNET module) and other nodes on the DNP3 Ethernet network under the module's different operating modes.

The module is configured to emulate a DNP3 Ethernet Client device and/or a DNP3 Ethernet server device. The operation of each depends on your configuration. The following topics discuss the operation of each mode.

DNP3 Ethernet Server Backplane Data Flow

The Server Driver Mode allows the MVI56E-DNPNET module to respond to data read and write commands issued by a Client on the DNPNET network. The following flow chart and associated table describe the flow of data into and out of the module.

Step	Description
1	The DNPNET server driver configuration data is obtained from the DNPNET configuration tags via ladder logic. This configuration information contains data that can be used to offset data in the database to addresses requested in messages received from Clients.
2	A host device (DNP3 Ethernet Client) issues a read or write command to the module's node address. The MVI56E-DNPNET port driver qualifies the message before accepting it.
3	After the module accepts the command, the data is immediately transferred to or from the appropriate internal database in the module. If the command is a read command, the data is read out of the database and a response message is built. If the command is a write command, the data is written directly into the database and a response message is built.
4	After the data processing has been completed in Step 3, the response is issued to the originating Client node.
5	Counters are available in the Status Block to permit the ladder logic program to determine the level of activity of the Server Driver.

The response messages from the server driver include an IIN (internal indication word) defined in the Reference section.

The server driver supports object 110 (octet string data). Four points are pre-assigned values as defined in the following table.

Point Number	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.

The variation used in the request message determines the length of the string returned for each point. The maximum string length used by the module is 100.

DNP3 Ethernet Client Backplane Data Flow

In Client mode, the MVI56E-DNPNET module issues read or write commands to server devices on the DNP3 Ethernet network. These commands are user configured in the module via the Client Command List received from the ControlLogix processor or issued directly from the ControlLogix processor (Special Function).

Command status for each individual command is returned to the processor in the command list status block.

The following flow chart and associated table describe the flow of data into and out of the module.
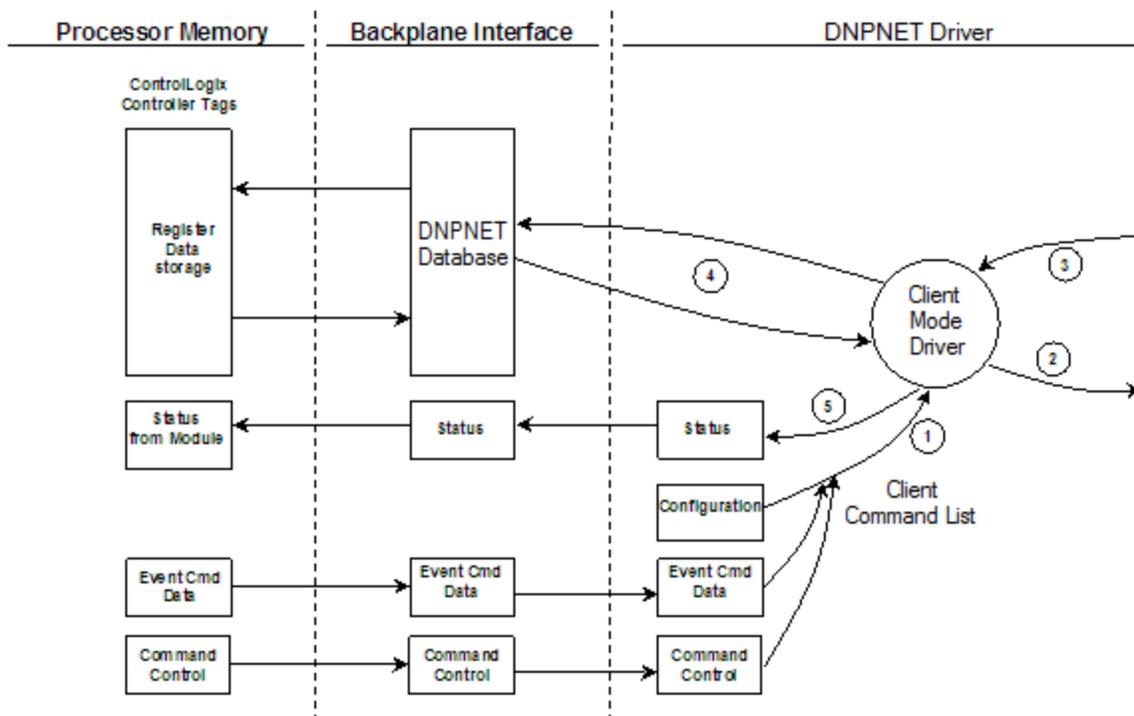
Step	Description
1	The Client driver configuration data is obtained from the DNPNET configuration tags via ladder logic. These values are used by the Client driver to determine the type of commands to be issued to the other nodes on the DNP3 Ethernet network.
2	After configuration, the Client driver 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 one of the module's internal databases to build the command.
3	Presuming successful processing by the node specified in the command, a response message is received into the Client driver for processing.
4	Data received from the node on the network is passed into the module's appropriate internal database, assuming a read command.
5	Status is returned to the ControlLogix processor for each command in the Client Command List.

4.2.2 Function Blocks

Data contained in this database is paged through the input and output images by coordination of the ControlLogix ladder logic and the MVI56E-DNPNET module's program. Up to 248 words of data can be transferred from the module to the processor at a time. Up to 247 words of data can be transferred from the processor to the module.

Each block transferred from the module to the processor or from the processor to the module contains a block identification code that describes the content of the block.

Block Number	Function/Description
0 or -1	Dummy Blocks: Used by module when no data is to be transferred
1 to 203	DNP and IED Data blocks
300	Error/Status, Error List Block, and slave IIN bits
1000 to 1022	DNP Output initialization blocks
1100 to 1193	IED Input initialization blocks
9000 to 9099	Configuration Data
9901	CROB Control Block for Digital Outputs
9902	Command Control Block (Adds command(s) to Command List Queue)
9903	Event Messages from Client port
9904	Places up to 24 Auxiliary Commands in the command queue.
9910	CROB Data received on DNPNET Port
9949	Server IED unit errors on Client port
9950	Command List Error data
9958	Binary Input Event data with Calendar time
9959	Analog Input Event data with Calendar time
9968	Binary Input Event data with CLX time
9969	Analog Input Event data with CLX time
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 and -1 are empty blocks used during module startup, when there is no data to transfer.

Blocks 1 to 203 are used to transfer the various kinds of DNPNET and IED process data. Block 300 transfers error and status data. Blocks 9901 to 9999 are used for Special Functions.

4.2.3 Module Function Blocks

Blocks 9000-9099 Configuration Data

The DNPNET configuration is requested from the PLC's ladder logic. The PLC will return a block 9000 with the section of the configuration data containing a fixed length. Within this block are the counts for client commands and this will determine how many subsequent blocks will be requested. Subsequent blocks can have a variable length configuration with blocks numbered 9001 and up (to a maximum block number of 9099).

Word Offset	DNPNET.Config. DNP3_Server.	Range	Description
1 to 20	DNP_Module_Name[0] to [39]	0 or 32 to 126	String of ASCII character bytes (up to 40) that gives the module a unique name. Terminate the string with a byte = 0. Module is named "MVI56E-DNPNET" by default.
21	Internal_Server_ID	0 to 32767	This is the DNP address for the module. All messages with this address from the client will be processed by the module.
22	Use_WhiteList	0 or 1	This parameter specifies if the IP address of the host connected to the system will be validated. If the parameter is set to 0, any host may connect to the unit. If the parameter is set to 1, only hosts in the IP list will be permitted to connect to the module. All other IP addresses will be ignored by the module and the module will issue a RST to the TCP/IP connection. The IP_List is contained in DNP.Config.DNP _ ENET_IP_Addresses.
23	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.
24	AI_Class	0 to 3	This parameter specifies the default class to be utilized for all the 16-bit analog input points in the DNP database that are not defined in the override list section.
25	A32I_Class	0 to 3	This parameter specifies the default class to be utilized for all the 32-bit analog input points in the DNP database that are not defined in the override list section.
26	Float_Class	0 to 3	This parameter specifies the default class to be utilized for all the float input points in the DNP database that are not defined in the override list section.
27	Double_Class	0 to 3	This parameter specifies the default class to be utilized for all the double input points in the DNP database that are not defined in the override list section.
28	AI_Deadband	0 to 32767	This parameter specifies the default deadband value assigned to all points not defined in the override list for the 16-bit analog input point type in the DNP database.
29	A32I_Deadband	0 to 2,147,483,647	This parameter specifies the default deadband value assigned to all points not defined in the override list for the 32-bit analog input point type in the DNP database.

Word Offset	DNPNET.Config. DNP3_Server.	Range	Description
31	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 float input point type in the DNP database.
33	Double_Deadband	0 to maximum double value	This parameter specifies the default deadband value assigned to all points not defined in the override list for the double input point type in the DNP database.
35	SelectOperate_ Arm_Time	1 to 32767	Time period, in milliseconds, 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.
36	Write_Time_Interval	0 to 1440	Time interval, in minutes, to set the need time IIN bit (0=never), which will cause the client to write the time.
37	Data_Link_Confirm_ Mode	Coded Value 0=Never 1=Sometimes 2=Always	IED can request acknowledgement from client station when sending data. Recommended value = 0.
38	Data_Link_Confirm_ Tout	1 to 32767	Time period, in milliseconds, to wait for client Data Link confirmation of last frame sent. This parameter is only used if the frame is sent with confirmation requested. Recommended value = 1000.
39	Data_Link_Max_Retry	0 to 255	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. Recommended value = 2.
40	App_Layer_ Confirm_Tout	1 to 32767 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.
41	Unsolicited_Response	0 or 1	If set to 0, the server will not send unsolicited responses. If set to 1, the server will send unsolicited responses.
42	Class_1_Unsol_ Resp_Min	1 to 255 events	Minimum number of events in Class 1 required before an unsolicited response will be generated.
43	Class_2_Unsol_ Resp_Min	1 to 255 events	Minimum number of events in Class 2 required before an unsolicited response will be generated.
44	Class_3_Unsol_ Resp_Min	1 to 255 events	Minimum number of events in Class 3 required before an unsolicited response will be generated.
45	Unsol_Resp_Delay	1 to 32767	Maximum number of milliseconds to wait after an event occurs before sending an unsolicited response message. If set to 0, only use minimum number of events.
46	UResp_Client_Address	0 to 255	DNP destination address where unsolicited response messages are sent.
47	AI_Events_with_time	0 or 1	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 0, the default is set to no time data. If the parameter is set to 1, the default object will include the time of the event.

Word Offset	DNPNET.Config. DNP3_Server.	Range	Description
48	Events_Require_ Time_Sync	0 or 1	This parameter is used to determine if events will be generated by the server module when its time is not synchronized from a client. If the parameter is set to 1, no events will be generated until the module's time has been synchronized. If the parameter is set to 0, events will always be generated.
49	Initialize_DNP_ Output_Database	0 or 1	This parameter determines if the module will request data from the processor to initialize the DNP database output data areas.
50	PassThrough_CROB	0 or 1	This parameter determines if the module will pass all received CROB messages received through to the processor. If it is set to 0 (default), then the messages will not be sent to the processor. If the parameter is set to 1, then block 9910 will be sent to the processor with the CROB information. The database will still be controlled by the CROB message, but the ladder can control other virtual BO data in the processor using this data. This feature is useful if the controlling station sends CROB data to the server driver with very short on or off times.
51	Use_TripClose_ Single_ Point	0 or 1	This parameter determines if data associated with CROB commands operate on a single or dual point. If the value of 0 is supplied (default value), then all points will be dual-point unless neither the trip or close bit is set in the control code of the command. If either bit is set, then the CROB block will interact with the bit database as a dual-point database. If the parameter is set to 1, then all CROB blocks received will operate on the database as single bits.
52	Unsol_Retry_Limit	7 to 32768	Configurable unsolicited retry limit. The module sends an unsolicited message and waits for a confirmation with the Application Layer Confirm Timeout up to the limit specified until the unsolicited message is confirmed. If the amount of unsolicited messages are exceeded, the Ethernet connection will be lost. Another DNP message could wake up the connection. The allowable limits are 7 to 32768.
53	Use_SOE_card	0 or 1	Using Allen Bradley 1756-SOE Sequence of Events Module. 0 = No, 1 = Yes
	Use_Data_from_Client _Connection	0 or 1	This parameter enables data from the MVI client connection to be shared with the MVI server database. 0 = No, 1 = Yes
	Use_Double_Floats	0 or 1	This parameter enables the DNP Double Float (64-bit) database

Word Offset	DNPNET.Config. DNP3_Server.	Range	Description
54	Block_Timeout_MS	11 to 5000 milliseconds	Backplane block heartbeat. If an out-of-range value is entered, the value will default to **1500**. For v1.30.063 and higher, the *Quality Flag Control* feature can be used. Please see *2.3.10 DNP3 Quality Flag Control* for more information. **Note:** If the *Block_Timeout_MS* < 10000, the *Quality Flags* are **enabled**. The module will report Quality Flags, along with DNP data. If the *Block_Timeout_MS* ≥ 10000, the *Quality Flags* are **disabled**. The module will force the Quality Flags to report only a status of **Online**, along with DNP data. The *Block_Timeout_MS* value and *Quality Flag Control* status can be checked at *PCB Diagnostics > DNP3 Server > Config*.
55	Server_Timeout	5 to 32767 seconds	Configurable TCP Server Connection timeout parameter. After the initial TCP connections has been made and there is no activity on the connection, the server will timeout and close the connection. If an out of range value is entered, the value will default to 5 seconds.
	Reserved_0		Reserved

4.2.4 Special Function Blocks

Special Function blocks are special blocks used to control the module or request special data from the module. The current version of the software supports several Special Function blocks.

Block 9901: CROB Control Block for Digital Output

If the ControlLogix processor sends a block 9901, the module places the digital output control commands to be sent to the server into the command queue of the modules client driver. Commands placed in the queue with this method are not contained in the normal command list. Data contained in the block completely defines the command to the system. The format for the block is as follows:

Word Offset in Block	Data Field(s)	Description
0	Block ID	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 24.
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.
…	…	…
232 to 241	Command #24	Data for the command relay block (CROB) to be generated.

The 10-word data area for each command is defined in the following table.

Word Offset	Definitions	Description
0	PortFlags	Clear 3rd bit to disable. Set 5th bit to select IED DB for write functions
1	Server Address	IED node address for the server to consider on the network.
2	Object	Object type always 12
3	Variation	Variation always 1
4	Function	Function codes 3 (select/operate), 5 (direct operate with ACK), and 6 (direct operate no ACK) supported. Function code 4 is automatically sent after a successful function 3.
5	Address in Server	Point in IED to consider with the CROB.
6	Control Code	This is a standard DNPNET 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. The CROB command is sent but no operation occurs since the *Operation Type* field is 'NUL'. 1 = Pulse on 2 = Pulse off 3 = Latch on 4 = Latch off. All other values are undefined in the DNPNET protocol.
4	Queue	0=Normal (execute once), 1=Re-queue (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.

The commands placed in the normal command list for the Client port does not provide the means for all the possible CROB operations. It only supports the latch on and off operations based on the status of the associated database value specified in the command. With the use of this block, outputs can be pulsed on or off for user specified count and time intervals. Additionally, this command provides support for trip/close relay control.

Block 9902: Command Control

If the ControlLogix processor sends a block 9902, the module will place the commands referenced in the block in the command queue. Commands placed in the queue with this method need not have their enable bit set. Only valid commands will be placed in the queue.

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 240.
2 to 241	Command Numbers to enable	These 240 words of data contain the command indices "**x**" in the *DNPNET.CONFIG.DNP_Client_Commands[**x**]* 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.
242 to 247	Spare	Not Used

Up to 240 commands can be enabled and placed in the command queue with one write request from the ControlLogix processor.

Note: There is no response to this block by the module. The module will place the selected commands into the command queue. If the command references an 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 9903: Event Message Block

When the DNPNET Client is configured to pass event messages from the port to the processor, block identification 9903 will be utilized. When the Client port receives an event message, it will place the data in the message into the event message queue. This queue has room for up to 1000 messages.

When the backplane task in the modules recognizes data in this queue, it will form 9903 blocks to transfer the data to the processor. Ladder logic extracts the event data from the 9903 block and places it in controller tags.

Word Offset in Block	Data Field(s)	Description
0	Reserved	
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 15 are valid.
3 to 18	Event 1	Event message
19 to 34	Event 2	Event message
35 to 50	Event 3	Event message
51 to 66	Event 4	Event message
67 to 82	Event 5	Event message
83 to 98	Event 6	Event message

Word Offset in Block	Data Field(s)	Description
99 to 114	Event 7	Event message
115 to130	Event 8	Event message
131 to 146	Event 9	Event message
147 to 162	Event 10	Event message
163 to 178	Event 11	Event message
179 to 194	Event 12	Event message
195 to 210	Event 13	Event message
211 to 226	Event 14	Event message
227 to 242	Event 15	Event message
243	Event Overflow	1 if module's 1000 count event buffer was full when this block was packed
244	Events Queued	Number of events still queued in module.
245 to 248	Spare	Not used
249	Block ID	This field contains the block identification code of 9903 for the block.

Up to 15 events are passed to the processor in each block. The format of each event message in the block is shown in the following table.

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	DNPNET Address	This field contains the DNPNET database index for the point. If set to -1, then not in database.
3	Server Address	This field contains the remote server 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	Reserved	Reserved for future use
8 to 9	Low Time	This field contains the least-significant double word of the 64-bit UTC time for the event.
10 to 11	High Time	This field contains the most- significant double word of the 64-bit UTC time for the event.
12 to 13	DINT Value	This field contains the double integer value for the point associated with the event message.
14 to 15	REAL Value	This field contains the double float point value for the point associated with the event message.

Block 9904: Auxiliary Client Commands Block

Block identification code 9904 is used to place up to 24 Auxiliary Commands in the command queue.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the block identification code of 9904 for the block.
1	Command Count	This field defines the number of commands contained in the block. The valid range for the field is 1 to 24.
2 to 11	Command #1	Data for command.
12 to 21	Command #2	Data for command.
...	…	…
232 to 241	Command #24	Data for command.
242 to 247	Reserved	Reserved for future use

The format of each message in the block is shown in the following table.

Word Offset	Definitions	Description
0	Port_Flag	Clear 3rd bit to disable. Set 5th bit to select IED DB for write functions
1	Server_Address	Address of server command is sent to
2	Object	Object number
3	Variation	Variation number
4	Function	Function number
5	Point_Number_In_Server	Starting point in server
6	Point count	Number of points in server
7	DNP_Point_Number_In_Client	Starting point number in DNP database to store data
8	IED_Point_Number_In_Client	Starting point number in IED database to store data
9	Reserved	-

Block 9910: CROB Data received on DNPNET Port

Block identification code 9910 is used to send CROB messages received on the DNPNET server port to the processor. For pulse or trip/close operations with a fast duration, this block can be used to pass the information into the ControlLogix processor. Additional ladder code can be written to perform the operation in locally in ladder code, making sure that a CROB message from an attached client is not missed due to database paging being too slow for the control operation.

Block Format for Read

Word Offset Start	Stop	Data Field(s)	Description
0	0	Reserved	
1	1	Block ID	This is the next block requested by the module.
2	2	CROB count	This field contains the number of CROB records that are contained in this block. The range is between 1 and 40.
3	8	CROB 1	CROB block data as defined below
9	14	CROB 2	CROB block data as defined below
15	20	CROB 3	CROB block data as defined below
21	26	CROB 4	CROB block data as defined below
27	32	CROB 5	CROB block data as defined below
33	38	CROB 6	CROB block data as defined below
39	44	CROB 7	CROB block data as defined below
45	50	CROB 8	CROB block data as defined below
51	56	CROB 9	CROB block data as defined below
57	62	CROB 10	CROB block data as defined below
63	68	CROB 11	CROB block data as defined below
69	74	CROB 12	CROB block data as defined below
75	80	CROB 13	CROB block data as defined below
81	86	CROB 14	CROB block data as defined below
87	92	CROB 15	CROB block data as defined below
93	98	CROB 16	CROB block data as defined below
99	104	CROB 17	CROB block data as defined below
105	110	CROB 18	CROB block data as defined below
111	116	CROB 19	CROB block data as defined below
117	122	CROB 20	CROB block data as defined below
123	128	CROB 21	CROB block data as defined below
129	134	CROB 22	CROB block data as defined below
135	140	CROB 23	CROB block data as defined below
141	146	CROB 24	CROB block data as defined below
147	152	CROB 25	CROB block data as defined below
153	158	CROB 26	CROB block data as defined below
159	164	CROB 27	CROB block data as defined below
165	170	CROB 28	CROB block data as defined below
171	176	CROB 29	CROB block data as defined below
177	182	CROB 30	CROB block data as defined below
183	188	CROB 31	CROB block data as defined below
189	194	CROB 32	CROB block data as defined below
195	200	CROB 33	CROB block data as defined below

Word Offset Start	Stop	Data Field(s)	Description
201	206	CROB 34	CROB block data as defined below
207	212	CROB 35	CROB block data as defined below
213	218	CROB 36	CROB block data as defined below
219	224	CROB 37	CROB block data as defined below
225	230	CROB 38	CROB block data as defined below
231	236	CROB 39	CROB block data as defined below
237	242	CROB 40	CROB block data as defined below
243	248	Spare	Not Used
249	249	Block ID	This field contains the block identification code of 9910 for the block.

The format of each 6 word data region in the block is as follows:

Word Offset	Definitions	Description
0	Point Number	This field contains the BO point number for the following CROB command.
1	Control Code	Byte value Control Code of the CROB message received. Control code 81 = trip, 41 = close, and 1 = pulse.
	Count	Byte value of number of pulses
2 to 3	On Time	This double-word contains the 'on' time interval for the CROB block.
4 to 5	Off Time	This double-word contains the 'off' time interval for the CROB block.

Block 9949: Request Server Communication Error Table

If the ControlLogix processor sends a block 9949, the MVI56E-DNPNET module responds with a server communication error listing.

These data values are updated after each command processed by the module. The block 9949 request is structured to retrieve data for up to 30 server units each call. The format of the block sent from the ControlLogix processor to the module is shown in the following table.

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 servers	This field contains the number of servers to report in the response message. The value has a range of 1 to 30.
2	Start Server Index	This parameter sets the index in the server array where to start. The first server in the array has a value of 0. The last index in the array has a value of (MaxServers -1).
3 to 247	Reserved	Reserved for future use

Using the data in this block, the module responds to the ControlLogix processor with a read block 9949 containing the requested server information in the following format:

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved (0)
1	Block ID	This is the next block requested by the module.
2	Server Count	This field contains the number of server records contained in the block that must be processed by the PLC. This field will have a value of 1 to 30.
3	Server Start Index	This field contains the index in the server array for the first record in the file. This field will have a value of 0 to (MaxServers-1).
4 to 11	Server Data #1	This is the server data for the first server in the block. The server index for the data is the Server Start Index given in word 3.
12 to 19	Server Data #2	This is the server data for the second server in the block.
20 to 27	Server Data #3	This is the server data for the third server in the block.
…	…	…
236 to 243	Server Data #30	Last server requested.
244 to 248	Spare	Not Used
249	Block ID	This field contains the value of 9949 identifying the block type to the PLC.

You can sequentially read through the list of all IED units up to 30 at a time to retrieve all the error information. This data can be transferred to the module's controller tag in the processor's ladder logic.

Below is the 8-word data area for each server.

Value	Description
Index	This value corresponds to the index in the device array for the server.
Server Address	This value corresponds to the DNP server address for the device.
Bad CRC	This value represents the number of bad CRC values received from the server device.
Buff Overflow	This value represents the number of buffer overflow messages received from the server device.
Transaction Sequence Number	This value represents the number of incorrect transport layer sequence number errors.
Confirm Retries	This value represents the number of data link layer confirm request retries.
Confirm Failures	This value represents the number of data link layer confirm request failures.
No Application Layer Responses	This value represents the number of application layer no responses to requests.

Block 9950: Read Command Error List

If the ControlLogix processor sends a block number of 9950 to the module, the application will respond with an MVI56E-DNPNET Client command error list. Each command in the system has a data word set aside for its last error code. This value is set by the DNPNET Client command list task and the values correspond to the errors listed in the error section of this documentation. This can be accessed 200 commands at a time.

Block format of Command Error List Request sent to the module from the ControlLogix processor.

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 200.
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 247	Spare	Not Used

Block format of Command Error List Response sent to the ControlLogix processor from module

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved
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 200.
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 203	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.
204 to 248	Spare	Not Used
249	Block ID	This field contains the value of 9950 identifying the block type to the PLC.

Block 9958: Binary Input Event With Calendar Time

Block identification code 9958 sends a set of binary input events with calendar time to the module. The following table lists the block format of Binary Input Event Request sent to the module from the ControlLogix processor.

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 24.
2	Sequence Counter	This field is used to hold the sequence counter for each 9958 block transfer. This is used to synchronize and confirm receipt of the block by the module.
	(Begin Event #1)	
3	Point_Number	Data point in the DNPNET binary input database represented by the event.
4	Class_ Override	Regardless of the Class defined for the given point number, this will be the Class of the event. (1, 2, or 3)
5	Value	Value of the point, 0 or 1, of bit 0 of this 16-bit word. All other bits are ignored.
6	Year	Year of the event timestamp
7	Month	Month of the event timestamp
8	Day	Day of the event timestamp
9	Hour	Hour of the event timestamp
10	Minute	Minute of the event timestamp
11	Seconds	Seconds of the event timestamp
12	Milliseconds	Milliseconds of the event timestamp
	(Next 23 Events)	
13 to 22		Ten words of data for Event #2.
…	…	…
233 to 242		Ten words of data for Event #24.
	(End of Event Data)	
243 to 247	Spare	Not used

Up to 24 events can be passed from the ControlLogix processor to the module in each block. To insure that the block reached the module and was processed, the module will send a response read block 9958 to the ControlLogix processor. The following table describes the format of the block.

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved
1	Block ID	Block identification code for request from PLC by the module.
2	Event Count Received	This field contains the number of events processed/received by the module. (1 to 24)
3	Sequence Counter Received	Sequence counter received for each block transfer. Used to synchronize and confirm receipt of the block by the module. This field contains the sequence counter of the last successful block 9958 received.
4 to 248	Spare	Not used
249	Block ID	Identification code for block set to 9958.

The sequence counter field in the returned block is set to the last successfully processed block 9958 from the ControlLogix processor. Compare this value to that sent by the ControlLogix processor.

If the values match, the events can be removed from the ControlLogix processor.

If the values do not match, or the ControlLogix processor does not receive a 9958 block, the ControlLogix processor must re-send the block.

Block 9959: Analog Input Event With Calendar Time

Block identification code 9959 sends a set of analog input events with calendar time to the module. The following table lists the block format of Analog Input Events sent to the module from the ControlLogix processor.

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 20.
2	Sequence Counter	This field is used to hold the sequence counter for each 9959 block transfer. This is used to synchronize and confirm receipt of the block by the module.
	(Begin Event #1)	
3	Point_Number	This is the data point in the DNPNET analog input database represented by the event.
4	Class_Override	Regardless of the Class defined for the given point number, this will be the Class of the event. (1, 2, or 3)
5 to 6	Value	Value of the point. 16-bit integer, 32-bit integer, or 32-bit float can be packed into this space
7	Year	Year of the event timestamp
8	Month	Month of the event timestamp
9	Day	Day of the event timestamp
10	Hour	Hour of the event timestamp
11	Minute	Minute of the event timestamp
12	Seconds	Seconds of the event timestamp
13	Milliseconds	Milliseconds of the event timestamp
14	Reserved	
	(Next 19 Events)	
15 to 26		Ten words of data for Event #2
…	…	…
231 to 242		Ten words of data for Event #20
	(End of Event data)	
243 to 247	Spare	Not Used

Up to 20 events can be passed from the ControlLogix processor to the module in each block. To insure that the block reached the module and was processed, the module will send a response read block 9959 to the ControlLogix processor.

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved(0)
1	Block ID	Block identification code for request from PLC by the module.
2	Event Count Received	This field contains the number of events processed/received by the module.
3	Sequence Counter Received	Sequence counter Received for each block transfer, used to synchronize and confirm receipt of the block by the module. This field contains the sequence counter of the last successful block 9959 received.
4 to 248	Spare	Not used
249	Block ID	Identification code for block set to 9959.

The sequence counter field in the returned block is set to the last successfully processed block 9959 from the ControlLogix processor. Compare this value to that sent by the ControlLogix processor.

If the values match, the events can be removed from the ControlLogix processor. If the values do not match, or the ControlLogix processor does not receive a 9959 block, the ControlLogix processor must re-send the block.

Block 9968: Binary Input Event With CLX Time

Block identification code 9968 sends a set of binary input events with ControlLogix processor time to the module.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9968 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 30.
2	Sequence Counter	This field is used to hold the sequence counter for each 9968 block transfer. This is used to synchronize and confirm receipt of the block by the module.
	(Begin Event #1)	
3	Point_Number	This is the data point in the DNPNET binary input database represented by the event.
4	Class_ Override	Bits 0 and 1 are used for class override values of 1, 2, or 3
5	Value	Value of the point, 0 or 1, of bit 0 of this 16-bit word. All other bits are ignored.
6	Reserved	
7 to 10	CLX_Time	64-bit Time as number of microseconds since Jan 1st, 1970
	(Next 29 Events)	
13 to 22		Ten words of data for Event #2
…	…	…
233 to 242		Ten words of data for Event #30
	(End of Event Data)	
243 to 247	Spare	Not used

Up to 30 events can be passed from the ControlLogix processor to the module in each block. To ensure that the block reached the module and was processed, the module will send a response read block 9968 to the ControlLogix processor.

The following table describes the format of the block.

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved (0)
1	Block ID	Block identification code for request from PLC by the module.
2	Event Count Received	This field contains the number of events processed/received by the module.
3	Sequence Counter Received	This field contains the sequence counter of the last successful block 9968 received. Used to synchronize and confirm receipt of the block by the module.
4 to 248	Spare	Not used
249	Block ID	Identification code for block set to 9968.

The sequence counter field in the returned block is set to the last successfully processed block 9968 from the ControlLogix processor. Compare this value to that sent by the ControlLogix processor.

If the values match, the events can be removed from the ControlLogix processor. If the values do not match, or the ControlLogix processor does not receive a 9968 block, the ControlLogix processor must re-send the block. This block is typically used to pass SOE data from a 1756-"SOE" module to the DNP server event buffer of the MVI56E-DNPNET module. Upon successful receipt of the data by the module, this data can then be cleared from the event queue of the 1756-"SOE" modules.

Block 9969: Analog Input Event With CLX Processor Time

Block identification code 9969 sends a set of analog input events with the ControlLogix processor time to the module. The following table lists the block format of Analog Input Events Request sent to the module from the ControlLogix processor.

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9969 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 30.
2	Sequence Counter	This field is used to hold the sequence counter for each 9969 block transfer. This is used to synchronize and confirm receipt of the block by the module.
	(Begin Event #1)	
3	Point_Number	This is the data point in the DNPNET analog input database represented by the event.
4	Class_Override	Regardless of the Class defined for the given point number, this will be the Class of the event. Bits 0 and 1 are used for class override values of 1, 2, or 3.
5 to 6	Value	Value of the point. 16-bit int, 32-bit int or 32-bit float can be packed into this space
7 to 10	CLX_Time	64-bit integer number of elapsed microseconds since Jan 1st, 1970
	(Next 29 Events)	
11 to 18	Month	Eight words of data for Event #2
…	…	...
235 to 242		Eight words of data for Event #30
	(End of Event data)	
243 to 247	Spare	Not Used

Up to 30 Events can be passed from the ControlLogix processor to the module in each block.

To insure that the block reached the module and was processed, the module will send a response read block 9969 to the ControlLogix processor.

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved
1	Block ID	Block identification code for request from PLC by the module.
2	Event Count Received	This field contains the number of events processed/received by the module.
3	Sequence Counter Received	This field contains the sequence counter of the last successful block 9969 received. Used to synchronize and confirm receipt of the block by the module.
4 to 248	Spare	Not used
249	Block ID	Identification code for block set to 9969.

The sequence counter field in the returned block is set to the last successfully processed block 9969 from the ControlLogix processor. Compare this value to that sent by the ControlLogix processor.

If the values match, the events can be removed from the ControlLogix processor. If the values do not match, or the ControlLogix processor does not receive a 9969 block, the ControlLogix processor must re-send the block.

Block 9970: Set CLX Processor Time From Module

This block transfers the module's time to the ControlLogix processor.

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 247	Not Used	Not Used

The module responds to the request with a read block 9970 with the following format.

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved
1	Block Write ID	This is the next block requested by the module.
2	Year	This field contains the four-digit year for 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 DNP3 Ethernet Client device on the module's server port. 1 = time has been set on the DNP3 Ethernet network. 0 = waiting for time sync from DNP3 Ethernet Client.
10 to 248	Not Used	Not Used
249	Block Read ID	This field contains the block identification code of 9970 for the block.

Block 9971: Set Module Time From CLX Processor

Block identification code 9971 passes the clock time in the ControlLogix processsor to the module. The date and time provided will be used to set the module's DNPNET clock.

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 for 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.
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 247	Not Used	Not Used

Block 9998: Warm Boot

If the ControlLogix processor sends a block number 9998, the module performs a warm-boot operation. The module will reconfigure the communication ports and reset the error and status counters.

Block 9999: Cold Boot

If the ControlLogix processor sends a block number 9999, the module performs a cold-boot operation. The firmware will reload the configuration file and reset all DNPNET memory, error and status data.

4.3 MVI56E-DNPNET Database Overview

	Output* Database (PLC <- DNPNET)		Input* Database (PLC -> DNPNET)
DNP OUTPUTS (from remote client to DNPNET Server)	BOCount BOPLC (BOIED)	**DNP INPUTS** (from DNPNET Server to remote client)	BICount BIPLC (BIIED)
	AOCount AOPLC (AOIED)		AICount AIPLC (AIIED)
	A32OCount A32OPLC (A32OIED)		A32ICount A32IPLC (A32IIED)
	FLTOCount FLTOPL (FLTOIED)		FLTICount FLTIPLC (FLTIIED)
	DBLOCount DBLOPLC (DBLOIED)		DBLICount DBLIPLC (DBLIIED)
IED INPUTS (from remote server to DNPNET Client)	BIIED		CCount CPLC (CIED)
	AIIED	**IED OUTPUTS** (from DNPNET Client to remote server)	BOIED
	A32IIED		AOIED
	FLTIIED		A32OIED
	CIIED		FLTOIED

* Between DNPNET module and PLC, with respect to DNPNET module

The diagram above shows how the DNPNET database is structured according to the configured point counts as named in firmware. Only the PLC and IED data sections are shared with the PLC.

The sections in parentheses are for data pass through only; not to be shared with the PLC. Data is shared with the PLC 240 words per block. Blocks are numbered 1 to 203. Block 1 transfers the first 240 words; block two transfers the next 240 words; etc., of PLC and IED data only.

The PLC and IED data are packed and unpacked into/from blocks contiguously. Block number assignments are independent of the point count assignments. Only the data with point counts that end in PLC and IED (not the ones in parentheses) get packed into blocks to be shared over the backplane with the PLC.

The block transfer logic transfers the database by packing blocks to their fullest payload until the end of the database is reached. There are no specific block number assignments to each variation.

This contiguous packing of PLC data necessitates a block numbering scheme that is not specific to the individual variations -- data blocks are numbered according to the 240 word (block payload) offset of the PLC and IED data to be shared with the PLC.

Block Description	Block ID Assignments
PLC and IED Database Transfer Blocks	1 through 203
Status Block	300
DNP Output Initialization Blocks	1000 through 1022
IED Input Initialization Blocks	1100 through 1193
Configuration Data	9000 through 9099

4.3.1 Normal Data Transfer

Normal data transfer includes the paging of the user data found in the module's internal databases between the module and the controller. These data are transferred through read (input image) and write (output image) blocks.

Refer to the Installing and Configuring the Module section for a description of the data objects used with the blocks and the ladder logic required. Each data block transferred between the module and the processor has a specific block identification code that defines the data type contained in the block.

The following table lists the block identification codes used for data transfer by the module.

DNP 3.0 Ethernet Data	Point Type	Range
DNP_Outputs	Binary Outputs	0 to 8000 points (500 16-bit words)
	16-bit Analog Outputs	0 to 5000 points (if all other DNP Outputs are 0)
	32-bit Analog Outputs	0 to 2500 points (if all other DNP Outputs are 0)
	Float Outputs	0 to 2500 points (if all other DNP Outputs are 0)
	Double Float Outputs	0 to 1250 points (if all other DNP Outputs are 0)
DNP_Inputs	Binary Inputs	0 to 8000 points ('500' 16-bit words)
	16-bit Analog Intputs	0 to 5000 points (if all other DNP Inputs are 0)
	32-bit Analog Intputs	0 to 2500 points (if all other DNP Inputs are 0)
	Float Inputs	0 to 2500 points (if all other DNP Inputs are 0)
	Double Float Inputs	0 to 1250 points (if all other DNP Inputs are 0)
	Counters	0 to 1000 points
IED_Outputs	Binary Outputs	0 to 8000 points (500 16-bit words)
	16-bit Analog Outputs	0 to 20000 points (if all other IED Outputs are 0)
	32-bit Analog Outputs	0 to 10000 points (if all other IED Outputs are 0)
	Float Outputs	0 to 10000 points (if all other IED Outputs are 0)
IED_Inputs	Binary Inputs	0 to 8000 points ('500' 16-bit words)
	16-bit Analog Inputs	0 to 20000 points (if all other IED Inputs are 0)
	32-bit Analog Inputs	0 to 10000 points (if all other IED Inputs are 0)
	Float Inputs	0 to 10000 points (if all other IED Inputs are 0)
	Counters	0 to 1000 points
	Binary Output Status	0 to 8000 points ('500' 16-bit words)
	16-bit Analog Output Status	0 to 20000 points
	32-bit Analog Output Status	0 to 10000 points
	Float Output Status	0 to 10000 points

The following illustration shows the direction of movement of these data types between the module and the processor.

	Output* Database (PLC <- DNPNET)		Input* Database (PLC -> DNPNET)
DNP OUTPUTS (from remote client to DNPNET Server)	BOCount BOPLC (BOIED)	**DNP INPUTS** (from DNPNET Server to remote client)	BICount BIPLC (BIIED)
	AOCount AOPLC (AOIED)		AICount AIPLC (AIIED)
	A32OCount A32OPLC (A32OIED)		A32ICount A32IPLC (A32IIED)
	FLTOCount FLTOPLC (FLTOIED)		FLTICount FLTIPLC (FLTIIED)
	DBLOCount DBLOPLC (DBLOIED)		DBLICount DBLIPLC (DBLIIED)
IED INPUTS (from remote server to DNPNET Client)	BIIED		CCount CPLC (CIED)
	AIIED	**IED OUTPUTS** (from DNPNET Client to remote server)	BOIED
	A32IIED		AOIED
	FLTIIED		A32OIED
	CIIED		FLTOIED

RSLogix 5000 CPU

DNPout Data
- Bin
- A16
- A32
- Float
- Double

IEDin Data
- Bin
- A16
- A32
- Float
- Counters
- Bin_Out_Stat
- A16_Out_Stat
- A32_Out_Stat
- FLT_Out_Stat

DNPin Data
- Bin
- A16
- A32
- Float
- Double
- Counters

IEDout Data
- Bin
- A16
- A32
- Float

Array / Data / Read

Local:I 0 to 249

Back Plane

Write / Data / Array

Local:O 0 to 247

MVI56E-DNPNET

Backplane Buffer

DNPout Server Data
Goes to PLC | stays in Module
- PLC Bin → Module Bin
- PLC A16 → Module A16
- PLC A32 → Module A32
- PLC FLT → Module FLT
- PLC DBL → Module DBL

IEDin Client Data
Goes to PLC | Goes to DNP Server
- Bin
- A16
- A32
- FLT
- Counters
- Bin_Out_Stat
- A16_Out_Stat
- A32_Out_Stat
- FLT_Out_Stat

DNPin Server Data
Comes from PLC | stays in Module
- PLC Bin → Module Bin
- PLC A16 → Module A16
- PLC A32 → Module A32
- PLC FLT → Module FLT
- PLC DBL → Module DBL
- PLC CNTR → Module CNTR

Events are generated this server

IEDout Client Data
Comes from PLC | Comes from DNP Server
- Bin
  ---- to Bin (not Cast) --->
- A16
  -----Or Cast to A16---->
- A32
  -----Or Cast to A32---->
- FLT
  -----Or Cast to FLT---->

From remote DNP Client to Module's DNP Server then from Module's IED Client to Module's DNP Server above PLC data

From remote Server by Module's IED Client to Module's DNP Server above PLC data

Remote Client writes data here to the Module's Server

Module's Client writes Data here as read from Remote Server

Remote Client reads data from here the Module's Server

Module's Client reads data from here and writes it to the Remote Server

Module's Client reads Data from here and writes it to the Remote Server

Ethernet Port

Backplane Buffer

Field DNP Servers and Clients
- IED 1 Server
- IED 2 Server
- Hub/Switch
- DNP 1 Client
- DNP 2 Client

Module Data Objects

These objects hold process and status data values. All supported DNPNET data types have their own UDTs and controller tags. This makes it easier to identify and use the various data types.

Read Block

READ Blocks transfer information from the module to the ControlLogix processor. The following table describes the basic block structure of an input image.

Block Offset	Content
0	Reserved
1	Write block ID
2 to 241	Read data
242 to 248	Spare (Not used)
249	Read block ID

The *Read Block ID* is an index value used to determine the location of where the data will be placed in the ControlLogix processor read data controller tag array. Each transfer can move up to 240 words (block offsets 2 to 241) of data. The value of the Read Block identification code identifies the type of data contained in the block, so the sample ladder logic can move it to the correct controller tag array.

The *Write Block ID* contained in the Read Block tells the ladder logic which block of data the module is expecting to receive from the ControlLogix processor during the next backplane transfer. Under normal program operation, the module sequentially sends read blocks and requests write blocks. For example, if one block each of binary and analog output data, one block of binary input data, two blocks of counter data and two blocks of analog input data are used with the application, the backplane transfer sequence block numbers will be:

R4W0→ R16W8→ R4W9→ R16W12→ R4W13→ R16W0→ R4W8→

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 DNPNET network or operator control through the module's Configuration/Debug port. This sequence is occasionally interrupted by the read block identification code 100. This block passes the error/status and error list information from the module to the processor. Refer to the Error/Status section of this document for the structure and data contained in a Status Read block.

Write Block

WRITE blocks transfer information from the ControlLogix processor to the module. The following table describes the structure of a typical output image Write Block.

Block Offset	Content
0	Write block ID
1 to 240	Write data
241 to 247	Spare (Not used)

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 240 words (block offsets 1 to 240) of data.

In cases where the ladder logic uses Special Function Blocks, the normal *Read Block IDs* and *Write Block IDs* will be replaced with a *Special Function Block ID*. Once the Special Function has been processed, the module will resume the normal data *Read Block ID* and *Write Block ID* sequence, starting from where the sequence was interrupted.

Trip/Close

The MVI56E-DNPNET module supports Trip/Close functionality for the DNP Binary Output points when operating as a server.

This allows Trip/Close commands to be sent to the MVI56E-DNPNET module as a server, for dual point control. Each DNPNET Trip/Close command will occupy 2 bits within the module memory.

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

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

DNPNET BO Database Point	BO Latch/Pulse Point	BO Trip/Close Point
0	BO 0	Close BO 0
1	BO 1	Trip BO 0
2	BO 2	Close BO 1
3	BO 3	Trip BO 1
4	BO 4	Close BO 2
5	BO 5	Trip BO 2
100	BO 100	Close BO 50
101	BO 101	Trip BO 50
1000	BO 1000	Close BO 500
1001	BO 1001	Trip BO 500
2000	BO 2000	Close BO 1000
2001	BO 2001	Trip BO 1001
3000	BO 3000	Close BO 1500
3001	BO 3001	Trip BO 1501
...and so on...		
7998	BO 7998	Close BO 3998
7999	BO 7999	Trip BO 3999

The trip/close values require 2 points within the module's DNPNET database. A Trip is represented by the binary value of '10' for those 2 points, and a Close is represented by the binary value of '01' for those same 2 points.

The module can process only 4000 trip/close dual points, as the database for the DNPNET BO is limited to 8000 total bits.

The dual point control on trip/close operations can be overwritten by selecting the parameter of "Use_TripClose_Single_Point" in the server configuration section of the module. When this parameter is enabled, all trip and close operations will be treated as pulse on commands, and only occupy a single boolean point.

4.3.2 DNPNETModuleDef Object

This object contains all the MVI56E-DNPNET module top-level data.

Name	Description
CONFIG	DNPNET module configuration parameters for blocks 9000 - 9099
DATA	Client and server data transferred between the processor and the module
STATUS	Status for various functionalities
CONTROL	Governs the data movement between the PLC rack and the module
UTIL	Generic tags used for internal ladder processing (DO NOT MODIFY)

4.3.3 DNPNETCONFIG Object

This object contains the data types that apply to the configuration of the module.

Name	Description
DNP_Module_Name	Module name description
DNP3_Server	Server configuration
DNP3_WhiteList	IP addresses of the Clients that the MVI56E-DNPNET Server will respond to.
DNP3_Client	Client configuration
DNP_Server_Override	Assigning Class and Deadband overrides to individual point types.
DNP_Server_List	List of servers the MVI56E-DNPNET Client connects to
DNP_Client_Commands	List of commands the MVI56E-DNPNET Client sends to servers
IP_Settings	IP Settings of module

4.3.4 DNPNETCONTROL Object

This object contains values that are a 'scratchpad' area of intermediate data storage variables. They are used by the AOI to keep track of various logic processing functions.

Name	Variable	Description
Events		Handling the transfer of events from the Events Message Buffer in the Client to the PLC.
EventMessages		Events Captured by DNP Client are moved to this structure.
CROBmsg_counter		CROB message data copied via block 9910 from DNP Server. CROBmsg_counter tags are status data used when MVI56E-DNPNET is configured as a server and passes the CROB message counter information to the sub CROBmsg_counter tags through the backplane.
	CROBmsg_counter.OneShot	Internally used in Add On Instruction (not designed to be used by the end user.
	CROBmsg_counter.CountPerBlock	Number of CROB command message(s) that was (were) contained in the last 9910 Backplane Block received in the Add On Instruction.
	CROBmsg_counter.CCC	Continuous CROB message Count. Everytime CROB message is received by the server, this counter will increment by 1. This may be reset by the user. Otherwise, it will be reset to 0 on the 2147483646th CROB message coming into the server by the Add On Instruction.
	CROBmsg_counter.CBCC	Continuous CROB message Block Count. Everytime 9910 block is received by the Add On Instruction, this counter increments by 1. This may be reset by the user. Otherwise it will be reset to 0 on the 2147483646th CROB message coming into the server by the Add On Instruction.
CROBmsg_data		CROB message data copied via block 9910 from DNPServer. CROBmsg_data tags are status data used when the MVI56E-DNPNET is configured as a server and passes the CROB message counter information to the sub CROBmsg_counter tags through the backplane.
	CROBmsg_data.Point_Number	This parameter was specified by the DNP client as the Control Relay Output Block point in the MVI56E-DNPNET (configured as a server).
	CROBmsg_data.Control_Code	This is a standard DNPNET protocol control code byte (see description on page 98).
	CROBmsg_data.count	This parameter was specified by the DNP3 client of the number of pulses to generate for pulse output control in the CROB command.

	CROBmsg_data.On_Time	This parameter was specified by the DNP3 client of the on-time interval for pulse control in the CROB command.
	CROBmsg_data.Off_Time	This parameter was specified by the DNP3 client of the off-time interval for pulse control in the CROB command.
Binary_Event_CalTime		For storing block 9958 Binary Event Messages to be sent to DNP Server.
Analog_Event_CalTime		For storing block 9959 Analog Event Messages to be sent to DNP Server.
Binary_Event_CLXTime		For storing Block 9968 Binary Input Events with CLX Time.
Analog_Event_CLXTime		For storing Block 9969 Binary Input Events with CLX Time.
Get_Module_Time		Reads the module time into the PLC.
Set_Module_Time		Sends the PLC time to the module.
CROB_Commands		9901 block of CROB commands.
Aux_Commands		9904 block for Auxiliary Client commands.
CommandControl		9902 block of commands to enable.
	CommandControl.Send	Triggers a command control block 9902.
	CommandControl.NumberofCommands	Total number of commands to enable.
	CommandControl.ClientCommands	List of command indices "**x**" *DNPNET.CONFIG.DNP_Client_Commands[**x**]* client commands to be sent.
Server_Comm_Errors		Count and offset of servers to report.
Command_Errors		Count and offset of commands to report.
ColdBoot		Cold boot the module via block 9999.
WarmBoot		Warm boot the module via block 9998.
ChangeDatabaseSize		Triggers database size change.

4.3.5 DNPNETDATA Object

This object stores all the process-related data for a MVI56E-DNPNET module. This includes data for the primary DNPNET server port (DNPNET data set) and the data received from or sent to DNP3 Ethernet server devices (IED data set) by the secondary DNPNET port when configured as a DNP3 Ethernet Client.

Contained within this data object is an array for each possible data type. The array sizes are set to match the maximum possible module configuration. If multiple MVI56E-DNPNET modules are used within a rack, a copy of this structure will have to be made to permit each module to have its own databases.

Each data type has its own set of unique block identification codes to distinguish the data contained in the read or write block.

Name	Description
DNPout	Module's Server Database. Remote Clients Write into this DB.
IEDin	Module's Client Database. The Client reads data from remote Servers and writes it to the DB.
DNPin	Module's Server Database. Remote Clients Read from this DB.
IEDout	Module's Client Database. The Client reads data from this DB and writes it to remote Servers.

4.3.6 DNPNETSTATUS Object

This object stores all status information of the module including client and server status.

Name	Description
GenStat	General status information
ErrList	List of last 60 DNPNET module errors
IINServerBits	IIN Bits received from Servers
ServerCommErrors	Server communication errors list
CommandErrors	Command errors list

4.3.7 DNPNETUTIL Object

This object contains variables for internal AOI usage and should not be accessed by user application.

Name	Description
LastRead	Index of last read block
LastWrite	Index of last write block
BlockIndex	Computed block offset for data table
ReadData	Buffer File for data Read from Module
WriteData	Buffer File for data Written to Module
OffsetCounter	For calculating buffer copy offset
CopyLength	For calculating buffer copy length
LastService	For keeping track of last special outgoing block serviced
Mutex	Only one special block gets serviced every other scan
OddScan	Keep track of every other scan
FirstRun	Ensures database counts are correct every time the PLC restarts
DNP_List_Entry_Counts	Lengths of the seven DNPNET Client List Data
IED_Database	Configure the Client database sizes
DNP_Server_Database	Configured Server database sizes

4.4   MVI56E-DNPNET User Defined Data Types

Several UDTs are defined in the MVI56E-DNPNET Add-On Instruction.

The main UDT, DNPNETMODULEDEF, contains all the data types for the module. It is used to create the main controller tag structure, DNPNET.

There are five UDTs one level below DNPNETMODULEDEF. These lower-level UDTs were used to create the DNPNET.CONFIG, DNPNET.DATA, DNPNET.STATUS, DNPNET.CONTROL, and DNPNET.UTIL controller tag structures.

Name:	DNPNETMODULEDEF
Description:	This defines the entire module which includes all tags used in the program

Members: Data Type Size: 173268

	Name	Data Type	Style	Description
⊞	CONFIG	DNPNETCONFIG		DNPNET module configuration parameters for blocks 9000 - 9099
⊞	DATA	DNPNETDATA		Server and Client Databases
⊞	STATUS	DNPNETSTATUS		Status Tags for various functionalities
⊞	CONTROL	DNPNETCONTROL		Tags for Controling module's functionality
⊞	UTIL	DNPNETUTIL		Tags that perform background tasks

Click the **[+]** signs to expand the UDT structures and view lower-level UDTs. For example, when DATA is expanded, it contains four UDTs: DNPout, IEDin, DNPin, and IEDout. They can further be expanded with the **[+]** sign.

Name	Data Type	Style	Description
⊞ CONFIG	DNPNETCONFIG		DNPNET module configuration parameters for blocks 9000 - 9099
⊟ DATA	DNPNETDATA		Server and Client Databases
⊞ DNP_Outputs	DNPNET_DNP_Output_Database		Module's Server Database. Remote Clients Write into this DB.
⊞ IED_Inputs	DNPNET_IED_Input_Database		Module's Client Database. The Client writes data into this DB that it Reads from remote Servers
⊞ DNP_Inputs	DNPNET_DNP_Input_Database		Module's Server Database. Remote Clients Read from this DB.
⊞ IED_Outputs	DNPNET_IED_Output_Database		Module's Client Database. The Client reads data from this DB and Writews it to remote Servers.
⊞ STATUS	DNPNETSTATUS		Status Tags for various functionalities
⊞ CONTROL	DNPNETCONTROL		Tags for Controling module's functionality
⊞ UTIL	DNPNETUTIL		Tags that perform background tasks

Notice these UDTs are the data types used to declare the DNPNET.DATA controller tag arrays.

4.4.1 DNPNET.CONFIG Controller Tags

This UDT structure contains the data types that apply to the configuration of the module. Refer to DNPNET Controller Tags Definitions (page 23) for a complete description of each element in this object.

Name	Data Type	Description
DNP_Module_Name	SINT[40]	Module Definition
DNP3_Server	DNPNET_Server_Parameters	Server configuration
DNP3_WhiteList	DNPNET_Whitelist_IP_Address[10]	IP addresses of clients server will respond to
DNP3_Client	DNPNET_Client_Parameters	Client configuration
DNP3_Server_Override	DNPNET_Override	Assigning Class and Deadband Override to individual point types.
DNP_Server_List	DNPNET_Server_List[5]	List of servers the Client connects to
DNP_Client_Commands	DNPNET_Client_Commands[x]	List of commands the Client sends to servers
IP_Settings	DNPNET_Module_IP_Addressing	IP Settings of module

4.4.2 DNPNET.CONFIG.Override Controller Tags

This UDT structure contains the data types that apply to the configuration of the module. Refer to DNPNET Controller Tags Definitions (page 23) for a complete description of each element in this object.

DNP_Server_Binary_Inputs	DNPNET_BinIn_Class_Override_type[10]	Assigning classes to individual points
DNP_Server_Analog16_Inputs	DNPNET_A16in_ClassDeadband_Override_type[10]	Assigning class/deadbands to individual points
DNP_Server_Analog32_Inputs	DNPNET_A32in_ClassDeadband_Override_type[10]	Assigning class/deadbands to individual points
DNP_Server_Float_Inputs	DNPNET_FLTin_ClassDeadband_Override_type[10]	Assigning class/deadbands to individual points
DNP_Server_Double_Inputs	DNPNET_DBLin_ClassDeadband_Override_type[10]	Assigning class/deadbands to individual points

4.4.3 DNPNET.DATA Controller Tags

Name	Tag Name	Range	Description
DNP_Outputs	DNPNET.DATA.DNP_Outputs.Binary	0 to 8000 points (500 16-bit words)	Object 12 binary INTs from module's server database
	DNPNET.DATA.DNP_Outputs.Analog16	0 to 5000 points (if all other DNP_Outputs are 0)	Object 41 analog INTs from module's server database
	DNPNET.DATA.DNP_Outputs.Analog32	0 to 2500 points (if all other DNP_Outputs are 0)	Object 41 analog DINTs from module's server database
	DNPNET.DATA.DNP_Outputs.Float	0 to 2500 points (if all other DNP_Outputs are 0)	Object 41 analog REALs from module's server database
	DNPNET.DATA.DNP_Outputs.Double	0 to 1250 points (if all other DNP_Outputs are 0)	Object 41 analog double precision floats from module's server database
DNP_Inputs	DNPNET.DATA.DNP_Inputs.Binary	0 to 8000 points ('500' 16-bit words)	Object 1 binary INTs for module's server database
	DNPNET.DATA.DNP_Inputs.Analog16	0 to 5000 points (if all other DNP_Inputs are 0)	Object 30 analog INTs for module's server database
	DNPNET.DATA.DNP_Inputs.Analog32	0 to 2500 points (if all other DNP_Inputs are 0)	Object 30 analog DINTs for module's server database
	DNPNET.DATA.DNP_Inputs.Float	0 to 2500 points (if all other DNP_Inputs are 0)	Object 30 analog REALs for module's server database
	DNPNET.DATA.DNP_Inputs.Double	0 to 1250 points (if all other DNP_Inputs are 0)	Object 30 analog double precision floats for module's server database
	DNPNET.DATA.DNP_Inputs.Counter	0 to 1000 points	Object 20 counter DINTs for module's server database
IED_Outputs	DNPNET.DATA.IED_Outputs.Binary	0 to 8000 points (500 16-bit words)	Object 12 binary INTs for module's client database
	DNPNET.DATA.IED_Outputs.Analog16	0 to 20000 points (if all other IED_Outputs are 0)	Object 41 analog INTs for module's client database
	DNPNET.DATA.IED_Outputs.Analog32	0 to 10000 points (if all other IED_Outputs are 0)	Object 41 analog DINTs for module's client database
	DNPNET.DATA.IED_Outputs.Float	0 to 10000 points (if all other IED_Outputs are 0)	Object 41 REALs for module's client database
IED_Inputs	DNPNET.DATA.IED_Inputs.Binary	0 to 8000 points ('500' 16-bit words)	Object 1 binary INTs from module's client database
	DNPNET.DATA.IED_Inputs.Analog16	0 to 20000 points (if all other IED_Inputs are 0)	Object 30 analog INTs from module's client database
	DNPNET.DATA.IED_Inputs.Analog32	0 to 10000 points (if all other IED_Inputs are 0)	Object 30 analog DINTs from module's client database
	DNPNET.DATA.IED_Inputs.Float	0 to 10000 points (if all other IED_Inputs are 0)	Object 30 analog REALs from module's client database
	DNPNET.DATA.IED_Inputs.Counter	0 to 1000 points	Object 20 Counter DINTs from module's client database
	DNPNET.DATA.IED_Inputs.Binary_Out_Status	0 to 8000 points ('500' 16-bit words)	Object 10 INTs output status from module's client database
	DNPNET.DATA.IED_Inputs.Analog16_Out_Status	0 to 20000 points	Object 40 INTs output status from client's database
	DNPNET.DATA.IED_Inputs.Analog32_Out_Status	0 to 10000 points	Object 40 analog DINTs. Output status from client's database
	DNPNET.DATA.IED_Inputs.Float_Out_Status	0 to 10000 points	Object 40 analog REALs output status from client's database

4.4.4 DNPNET.STATUS Controller Tags

This status data is returned on each read block and can be used to detect proper module operation.

Name	Description
GenStat	Contains general status information including error counts, block errors.
ErrorList	List of last 60 MVI56E-DNPNET module errors.
IINServerBits	IIN Bits received from Servers.
ServerCommErrors	Server communication errors list.
CommandErrors	Command errors list.
Network	Server TCP/UDP network status.

DNPNET.STATUS.GenStat

This array contains MVI56E-DNPNET general status information.

Name	Description
GenStat.Cur_Port	Current DNP Slave port.
GenStat.Last_Err	Last DNP slave error code.
GenStat.Msg_Me	Total number of frames sent and received by the module.
GenStat.Msg_Sent	Total message frames sent by the module.
GenStat.Msg_Rec	Total message frames received by the module.
GenStat.Err_Sync	Total number of DNP physical layer synchronization errors.
GenStat.Err_Overrun	Total number of DNP physical layer overrun errors.
GenStat.Err_Length	Total number of DNP physical layer length errors.
GenStat.Err_CRC	Total number of DNP data link layer bad CRC errors.
GenStat.Err_Overflow	Total number of DNP transport layer user data overflow errors.
GenStat.Err_Seq	Total number of DNP transport layer sequence errors.
GenStat.Err_Address	Total number of DNP transport layer address errors.
GenStat.Err_Func	Total number of bad function errors.
GenStat.Err_Obj	Total number of object unknown errors.
GenStat.Err_Range	Total number of out of range errors.
GenStat.Err_MOverflow	Total number of buffer overflow errors.
GenStat.Err_Frame	Total number of multi-frame from master errors.
GenStat.Blk_Err	Total number of block transfer errors.
GenStat.Blk_Good	Total number of successful block transfers.
GenStat.Blk_RErr	Total number of read block errors.
GenStat.Blk_WErr	Total number of write block errors.
GenStat.Blk_NErr	Total number of blocks to write with wrong block ID.
GenStat.Blk_ECntr	Total number of sequential block errors.
GenStat.Blk_EFlag	Block transfer error flag.
GenStat.Cfg_Type	Slave configuration type: 0 = Single Slave (default) 1 = Dual Slave 2 = Slave/Master
GenStat.Product	These two registers contain the product code of "DNPNET".
GenStat.Rev	These two registers contain the product version for the current running software.

GenStat.Op_Sys	These two registers contain the month and year values for the program operating system.
GenStat.Run	These two registers contain the run number value for the currently running software.
GenStat.Slave_Count	Total number of slaves configured in the Master server list.
GenStat.Cmd_Count	Total number of commands configured in the Master.
GenStat.Mem_Blk	Total number of allocated memory blocks.
GenStat.Mem_Frame	Total number of allocated frame blocks.
GenStat.Mem_DLRec	Total number of received allocated Data Link layer memory blocks.
GenStat.Mem_DLTx	Total number of transmitted allocated Data Link layer memory blocks.
GenStat.Mem_AppRec	Total number of received allocated application layer memory blocks.
GenStat.Mem_AppTx	Total memory size of Transmitted Application Layer Frame.
GenStat.Mem_DevErr	Total number of device memory allocation errors.
GenStat.Mem_PhyErr	Total number of physical layer memory allocation errors.
GenStat.Mem_DLRErr	Total number of data link layer receive memory allocation errors.
GenStat.Mem_DLTErr	Total number of data link layer transmit memory allocation errors.
GenStat.Mem_AppRErr	Total number of application layer receive memory allocation errors.
GenStat.Mem_AppTErr	Total number of application layer transmit memory allocation errors.
GenStat.Mstr_Sync	Total number of DNP synchronization errors.
GenStat.Mstr_Length	Total number of DNP length errors.
GenStat.Mstr_CRC	Total number of DNP bad CRC errors.
GenStat.P1_TX_State	Port 1 transmit state: 0 = Initialize primary DNP port transmit state to idle. 1 = Set for initial direct connect.
GenStat.Scan_Count	This value is incremented each time a complete program cycle occurs in the module.
GenStat.Mem_Free	Total amount of free memory available in the module.
GenStat.BI_Events	Total number of Binary Input events.
GenStat.BI_Buffer	Total number of Binary Input events in buffer.
GenStat.AI_Events	Total number of Analog Input events.
GenStat.A16I_Buffer	Total number of Analog 16 Input events in buffer.
GenStat.A32I_Events	Total number of Analog 32 Input events.
GenStat.A32I_Buffer	Total number of Analog 32 Input events in buffer.
GenStat.FLTI_Events	Total number of Float Input events.
GenStat.FLTI_Buffer	Total number of Float Input events in buffer.
GenStat.DBLI_Events	Total number of Double Input events.
GenStat.DBLI_Buffer	Total number of Double Floating-Point Input events in buffer.
GenStat.Events_In_Client	Total number of unread event messages.
GenStat.Events_In_Client_OF	This field contains the unread overflow for events buffer. 0 = No 1 = Yes
GenStat.Slave_IIN_Bits	This field contains the slave IIN bits.

DNPNET.STATUS. ErrorList

This array contains a list of the last 60 MVI56E-DNPNET module errors.

Name	Description
ErrorList[0] to [59]	List of MVI56E-DNPNET module errors.

DNPNET.STATUS. IINServerBits

This array contains the IIN Bits received from DNP Servers.

Name	Description
IINServerBits[0] to [39]	List of IIN Bits received from DNP Servers.

DNPNET.STATUS. ServerCommErrors

This array contains the MVI56E-DNPNET Server communication errors list.

Name	Description
ServerCommErrors[0] to [39]	List of MVI56E-DNPNET server communication errors.

DNPNET.STATUS. CommandErrors

This array contains the MVI56E-DNPNET client command errors list.

Name	Description
CommandErrors[0] to [39]	List of MVI56E-DNPNET client command errors.

DNPNET.STATUS. Network

This array contains MVI56E-DNPNET server TCP/UDP network status.

Name	Description
Network.TCP_Parsing	Indication of TCP parsing: -1 = Offline 0 = No 1 = Yes
Network.Multi_Frag	Multi-fragment message indicator: 0 = No 1 = Yes
Network.Transport_Frag	Transport fragmented message indicator.
Network.Confirm_Pend	Confirmation pending indicator: 0 = No 1 = Yes
Network.TCP_Rx_Count	TCP message received count.
Network.TCP_Tx_Count	TCP message transmitted count.
Network.Tx_State	Module transmit state.
Network.TCP_State	TCP communication: 0 = Not communicating using TCP. 1 = Communicating using TCP. 2 = Received multi-message packet.
Network.Busy_Flag	Communication busy indicator. 0 = Not busy 1 = TCP busy 2 = Received multi-message packet.
Network.App_Frame	Application frame fragmented message indicator. 0 = No 1 = Yes
Network.Tx_Frame	Transmit frame state.
Network.TCP_PacketLength	Packet length of TCP message.
Network.UDP_Parsing	Indication of UDP parsing. 0 = No 1 = Yes
Network.UDP_Rx_Count	UDP message received count.
Network.UDP_Tx_Count	UDP message transmitted count.
Network.UDP_State	UDP communication state: 0 = Not communicating using UDP. 1 = Communicating using UDP.
Network.UDP_PacketLength	Packet length of UDP message.

4.4.5 DNPNET.CONTROL Controller Tags

These values are a 'scratchpad' area of intermediate data storage variables used by the ladder logic to keep track of various logic processing functions.

Name		Data Type	Description
Events		DNPNET_Events Handler	Handling the transfer of events from the Events Message Buffer in the client to the PLC
EventMessages		DNPNET_Event Msg[15]	Events captured by the DNP client are move to this structure
CROBmsg_counter		DNPNET.CROB msg_Handler	CROB message data copied via block 9910 from DNP Server. CROBmsg_counter tags are status data used when MVI56E-DNPNET is configured as a server and passes the CROB message counter information to the sub CROBmsg_counter tags through the backplane
	CROBmsg_counter. OneShot	BOOL	Internally used in Add On Instruction (not designed to be used by the end user)
	CROBmsg_counter. CountPerBlock	INT	Number of CROB command message(s) that was(were) contained in the last 9910 Backplane Block received in the Add On Instruction.
	CROBmsg_counter. CCC	DINT	Continuous CROB message Count. Everytime CROB message is received by the server, this counter will increment by 1. This may be reset by the user. Otherwise it will be reset to 0 on the 2147483647th CROB message coming in to the server by the Add On Instruction
	CROBmsg_counter. CCBC	DINT	Continuous CROB message Block Count. Everytime 9910 block is received by the Add On Instruction, this counter will increment by 1. This may be reset by the user. Otherwise it will be reset to 0 on the 2147483647th CROB message coming in to the server by the Add On Instruction
CROBmsg_data		DNPNET_CROB_ Data[40]	CROB message data copied via block 9910 from DNP Server. CROBmsg_data tags are status data used when MVI56E-DNPNET is configured as a server and passes the CROB message counter information to the sub CROBmsg_counter tags through the backplane.
	CROBmsg_data. Point_Number	INT	This parameter was specified by the DNP3 client as the Control Relay Output Block point in the MVI56E-DNPNET (configured as a server).
	CROBmsg_data. Control_Code	SINT	This is a standard DNPNET protocol control code byte.
	CROBmsg_data. count	SINT	This parameter was specified by the DNP3 client of the number of pulses to generate for pulse output control in the CROB command.

Name		Data Type	Description
	CROBmsg_data. On_Time	DINT	This parameter was specified by the DNP3 client of the on-time interval for pulse control in the CROB command
	CROBmsg_data. Off_Time	DINT	This parameter was specified by the DNP3 client of the off-time interval for pulse control in the CROB command
Binary_Event_CalTime		DNPNET_Event_ Binary_CalTime	For storing block 9958 Binary Event Messages sent to the DNP server
Analog_Event_CalTime		DNPNET_Event_ Analog_CalTime	For storing block 9959 Analog Event Messages to be sent to DNP server
Binary_Event_CLXTime		DNPNET_Event_ Binary_CLXTime	For storing block 9968 Binary Input Events with CLX Time
Analog_Event_CLXTime		DNPNET_Event_ Analog_CLXTime	For storing Block 9969 Binary Input Events with CLX Time
Get_Module_Time		DNPNET_Module _Time_Get	Reads the module time into the PLC
Set_Module_Time		DNPNET_Module _Time_Se	Reads the PLC time to the module
CROB_Commands		DNPNET_CROB_ Commands	9901 block of CROB commands
Aux_Commands		DNPNET_Aux_ Command	9904 block of Auxiliary Client commands
CommandControl		DNPNET_ CommandControl	9902 blocks of commands to enable
Server_Comm_Errors		DNPNET_Server_ Error_Request	Count and offset of servers to report
Command_Errors		DNPNET_ Command_Error_ Request	Count and offset of commands to report
ColdBoot		BOOL	Cold Boot
WarmBoot		BOOL	Warm Boot

4.4.6 DNPNET.UTIL Controller Tags

These values contain overall module status information.

Name	Data Type	Description
LastRead	INT	Index of last read block
LastWrite	INT	Index of last write block
BlockIndex	DINT	Computed block offset for data table
ReadData	INT[50500]	Buffer file for data read from the module
WriteData	INT[30000]	Buffer file for the data written to the module
OffsetCounter	DINT	Used to calculate the buffer copy offset
CopyLength	INT	Used to calculate the buffer copy length
LastService	INT	Used to keep track of the last special outgoing block serviced
Mutex	INT	Only one special block gets serviced on every other scan
OddScan	INT	Keeps track of every other scan
FirstRun	INT	Ensures database counts are correct on PLC restart
DNP_List_Entry_Counts	DNPNET_List_Entry_Counts	Lengths of the 7 DNP3 Ethernet database lists
IED_Database	DNPNET_IED_db_Definitions	Used to configure the Client database sizes
DNP_Server_Database	DNPNET_Server_DB_Points	Used to configure the Server database sizes

4.5 Cable Connections

4.5.1 Ethernet Cable Specifications

The recommended cable is Category 5 or better. A Category 5 cable has four twisted pairs of wires, which are color-coded and cannot be swapped. The module uses only two of the four pairs.

The Ethernet port or ports on the module are Auto-Sensing. You can use either a standard Ethernet straight-through cable or a crossover cable when connecting the module to an Ethernet hub, a 10/100 Base-T Ethernet switch, or directly to a PC. The module detects the cable type and uses the appropriate pins to send and receive Ethernet signals.

Some hubs have one input that can accept either a straight-through or crossover cable, depending on a switch position. In this case, you must ensure that the switch position and cable type agree.

Refer to Ethernet Cable Configuration (page 133) for a diagram of how to configure Ethernet cable.

4.5.2 Ethernet Cable Configuration

Note: The standard connector view shown is color-coded for a straight-through cable.
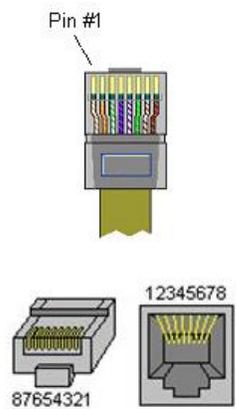
Crossover cable			Straight- through cable	
RJ-45 PIN	**RJ-45 PIN**		**RJ-45 PIN**	**RJ-45 PIN**
1 Rx+	3 Tx+		1 Rx+	1 Tx+
2 Rx-	6 Tx-		2 Rx-	2 Tx-
3 Tx+	1 Rx+		3 Tx+	3 Rx+
6 Tx-	2 Rx-		6 Tx-	6 Rx-

4.5.3 Ethernet Performance

Ethernet performance in the MVI56E-DNPNET can be affected in the following way:

- Accessing the web interface (refreshing the page, downloading files, and so on) may affect performance
- Also, high Ethernet traffic may impact performance, so consider one of these options:
  - Use managed switches to reduce traffic coming to module port
  - Use CIPconnect for these applications and disconnect the module Ethernet port from the network

DNPNET V2.00 DEVICE PROFILE DOCUMENT	
Vendor Name: ProSoft Technology, Inc.	
Device Name: MVI56E-DNPNET (VERSION 2.00)	
Highest DNPNET Level Supported: For Request: L2 For Responses: L2	Device Function: Server & Client
Notable objects, functions, and/or qualifiers supported in addition to the highest DNPNET level stated above (see attached table for complete list). Definition of selected IIN bits: Device Trouble - PLC data transfer operation is not taking place Configuration Error - User specified point or event count is too high for application (can correct only by changing configuration in PLC). 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. Floating-point variations are supported for analog input and output objects (both single and double floating-point types). Support for Obj110 (octet string) available only using read function. Events generated by IED units attached to a Client may pass their events directly to the server 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. Counter Freeze with reset will not zero values in the processor. Therefore, this function should only be used for the Client. Module will not generate events until Restart IIN bit is cleared by DNPNET Client except for events passed through module from attached IED units.	
Maximum Data Link Frame Size (octets): Transmitted: 292 Received: 292	Maximum Application Fragment Size (octets): Transmitted: 2048 Received: 2048
Maximum Data Link Re-tries: Configurable from 0 - 255	Maximum Application Layer Re-tries: None
Requires Data Link Layer Confirmation: Configurable at module start-up (never, sometimes, & always)	
Requires Application Layer Confirmation: When reporting Event Data as a server unit	
Time-outs while waiting for: Data Link Confirm : Configurable at module start-up (1 to 32767 milliseconds) Complete Application : Configurable at module start-up Fragment Application Confirm : Configurable at module start-up (1 to 32767 milliseconds) Complete Application : None Response	
Sends/Executes Control Operations: WRITE Binary : Never Outputs SELECT/OPERATE : Always DIRECT OPERATE : Always DIRECT OPERATE- : Always NO ACK Count > 1 : Always (1 to 65535) Pulse On : Always Pulse Off : Always	

DNPNET V2.00 **DEVICE PROFILE DOCUMENT**	
Latch On : Always Latch Off : Always Queue : Never Clear Queue : Never	
Reports Binary Input Change Events when no specific variation requested: 　　　　　Only time-tagged	Reports time-tagged Binary Input Change Events when no specific variation requested: 　　　　　Binary Input Change with Time
Sends Unsolicited Responses: This is configurable at module start-up. If the number of events for the Binary or Analog Input Events is greater than 0, unsolicited responses are supported. Use the Enable/Disable Unsolicited function code from the DNPNET Client for control.	Sends Static Data in Unsolicited Responses: 　　　　　Never
Default Counter Object/Variation: 　　　　Object　　　: 20 　　　　Variation　　: 5	Counters Roll Over at: 　　　　32 Bits
Sends Multi-Fragment Responses: Yes	

OBJECT			REQUEST		RESPONSE		Data Size (bits)	NOTES
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)		
1	0	Binary Input - All Variations	1	00, 01, 06, 07, 08, 17, 28			1	Server will return variation 1 data
	1	Binary Input	1	00, 01, 06, 07, 08, 17, 28	129, 130	00, 01, 06, 07, 08, 17, 28	1	Server will return this variation
	2	Binary Input with Status	1	00, 01, 06, 07, 08, 17, 28	129, 130	00, 01, 06, 07, 08, 17, 28	8	Server will return Unknown Object to this request
2	0	Binary Input Change - All Variations	1	06, 07, 08			56	Server will return variation 2 data
	1	Binary Input Change Without Time	1	06, 07, 08	129, 130	17, 28	8	Server will return this variation
	2	Binary Input Change With Time	1	06, 07, 08	129, 130	17, 28	56	Server will return this variation
	3	Binary Input Change With Relative Time	1	06, 07, 08	129, 130	17, 28	24	Server will parse this message and return no data
10	0	Binary Output - All Variations	1	06			8	Server will return variation 2 data
	1	Binary Output	1	00, 01, 06			1	Server will return Unknown Object to this request
	2	Binary Output Status	1	00, 01, 06	129, 130	00, 01	8	Server will return this variation
12	0	Control Block - All Variations					88	Server will use variation 1 control
	1	Control Relay Output Block	3, 4, 5, 6	00, 01, 17, 28	129	Echo of request	88	Server will respond correctly to this variation
	2	Pattern Control Block					88	Server will return Unknown Object to this request
	3	Pattern Mask					16	Server will return Unknown Object to this request
20	0	Binary Counter - All Variations	1, 7, 8	06			32	Server will return variation 5 data
	1	32-Bit Binary Counter	1, 7, 8	00, 01, 06, 07, 08	129, 130	00, 01	40	Server will return Unknown Object to this request
	2	16-Bit Binary Counter	1, 7, 8	00, 01, 06, 07, 08	129, 130	00, 01	24	Server will return Unknown Object to this request
	3	32-Bit Delta Counter			129, 130	00, 01	40	Server will return Unknown Object to this request
	4	16-Bit Delta Counter			129, 130	00, 01	24	Server will return Unknown Object to this request
	5	32-Bit Binary Counter Without Flag	1, 7, 8	06	129, 130	00, 01	32	Server will return this variation
	6	16-Bit Binary Counter Without Flag	1, 7, 8	06	129, 130	00, 01	16	Server will return this variation (counter upper 16-bits removed)
	7	32-Bit Delta Counter Without Flag			129, 130	00, 01	32	Server 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 Delta Counter Without Flag			129, 130	00, 01	16	Server will return Unknown Object to this request
21	0	Frozen Counter - All Variations	1				32	Server will return variation 9 data
	1	32-Bit Frozen Counter	1	00, 01, 06, 07, 08	129, 130	00, 01	40	Server will return Unknown Object to this request
	2	16-Bit Frozen Counter	1	00, 01, 06, 07, 08	129, 130	00, 01	24	Server will return Unknown Object to this request
	3	32-Bit Frozen Delta Counter					40	Server will return Unknown Object to this request
	4	16-Bit Frozen Delta Counter					24	Server will return Unknown Object to this request
	5	32-Bit Frozen Counter With Time Of Freeze					88	Server will return Unknown Object to this request
	6	16-Bit Frozen Counter With Time Of Freeze					72	Server will return Unknown Object to this request
	7	32-Bit Frozen Delta Counter With Time Of Freeze					88	Server will return Unknown Object to this request
	8	16-Bit Frozen Delta Counter With Time Of Freeze					72	Server will return Unknown Object to this request
	9	32-Bit Frozen Counter Without Flag	1	00, 01, 06, 07, 08	129, 130	00, 01	32	Server will return this variation
	10	16-Bit Frozen Counter Without Flag	1	00, 01, 06, 07, 08	129, 130	00, 01	16	Server will return this variation (counter upper 16-bits removed)
	11	32-Bit Frozen Delta Counter Without Flag					32	Server will return Unknown Object to this request
	12	16-Bit Frozen Delta Counter Without Flag					16	Server will return Unknown Object to this request
22	0	Counter Change Event - All Variations	1	06, 07, 08				Server will parse this request and return no data
	1	32-Bit Counter Change Event Without Time			129, 130	17, 28	40	Server will return Unknown Object to this request
	2	16-Bit Counter Change Event Without Time	1	06, 07, 08	129, 130	17, 28	24	Server will return Unknown Object to this request
	3	32-Bit Delta Counter Change Event Without Time					40	Server will return Unknown Object to this request
	4	16-Bit Delta Counter Change Event Without Time					24	Server will return Unknown Object to this request
	5	32-Bit Counter Change Event With Time					88	Server 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
	6	16-Bit Counter Change Event With Time					72	Server will return Unknown Object to this request
	7	32-Bit Delta Counter Change Event With Time					88	Server will return Unknown Object to this request
	8	16-Bit Delta Counter Change Event With Time					72	Server will return Unknown Object to this request
23	0	Frozen Counter Event - All Variations						Server will return Unknown Object to this request
	1	32-Bit Frozen Counter Event Without Time					40	Server will return Unknown Object to this request
	2	16-Bit Frozen Counter Event Without Time					24	Server will return Unknown Object to this request
	3	32-Bit Frozen Delta Counter Event Without Time					40	Server will return Unknown Object to this request
	4	16-Bit Frozen Delta Counter Event Without Time					24	Server will return Unknown Object to this request
	5	32-Bit Frozen Counter Event With Time					88	Server will return Unknown Object to this request
	6	16-Bit Frozen Counter Event With Time					72	Server will return Unknown Object to this request
	7	32-Bit Frozen Delta Counter Event With Time					88	Server will return Unknown Object to this request
	8	16-Bit Frozen Delta Counter Event With Time					72	Server will return Unknown Object to this request
30	0	Analog Input - All Variations	1	06			16	Server will respond with variation 4 data
	1	32-Bit Analog Input	1	00, 01, 06, 17, 28	129, 130	00, 01	40	Server will return this variation
	2	16-Bit Analog Input	1	00, 01, 06, 17, 28	129, 130	00, 01	24	Server will return this variation
	3	32-Bit Analog Input Without Flag	1	00, 01, 06, 17, 28	129, 130	00, 01	32	Server will return this variation
	4	16-Bit Analog Input Without Flag	1	00, 01, 06, 17, 28	129, 130	00, 01	16	Server will return this variation
	5	Short Floating Point Analog Input	1	00, 01, 06, 17, 28	129, 130	00, 01	40	Server will return this variation
	6	Long Floating Point Analog Input	1	00, 01, 06, 17, 28	129, 130	00, 01	72	Server will return this variation

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
31	0	Frozen Analog Input - All Variations						Server will return Unknown Object to this request
	1	32-Bit Frozen Analog Input					40	Server will return Unknown Object to this request
	2	16-Bit Frozen Analog Input					24	Server will return Unknown Object to this request
	3	32-Bit Frozen Analog Input With Time To Freeze					88	Server will return Unknown Object to this request
	4	16-Bit Frozen Analog Input With Time To Freeze					72	Server will return Unknown Object to this request
	5	32-Bit Frozen Analog Input Without Flag					32	Server will return Unknown Object to this request
	6	16-Bit Frozen Analog Input Without Flag					16	Server will return Unknown Object to this request
	7	Short Floating Point Frozen Analog Input					40	Server will return Unknown Object to this request
	8	Long Floating Point Frozen Analog Input					72	Server will return Unknown Object to this request
32	0	Analog Change Event - All Variations	1	06, 07, 08			24	Server will return variation 2 data
	1	32-Bit Analog Change Event Without Time	1	06, 07, 08	129, 130	17, 28	40	Server will return this variation
	2	16-Bit Analog Change Event Without Time	1	06, 07, 08	129, 130	17, 28	24	Server will return this variation
	3	32-Bit Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	88	Server will return this variation
	4	16-Bit Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	72	Server will return this variation
	5	Short Floating Point Analog Change Event	1	06, 07, 08	129, 130	17, 28	40	Server will return this variation
	7	Short Floating Point Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	88	Server will return this variation
	8	Long Floating Point Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	120	Server will return this variation
33	0	Frozen Analog Event - All Variations						Server will return Unknown Object to this request
	1	32-Bit Frozen Analog Event Without Time					40	Server will return Unknown Object to this request
	2	16-Bit Frozen Analog Event Without Time					24	Server will return Unknown Object to this request
	3	32-Bit Frozen Analog Event With Time					88	Server will return Unknown Object to this request
	4	16-Bit Frozen Analog Event With Time					72	Server 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
	5	Short Floating Point Frozen Analog Event					40	Server will return Unknown Object to this request
	6	Long Floating Point Frozen Analog Event					72	Server will return Unknown Object to this request
	7	Short Floating Point Frozen Analog Event With Time					88	Server will return Unknown Object to this request
	8	Long Floating Point Frozen Analog Event With Time					120	Server will return Unknown Object to this request
40	0	Analog Output Status - All Variations	1	06	129, 130	00, 01	24	Server will return variation 2 data
	1	32-Bit Analog Output Status	1	06	129, 130	00, 01	40	Server will return this variation
	2	16-Bit Analog Output Status	1	06	129, 130	00, 01	24	Server will return this variation
	3	Short Floating Point Analog Output Status	1	06	129, 130	00, 01	40	Server will return this variation
	4	Long Floating Point Analog Output Status	1	06	129, 130	00, 01	72	Server will return this variation
41	0	Analog Output Block - All Variations					24	Server 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	Server will respond to this request
	2	16-Bit Analog Output Block	3, 4, 5, 6	17, 28	129	Echo of Request	24	Server will respond to this request
	3	Short Floating Point Analog Output Block	3, 4, 5, 6	17, 28	129	Echo of Request	40	Server will respond to this request
	4	Long Floating Point Analog Output Block	3, 4, 5, 6	17, 28	129	Echo of Request	72	Server will respond to this request
50	0	Time and Date – All Variations	1, 2	07, Quant=1			48	Server will use variation 3
	1	Time and Date – Absolute Time					48	Server will respond to this variation
	2	Time and Date – Absolute Time and Interval					80	Server will return Unknown Object to this request
	3	Time and Date – Absolute Time at Last Recorded Time	2	07, Quant=1			48	Server will respond to this variation
51	0	Time and Date CTO - All Variations						Server will return Unknown Object to this request
52	1	Time Delay Coarse			129	07, Quant=1	16	Server will never return this variation
60	0	Not Defined						Not Defined in DNPNET
	1	Class 0 Data	1	06, 07, 08, Event				Server will respond to this variation with all static data
	2	Class 1 Data	1	06, 07, 08, Event				Server will respond to this variation with all class 1 data
	3	Class 2 Data	1	06, 07, 08, Event				Server will respond to this variation with all class 2 data (binary input events by default)

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	4	Class 3 Data	1	06, 07, 08, Event				Server will respond to this variation with all class 3 data (analog input events by default)
70	0	Not Defined						Not Defined in DNPNET
	1	File Identifier						Server will return Unknown Object to this request
80	0	Not Defined						Not Defined in DNPNET
	1	Internal Indications	2	00, Index=7			1	Server will respond to this variation
81	0	Not Defined						Not Defined in DNPNET
	1	Storage Object						Server will return Unknown Object to this request
82	0	Not Defined						Not Defined in DNPNET
	1	Device Profile						Server will return Unknown Object to this request
83	0	Not Defined						Not Defined in DNPNET
	1	Private Registration Object						Server will return Unknown Object to this request
	2	Private Registration Objection Descriptor						Server will return Unknown Object to this request
90	0	Not Defined						Not Defined in DNPNET
	1	Application Identifier						Server will return Unknown Object to this request
100	0	Not Defined						Not Defined in DNPNET
	1	Short Floating Point					48	Server will return Unknown Object to this request
	2	Long Floating Point					80	Server will return Unknown Object to this request
	3	Extended Floating Point					88	Server will return Unknown Object to this request
101	0	Not Defined						Not Defined in DNPNET
	1	Small Packed Binary-Coded Decimal					16	Server will return Unknown Object to this request
	2	Medium Packed Binary-Coded Decimal					32	Server will return Unknown Object to this request
	3	Large Packed Binary-Coded Decimal					64	Server will return Unknown Object to this request
110	0	Not Defined						Not Defined as the variation determines the string length
	1 to 100	Octet String	1	00, 01, 06, 07, 08, 17, 28	129, 130	00, 01, 07, 08, 17, 28	8 * Var #	The module will return this variation for the points defined in the module. The variation determines the returned string length.
No Object			13					Server supports the Cold Restart Function and will return Obj 52, Var 2, Qual 7, Cnt 1

OBJECT			REQUEST		RESPONSE		Data Size (bits)	NOTES
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)		
			14					Server supports the Warm Restart Function and will return Obj 52, Var 2, Qual 7, Cnt 1
			20					Server supports the Enable Unsolicited Function
			21					Server supports the Disable Unsolicited Function
			23					Server does not support the Delay Measurement & Time Synchronization Function

OBJECT			REQUEST		RESPONSE		Data Size (bits)	NOTES
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)		
1	0	Binary Input - All Variations	1	00, 01, 06			1	Client will generate this variation
	1	Binary Input	1	00, 01, 06	129, 130	00, 01	1	Client will generate and process this variation
	2	Binary Input with Status	1	00, 01, 06	129, 130	00, 01	8	Client will generate and process this variation
2	0	Binary Input Change - All Variations	1	06, 07, 08			56	Client will generate this variation
	1	Binary Input Change Without Time	1	06, 07, 08	129, 130	17, 28	8	Client will generate and process this variation
	2	Binary Input Change With Time	1	06, 07, 08	129, 130	17, 28	56	Client will generate and process this variation
	3	Binary Input Change With Relative Time	1	06, 07, 08	129, 130	17, 28	24	Client will generate and process this variation
10	0	Binary Output - All Variations	1	00, 01, 06			8	
	1	Binary Output	1	00, 01, 06			1	Client will generate and process these variations
	2	Binary Output Status	1	00, 001, 06	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	Client 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	Client will generate this variation
	1	32-Bit Binary Counter			129, 130	00, 01	40	Client will process this variation
	2	16-Bit Binary Counter			129, 130	00, 01	24	Client will process this variation
	3	32-Bit Delta Counter			129, 130	00, 01	40	Client 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	Client will process this variation
	5	32-Bit Binary Counter Without Flag	1, 7, 8, 9, 10	00, 01, 06	129, 130	00, 01	32	Client will generate and process this variation
	6	16-Bit Binary Counter Without Flag	1, 7, 8, 9, 10	00, 01, 06	129, 130	00, 01	16	Client will generate and process this variation
	7	32-Bit Delta Counter Without Flag			129, 130	00, 01	32	Client will process this variation
	8	16-Bit Delta Counter Without Flag			129, 130	00, 01	16	Client will process this variation
21	0	Frozen Counter - All Variations	1	06			32	Client will generate this variation
	1	32-Bit Frozen Counter			129, 130	00, 01	40	Client will process this variation
	2	16-Bit Frozen Counter			129, 130	00, 01	24	Client 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	00, 01, 06	129, 130	00, 01	32	Client will generate and process this variation
	10	16-Bit Frozen Counter Without Flag	1	00, 01, 06	129, 130	00, 01	16	Client 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				Client will not generate a request for this variation
	1	32-Bit Counter Change Event Without Time			129, 130	17, 28	40	Client will process this variation
	2	16-Bit Counter Change Event Without Time			129, 130	17, 28	24	Client 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	

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	7	32-Bit Delta Counter Change Event With Time					88	
	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	00, 01, 06			16	Client will generate this variation
	1	32-Bit Analog Input	1	00, 01, 06	129, 130	00, 01	40	Client will generate and process this variation
	2	16-Bit Analog Input	1	00, 01, 06	129, 130	00, 01	24	Client will generate and process this variation
	3	32-Bit Analog Input Without Flag	1	00, 01, 06	129, 130	00, 01	32	Client will generate and process this variation
	4	16-Bit Analog Input Without Flag	1	00, 01, 06	129, 130	00, 01	16	Client will generate and process this variation
	5	Short Floating Point Analog Input	1	00, 01, 06	129, 130	00, 01	40	Client will generate and process this variation
	6	Long Floating Point Analog Input						
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	

OBJECT			REQUEST		RESPONSE		Data Size (bits)	NOTES
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)		
	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	Client will generate this variation
	1	32-Bit Analog Change Event Without Time	1	06, 07, 08	129, 130	17, 28	40	Client will generate and process this variation.
	2	16-Bit Analog Change Event Without Time	1	06, 07, 08	129, 130	17, 28	24	Client will generate and process this variation
	3	32-Bit Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	88	Client will generate and process this variation.
	4	16-Bit Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	72	Client will generate and process this variation
	5	Short Floating Point Analog Change Event	1	06, 07, 08	129, 130	17, 28	40	Client will generate and process this variation
	7	Short Floating Point Frozen Analog Input	1	06, 07, 08	129, 130	17, 28	88	
		Long Floating Point Frozen Analog Input	1	06, 07, 08	129, 139	17, 28	120	
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	
	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	00, 01, 06	129, 130	00, 01	24	Client will generate these variations and parse the responses.
	1	32-Bit Analog Output Status	1	00, 01, 06	129, 130	00,01	40	
	2	16-Bit Analog Output Status	1	00, 01, 06	129, 130	00, 01	24	
	3	Short Floating Point Analog Output Status	1	00, 01, 06	129, 130	00, 01	40	
	4	Long Floating Point Analog Output Status	1	00, 01, 06	129, 130	00, 01	72	
41	0	Analog Output Block - All Variations	3, 4, 5, 6	17.28	12		24	
	1	32-Bit Analog Output Block	3, 4, 5, 6	17.28			40	
	2	16-Bit Analog Output Block	3, 4, 5, 6	17, 28	129	Echo of Request	24	Maximum number of points supported: 20. Client will generate this variation and parse the response.
	3	Single-Precision Float Analog Output Block	3, 4, 5, 6	17, 28	129	Echo of Request	24	Client will generate this variation and parse the response.
50	0	Time and Date – All Variations					48	

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	1	Time and Date – Absolute Time					48	
	2	Time and Date – Absolute Time and Interval					80	
	3	Time and Date – Absolute Time at Last Recorded Time	2	07, Quant=1			48	Client will generate this variation
51	0	Time and Date CTO - All Variations						
52	1	Time Delay Coarse			129, 130	07, With Quant=1	16	Client will not process this variation
60	0	Not Defined						Not Defined in DNPNET
	1	Class 0 Data	1	06	129	07, Quantity = 1	16	Client will generate this variation
	2	Class 1 Data	1	06	129	07, Quant= 1	16	Client will generate this variation
	3	Class 2 Data	1	06				Client will generate this variation
	4	Class 3 Data	1	06				Client will generate this variation
70	0	Not Defined						Not Defined in DNPNET
	1	File Identifier						
80	0	Not Defined						Not Defined in DNPNET
	1	Internal Indications	2	00, Index=7			24	The Client will generate this variation
81	0	Not Defined						Not Defined in DNPNET
	1	Storage Object						
82	0	Not Defined						Not Defined in DNPNET
	1	Device Profile						
83	0	Not Defined						Not Defined in DNPNET
	1	Private Registration Object						
	2	Private Registration Objection Descriptor						
90	0	Not Defined						Not Defined in DNPNET
	1	Application Identifier						
100	0	Not Defined						Not Defined in DNPNET
	1	Short Floating Point					48	
	2	Long Floating Point					80	
	3	Extended Floating Point					88	
101	0	Not Defined						Not Defined in DNPNET
	1	Small Packed Binary-Coded Decimal					16	
	2	Medium Packed Binary-Coded Decimal					32	
	3	Large Packed Binary-Coded Decimal					64	
No Object			13					Client supports the Cold Restart Function

OBJECT			REQUEST		RESPONSE			NOTES
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	
			14					Client supports the Warm Restart Function
			20					Client supports the Enable Unsolicited Function
			21					Client supports the Disable Unsolicited Function

5 Support, Service & Warranty

5.1 Contacting Technical Support

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

1 Product Version Number
2 System architecture
3 Network details

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

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

Note: For technical support calls within the United States, ProSoft Technology's 24/7 after-hours phone support is available for urgent plant-down issues.

North America (Corporate Location)	Europe / Middle East / Africa Regional Office
Phone: +1.661.716.5100 info@prosoft-technology.com Languages spoken: English, Spanish REGIONAL TECH SUPPORT support@prosoft-technology.com	Phone: +33.(0)5.34.36.87.20 france@prosoft-technology.com Languages spoken: French, English REGIONAL TECH SUPPORT support.emea@prosoft-technology.com
Latin America Regional Office	Asia Pacific Regional Office
Phone: +52.222.264.1814 latinam@prosoft-technology.com Languages spoken: Spanish, English REGIONAL TECH SUPPORT support.la@prosoft-technology.com	Phone: +60.3.2247.1898 asiapc@prosoft-technology.com Languages spoken: Bahasa, Chinese, English, Japanese, Korean REGIONAL TECH SUPPORT support.ap@prosoft-technology.com

For additional ProSoft Technology contacts in your area, please visit:
www.prosoft-technology.com/About-Us/Contact-Us.

5.2 Warranty Information

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