ProSoft® TECHNOLOGY

Where Automation Connects.

inRAx®

MVI69-N2

CompactLogix or MicroLogix Platform

Johnson Controls N2 Slave Interface Module

10/31/2008

USER MANUAL

Please Read This Notice

Successful application of this module requires a reasonable working knowledge of the Rockwell Automation CompactLogix or MicroLogix hardware, the MVI69-N2 Module and the application in which the combination is to be used. For this reason, it is important that those responsible for implementation satisfy themselves that the combination will meet the needs of the application without exposing personnel or equipment to unsafe or inappropriate working conditions.

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

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

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

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

Battery Life Advisory

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

Note: The battery is not user replaceable.

Your Feedback Please

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

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

MVI69-N2 User Manual
10/31/2008

ProSoft® Product Documentation

In an effort to conserve paper, ProSoft Technology no longer includes printed manuals with our product shipments. User Manuals, Datasheets, Sample Ladder Files, and Configuration Files are provided on the enclosed CD and are available at no charge from our web site: http://www.prosoft-technology.com
Printed documentation is available for purchase. Contact ProSoft Technology for pricing and availability.
Asia Pacific: +603.7724.2080
Europe, Middle East, Africa: +33.5.34.36.87.20
Latin America: +1.281.298.9109
North America: +1.661.716.5100

Contents

Guide to the MVI69-N2 User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 9)	This Section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Verify Communication, Diagnostic and Troubleshooting	→	Verifying Communication (page 51) Diagnostics and Troubleshooting (page 43)	This section describes how to verify communications with the network. Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview Glossary	→	Reference (page 53) Functional Overview (page 55) Product Specifications (page 53)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 73)	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

Installing the MVI69-N2 module requires a reasonable working knowledge of the Rockwell Automation hardware, the MVI69-N2 Module and the application in which they will be used.

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

1.1 System Requirements

The MVI69-N2 module requires the following minimum hardware and software components:

- Rockwell Automation CompactLogix or MicroLogix processor, with compatible power supply and one free slot in the rack, for the MVI69-N2 module. The module requires 800mA of available power.

Important: The MVI69-N2 module has a power supply distance rating of 2 (L43 and L45 installations on first 2 slots of 1769 bus).

Important: For 1769-L23x processors, please make note of the following limitations.

- 1769-L23-QBFC1B = 800mA at 5Vdc (1 MVI69-N2 will use all 800mA of available power. No other modules can be used with an MVI69 module connected to this processor).
- 1769-L23E-QB1B = 1000mA at 5Vdc (1 MVI69-N2 will use 800mA of available power. One other module can be used on this rack provided it consumes less than 200mA at 5Vdc.
- 1769-L23E-QBFC1B = 450mA at 5Vdc (no MVI69 module can be used with this processor)

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

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

1.2 Package Contents

The following components are included with your MVI69-N2 module, and are all required for installation and configuration.

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

Qty.	Part Name	Part Number	Part Description
1	MVI69-N2 Module	MVI69-N2	Johnson Controls N2 Slave Interface Module
1	Cable	Cable #15, RS232 Null Modem	For RS232 Connection to the CFG Port
3	Cable	Cable #14, RJ45 to DB9 Male Adapter cable	For DB9 Connection to Module's Port
2	Adapter	1454-9F	Two Adapters, DB9 Female to Screw Terminal. For RS422 or RS485 Connections to Port 1 and 2 of the Module
1	ProSoft Solutions CD		Contains sample programs, utilities and documentation for the MVI69-N2 module.

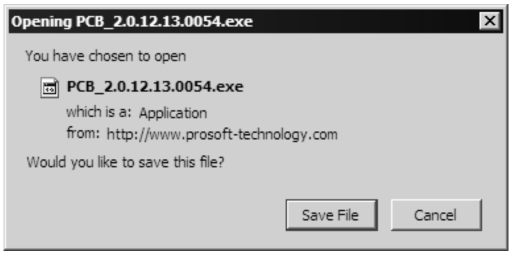
If any of these components are missing, please contact ProSoft Technology Support for replacement parts.

1.3 Install ProSoft Configuration Builder Software

You must install the ProSoft Configuration Builder (PCB) software in order to configure the MVI69-N2 module. You can always get the newest version of ProSoft Configuration Builder from the ProSoft Technology web site.

To install ProSoft Configuration Builder from the ProSoft Web Site

1 Open your web browser and navigate to http://www.prosoft-technology.com/pcb
2 Click the **Download Here** link to download the latest version of ProSoft Configuration Builder.
3 Choose "Save" or "Save File" when prompted. The following illustrations show the file download prompt for two of the most common web browsers.



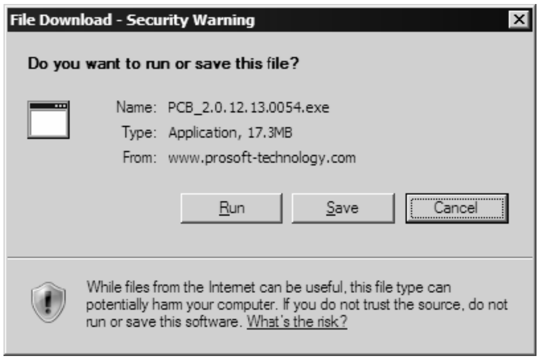
Opening PCB_2.0.12.13.0054.exe

You have chosen to open

PCB_2.0.12.13.0054.exe

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

Would you like to save this file?



File Download - Security Warning

Do you want to run or save this file?

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

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

4 Make a note of the location where you saved the file, for example "Desktop", or "My Documents", so you can start the installation program.
5 When the download is complete, locate and open the file, and then follow the instructions on your screen to install the program.

If you do not have access to the Internet, you can install ProSoft Configuration Builder from the ProSoft Solutions CD-ROM, included in the package with your MVI69-N2 module.

To install ProSoft Configuration Builder from the CD-ROM

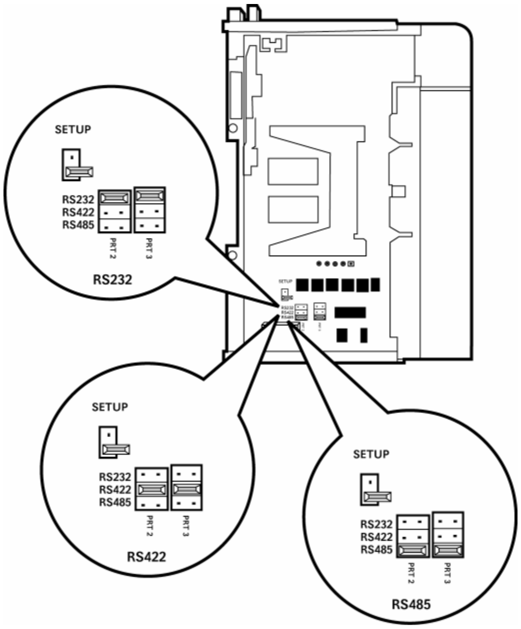
1 Insert the ProSoft Solutions CD-ROM into the CD drive of your PC. Wait for the startup screen to appear.
2 On the startup screen, click *Product Documentation*. This action opens an explorer window.

3 Click to open the *Utilities* folder. This folder contains all of the applications and files you will need to set up and configure your module.
4 Double-click the *ProSoft Configuration Builder Setup* program and follow the instructions on your screen to install the software on your PC.

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

1.4 Setting Jumpers

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



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

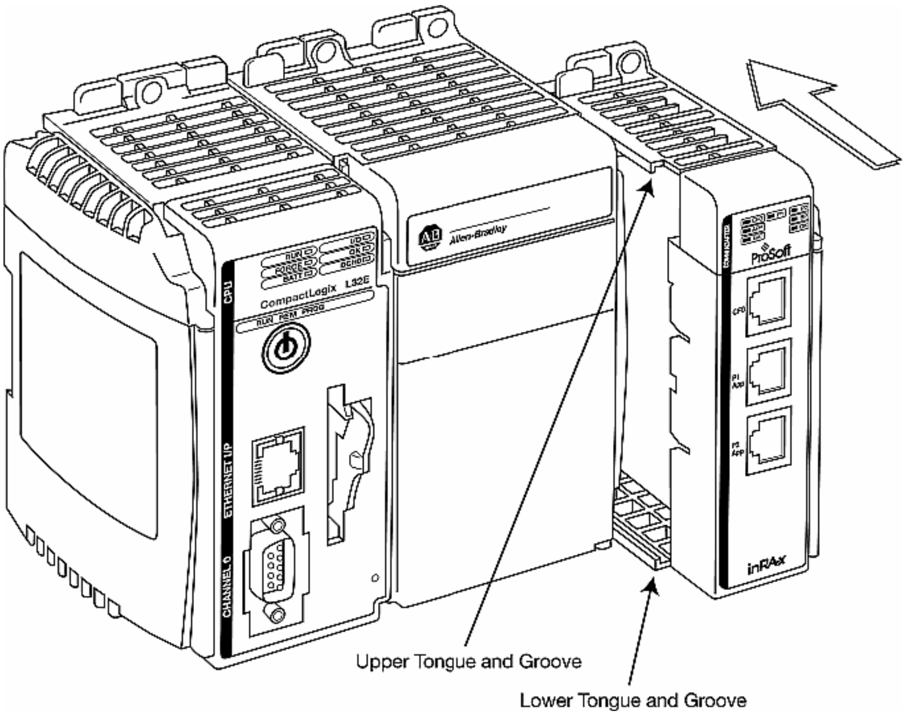
1.5 Install the Module in the Rack

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

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

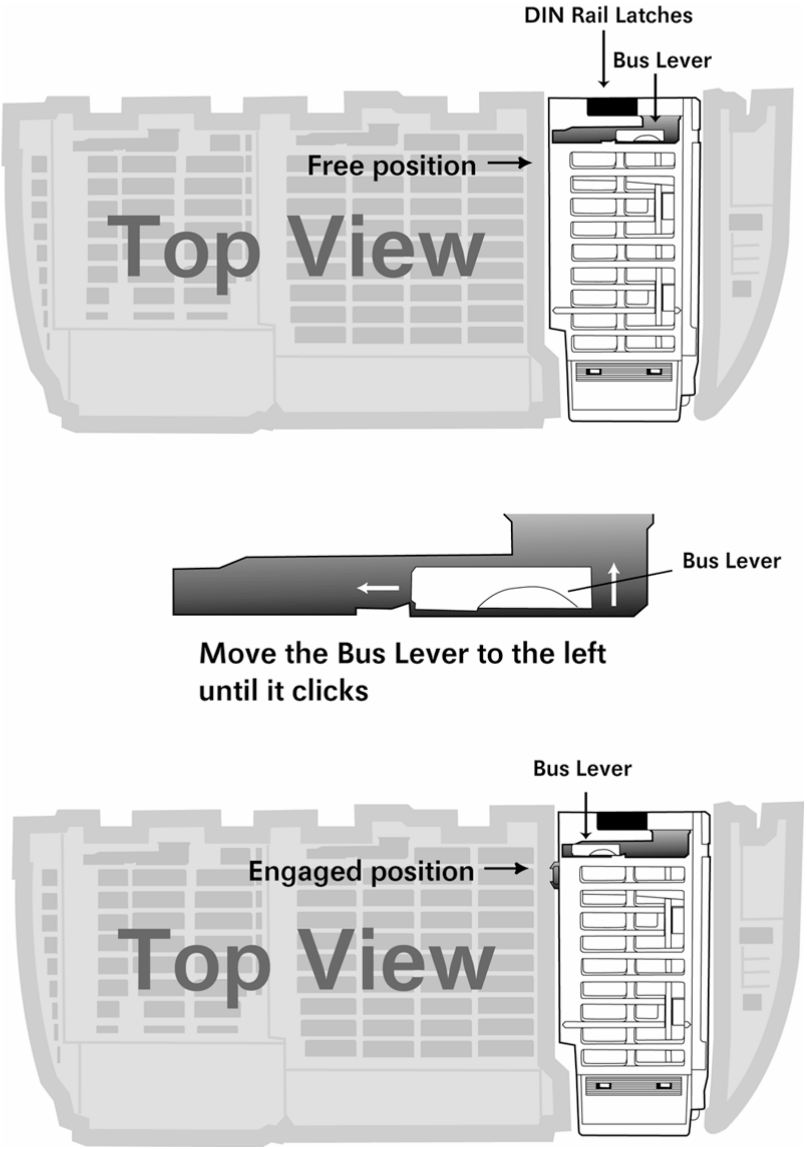
> Warning: This module is not hot-swappable! Always remove power from the rack before inserting or removing this module, or damage may result to the module, the processor, or other connected devices.

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



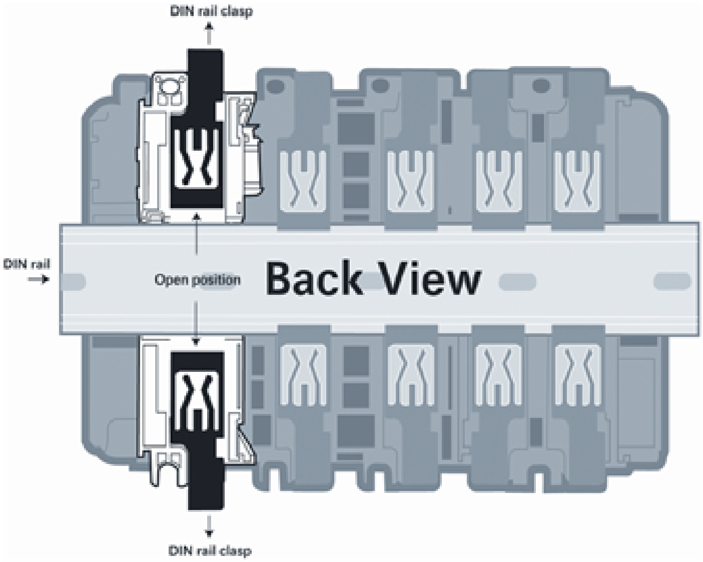
2 Move the module back along the tongue-and-groove slots until the bus connectors on the MVI69 module and the adjacent module line up with each other.

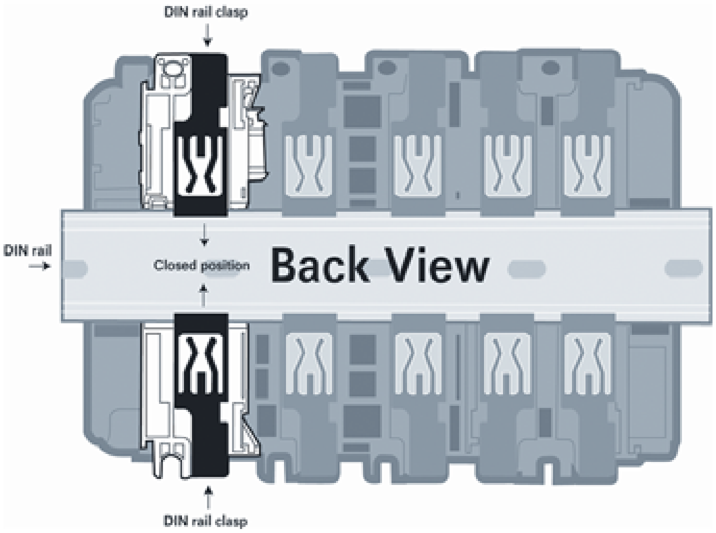
3 Push the module's bus lever back slightly to clear the positioning tab and move it firmly to the left until it clicks. Ensure that it is locked firmly in place.



4 Close all DIN rail latches.

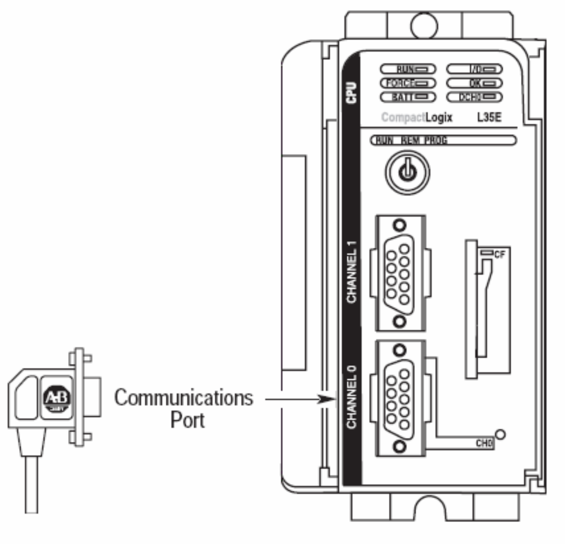
5 Press the DIN rail mounting area of the controller against the DIN rail. The latches will momentarily open and lock into place.



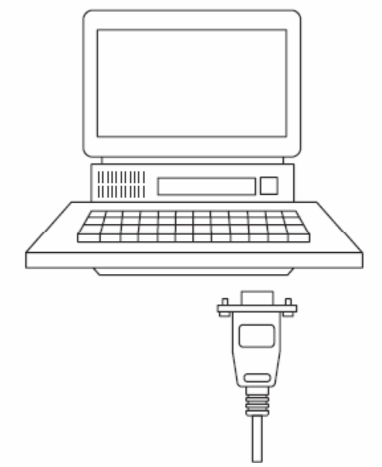


1.6 Connect your PC to the Processor

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



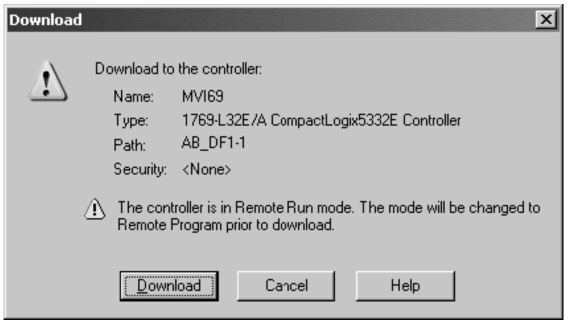
2 Connect the straight connector end of the cable to the serial port on your computer.



1.7 Download the Sample Program to the Processor

Important: For most applications, the sample program will work without modification.
Note: The key switch on the front of the CompactLogix processor must be in the REM position.

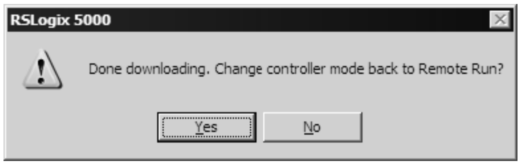
1. If you are not already online to the processor, open the Communications menu, and then choose Download. RSLogix will establish communication with the processor.
2. When communication is established, RSLogix will open a confirmation dialog box. Click the Download button to transfer the sample program to the processor.


Download

Download to the controller:
Name: MVI69
Type: 1769-L32E/A CompactLogix5332E Controller
Path: AB_DF1-1
Security: <None>

The controller is in Remote Run mode. The mode will be changed to Remote Program prior to download.

3. RSLogix will compile the program and transfer it to the processor. This process may take a few minutes.
4. When the download is complete, RSLogix will open another confirmation dialog box. Click OK to switch the processor from Program mode to Run mode.


RSLogix 5000

Done downloading. Change controller mode back to Remote Run?

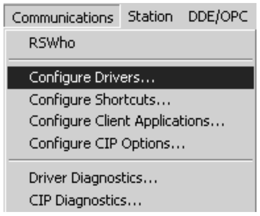
Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

1.7.1 Configuring RSLinx

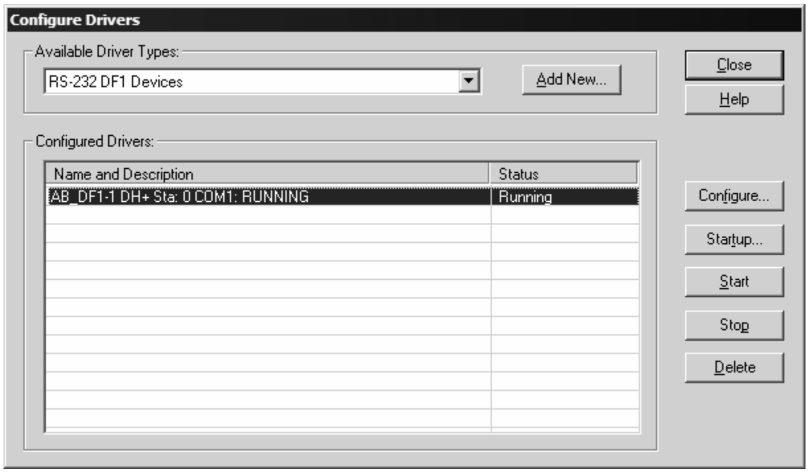
If RSLogix is unable to establish communication with the processor, follow these steps:

1 Open RSLinx.

2 Open the Communications menu, and choose Configure Drivers.



This action opens the Configure Drivers dialog box.



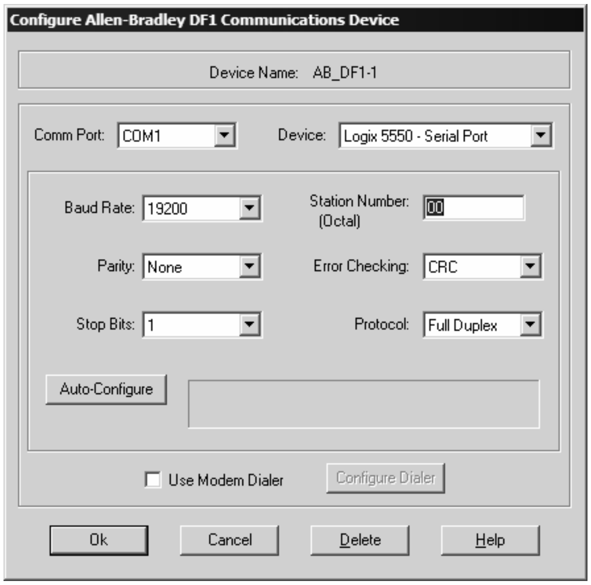
Available Driver Types: RS-232 DF1 Devices

Configured Drivers:

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

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

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



Configure Allen-Bradley DF1 Communications Device

Device Name: AB_DF1-1

Comm Port: COM1

Device: Logix 5550 - Serial Port

Baud Rate: 19200

Station Number: (Octal) 00

Parity: None

Error Checking: CRC

Stop Bits: 1

Protocol: Full Duplex

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

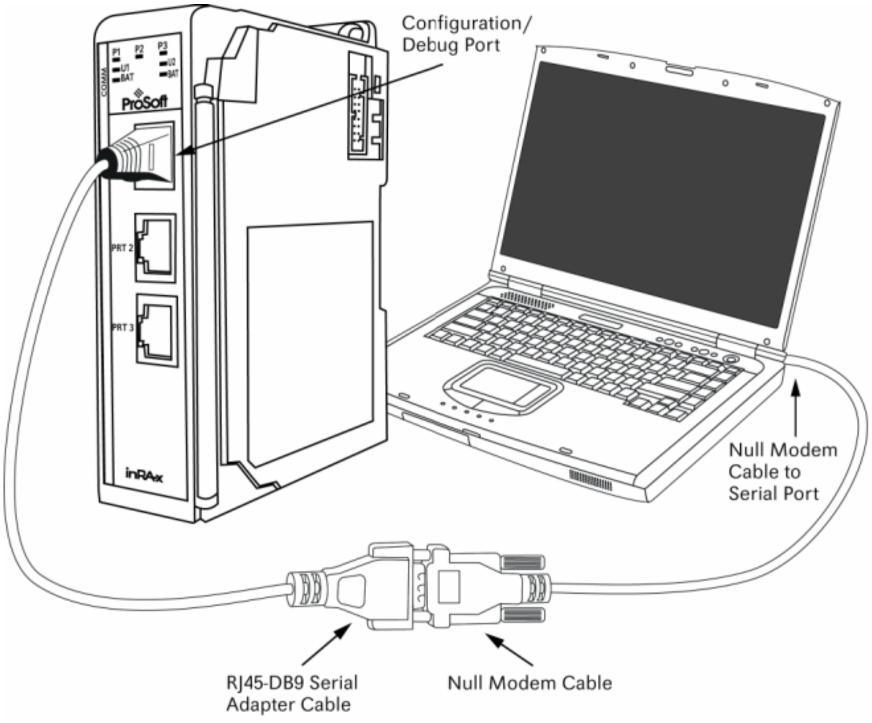
5 When you see the message "Auto Configuration Successful", click the OK button to dismiss the dialog box.

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

1.8 Connect your PC to the Module

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

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



2 Configuring the MVI69-N2 Module

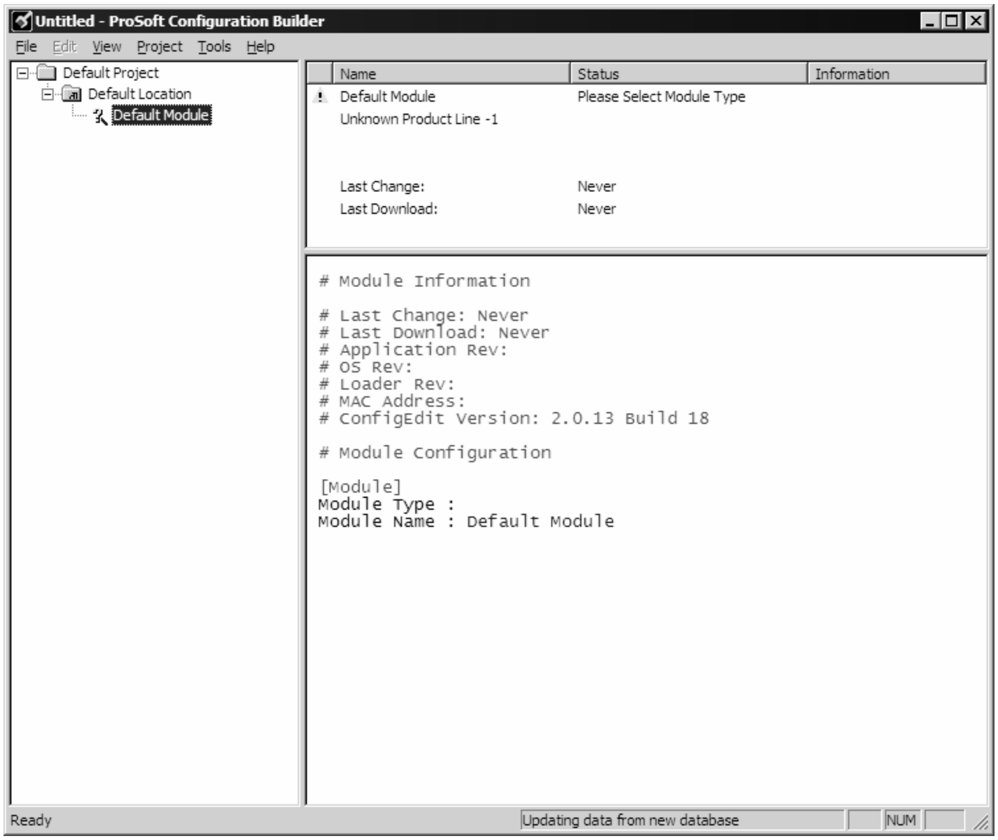
In This Chapter

2.1 ProSoft Configuration Builder

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

2.1.1 Set Up the Project

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



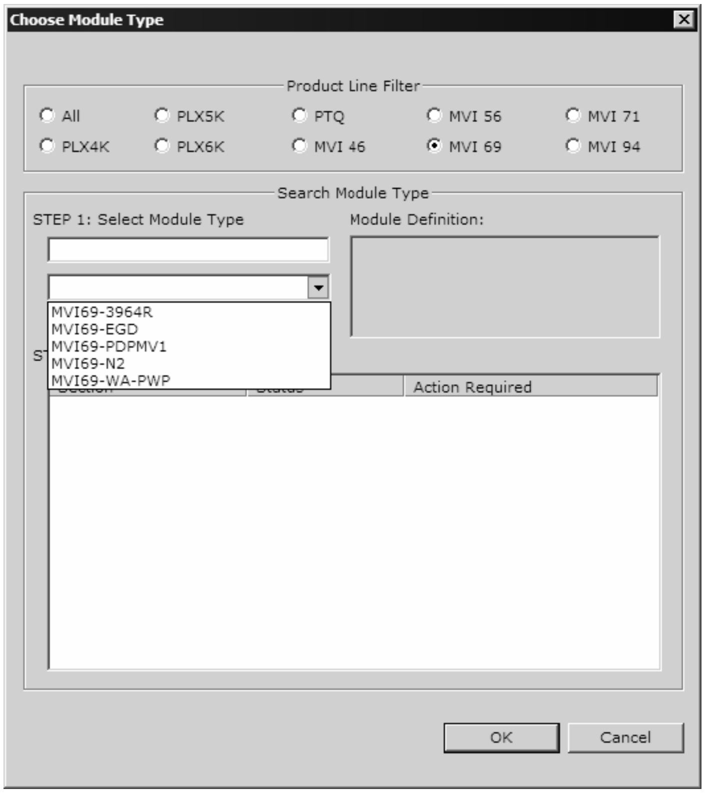
Name	Status	Information
Default Module	Please Select Module Type	
Unknown Product Line -1		
Last Change:	Never	
Last Download:	Never	

```
# Module Information

# Last Change: Never
# Last Download: Never
# Application Rev:
# OS Rev:
# Loader Rev:
# MAC Address:
# ConfigEdit Version: 2.0.13 Build 18

# Module Configuration

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

Your first task is to add the MVI69-N2 module to the project.

1 Use the mouse to select "Default Module" in the tree view, and then click the right mouse button to open a shortcut menu.

2 On the shortcut menu, choose "Choose Module Type". This action opens the Choose Module Type dialog box.



Choose Module Type

Product Line Filter

○ All ○ PLX5K ○ PTQ ○ MVI 56 ○ MVI 71
○ PLX4K ○ PLX6K ○ MVI 46 ● MVI 69 ○ MVI 94

3 In the Product Line Filter area of the dialog box, select MVI69. In the Select Module Type dropdown list, select MVI69-N2, and then click OK to save your settings and return to the ProSoft Configuration Builder window.

Adding a Project

To add a project to an existing project file:

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

Adding a Module

To add a module to your project:

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

Or

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

To add a module to a different location:

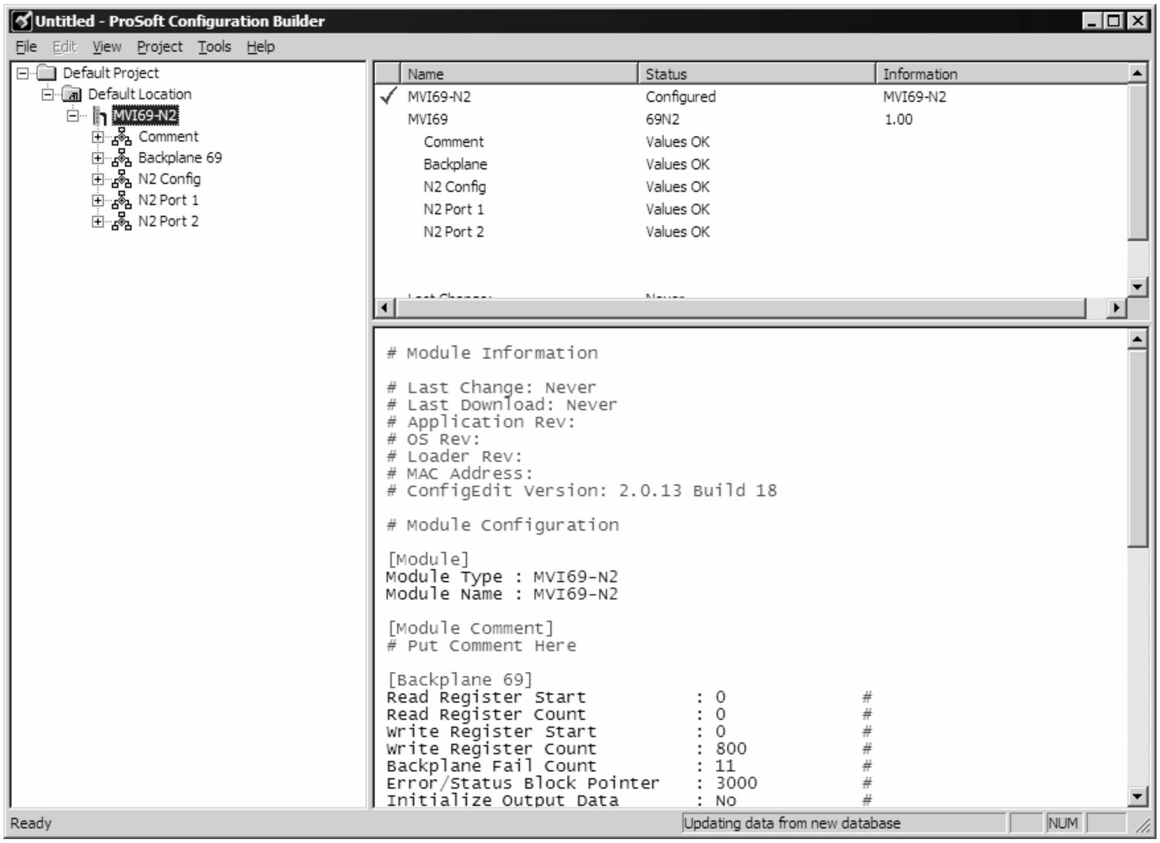
1 Right-click the Location folder and choose Add Module. A new module icon appears.

Or

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

2.1.2 Set Module Parameters

Notice that the contents of the information pane and the configuration pane changed when you added the MVI69-N2 module to the project.



Name	Status	Information
✓ MVI69-N2	Configured	MVI69-N2
MVI69	69N2	1.00
Comment	Values OK	
Backplane	Values OK	
N2 Config	Values OK	
N2 Port 1	Values OK	
N2 Port 2	Values OK	

```
# Module Information

# Last Change: Never
# Last Download: Never
# Application Rev:
# OS Rev:
# Loader Rev:
# MAC Address:
# ConfigEdit Version: 2.0.13 Build 18

# Module Configuration

[Module]
Module Type : MVI69-N2
Module Name : MVI69-N2

[Module Comment]
# Put Comment Here

[Backplane 69]
Read Register Start          : 0            #
Read Register Count          : 0            #
Write Register Start         : 0            #
Write Register Count         : 800          #
Backplane Fail Count         : 11           #
Error/Status Block Pointer   : 3000         #
Initialize Output Data       : No           #
```

Updating data from new database

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

To rename an object:

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

Module Entries

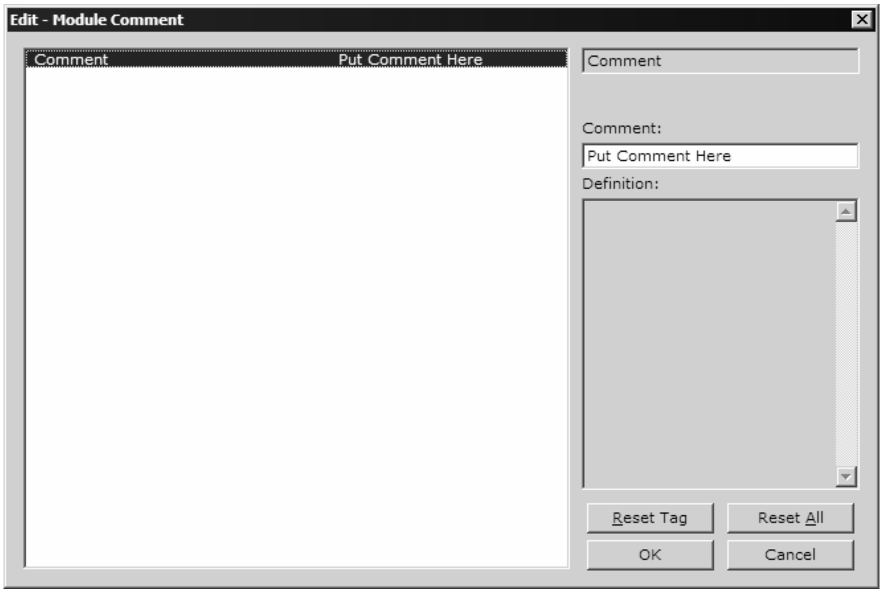
To configure module parameters

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

Comment Entries

To add comments to your configuration file:

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



Edit - Module Comment

Comment	Put Comment Here

Comment

Comment: Put Comment Here

Definition:

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

Printing a Configuration File

To print a configuration file:

1. Select the Module icon, and then click the right mouse button to open a shortcut menu.
2. On the shortcut menu, choose View Configuration. This action opens the View Configuration window.
3. On the View Configuration window, open the File menu, and choose Print. This action opens the Print dialog box.
4. On the Print dialog box, choose the printer to use from the dropdown list, select printing options, and then click OK.

2.2 [Backplane 69]

2.2.1 Read Register Start

Range 0 to 3999

This parameter specifies the starting register in the module where data will be transferred from the module to the processor. Valid range for this parameter is 0 to 3999.

2.2.2 Read Register Count

0 to 4000

The Read Register Count parameter defines the size of the module's input database, up to a maximum value of 4000 words.

2.2.3 Write Register Start

0 to 3999

This parameter specifies the starting register in the module where the data will be transferred from the processor to the module.

2.2.4 Write Register Count

Range 0 to 4000

This parameter specifies the number of registers to be transferred from the module to the processor. Valid entry for this parameter is 0 to 4000.

2.2.5 Backplane Fail Count

0 to 65535

This parameter specifies the number of consecutive backplane transfer failures that can occur before communications should be halted.

2.2.6 Error/Status Block Pointer

1 to 7000

The Error/Status Block Pointer parameter is used to specify the range of database registers to use for error and status data. The value range should be no lower than 700, to avoid overwriting data, and no higher than 6970, to allow sufficient space for the error/status block.

2.2.7 Initialize Output Data

Yes or No

This parameter determines if the output data for the module should be initialized with values from the processor. If the value is set to N, the output data will be initialized to 0. If the value is set to Y, the data will be initialized with data from the processor.

2.2.8 Block Transfer Size

60, 120 or 240

This read-only parameter specifies the number of words in each block transferred between the module and processor. Valid values for this parameter are 60, 120 and 240.

2.3 [N2 Config]

This section is used to define the N2 configuration data.

2.3.1 Binary Input Object Count

0 to 960

Determines the number of Binary Input Objects available

2.3.2 Analog Input Object Count

0 to 256

Determines the number of Analog Input Objects available.

2.3.3 Binary Output Object Count

0 to 960

Determines the number of Binary Output Objects available.

2.3.4 Analog Output Object Count

0 to 256

Determines the number of Analog Output Objects available.

2.4 [N2 Port x]

This section is used to define the port configuration for the N2 device

2.4.1 Enable

Yes or No

This parameter specifies whether to enable or disable the port. No = Port Disabled, Yes = Port Enabled.

2.4.2 Slave Address

1 to 255

This parameter defines the Slave Node Address for the internal database. All requests received by the port with this address are processed by the module. Verify that each device has a unique address on a network. Valid range for this parameter is 1 to 255.

2.4.3 Baud Rate

300 to 115200

This parameter specifies the baud rate to be used on the port. Valid values are 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200.

2.4.4 Parity

None, Odd, Even

Parity is a simple error checking algorithm used in serial communication. This parameter specifies the type of parity checking to use.

All devices communicating through this port must use the same parity setting.

2.4.5 Data Bits

7 or 8

This parameter sets the number of data bits for each word used by the protocol. All devices communicating through this port must use the same number of data bits.

2.4.6 Stop Bits

1 or 2

Stop bits signal the end of a character in the data stream. For most applications, use one stop bit. For slower devices that require more time to resynchronize, use two stop bits.

All devices communicating through this port must use the same number of stop bits.

2.4.7 RTS On

0 to 65535 milliseconds

This parameter sets the number of milliseconds to delay after RTS is asserted before the data will be transmitted.

2.4.8 RTS Off

0 to 65535 milliseconds

This parameter sets the number of milliseconds to delay after the last byte of data is sent before the RTS modem signal will be set low.

2.4.9 Minimum Response Delay

0 to 65535

This parameter sets the number of milliseconds to wait to respond to a request on the port. This is required for slow reacting devices.

2.4.10 Use CTS Line

Yes or No

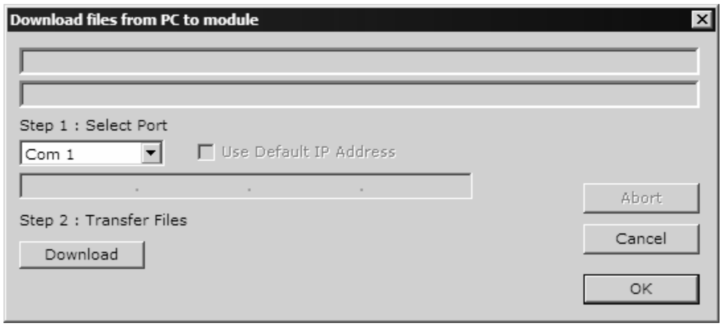
This parameter specifies if the CTS modem control line is to be used. If the parameter is set to No, the CTS line will not be monitored. If the parameter is set to Yes, the CTS line will be monitored and must be high before the module will send data. Normally, this parameter is required when half-duplex modems are used for communication (2-wire).

2.5 Download the Project to the Module

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

To Download the Project File

1. In the tree view in ProSoft Configuration Builder, click once to select the MVI69-N2 module.
2. Open the **Project menu,** and then choose **Module / Download**. The program will scan your PC for a valid com port (this may take a few seconds). When PCB has found a valid com port, the following dialog box will open.



Download files from PC to module

Step 1 : Select Port

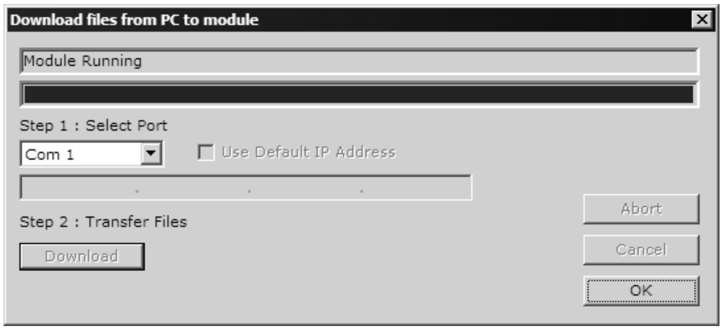
Com 1

☐ Use Default IP Address

Step 2 : Transfer Files

3. Choose the com port to use from the dropdown list, and then click the Download button.

   The module will perform a platform check to read and load its new settings. When the platform check is complete, the status bar in ProSoft Configuration Builder will be updated with the message *"Module Running*".



Download files from PC to module

Module Running

Step 1 : Select Port

Com 1

☐ Use Default IP Address

Step 2 : Transfer Files

3 Ladder Logic

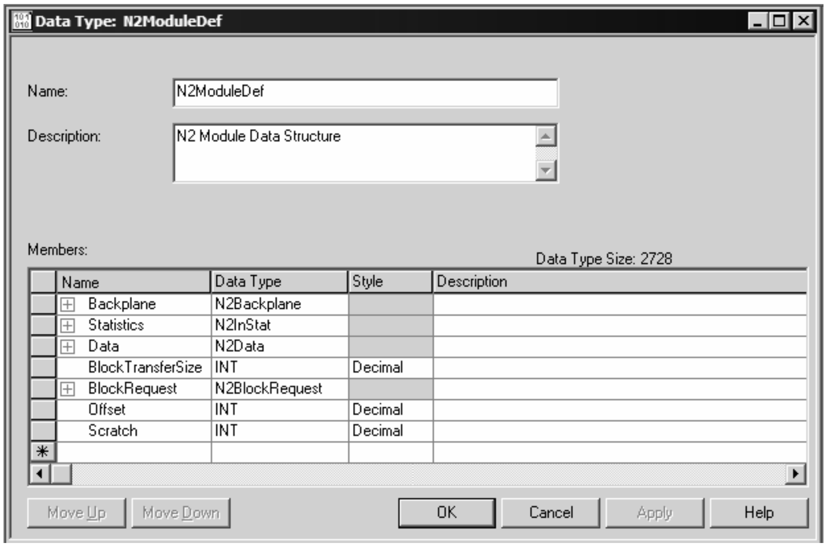
In This Chapter

Ladder logic is required for application of the MVI69-N2 module. Tasks that must be handled by the ladder logic are module data transfer, special block handling and status data receipt. Additionally, a power-up handler may be needed to handle the initialization of the module's data and to clear any processor fault conditions.

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

3.1 Module Data Object (N2)

All data related to the MVI69-N2 is stored in a user defined data type. An instance of the data type is required before the module can be used. This is done by declaring a variable of the data type in the Controller Tags Edit Tags dialog box. The structure of the object is displayed in the following illustration:



Name: N2ModuleDef

Description: N2 Module Data Structure

Members:

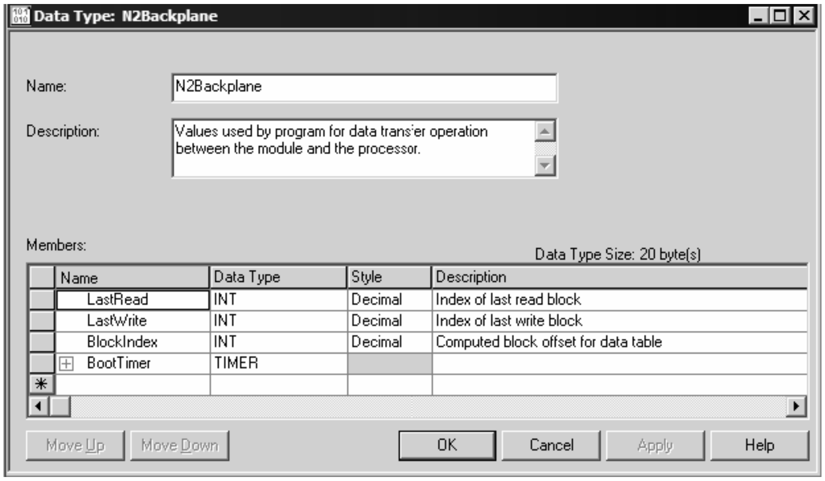
Data Type Size: 2728

Name	Data Type	Style	Description
⊞ Backplane	N2Backplane		
⊞ Statistics	N2InStat		
⊞ Data	N2Data		
BlockTransferSize	INT	Decimal	
⊞ BlockRequest	N2BlockRequest		
Offset	INT	Decimal	
Scratch	INT	Decimal	

This object contains objects that define the configuration, user data, status, and command control data related to the module. Each of these object types is discussed in the following topics of the document.

3.1.1 Backplane Control Object

This data object stores the variables required for the data transfer between the processor and the MVI69-N2 module. The structure of the object is shown in the following illustration.



Name: N2Backplane

Description: Values used by program for data transfer operation between the module and the processor.

Members:

Data Type Size: 20 byte(s)

Name	Data Type	Style	Description
LastRead	INT	Decimal	Index of last read block
LastWrite	INT	Decimal	Index of last write block
BlockIndex	INT	Decimal	Computed block offset for data table
⊞ BootTimer	TIMER		

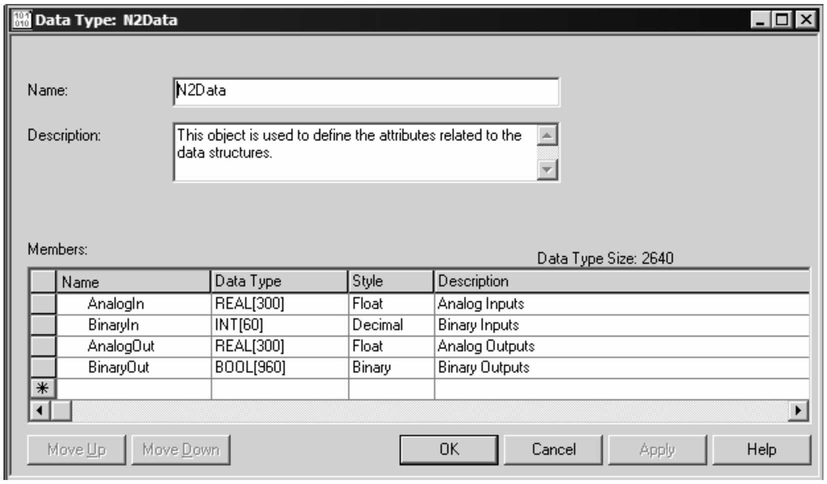
3.1.2 User Data Objects

These objects hold data to be transferred between the processor and the MVI69-N2 module. The user data is the read and write data transferred between the processor and the module as "pages" of data whose size depends on the block transfer size. The sample configuration file transfers 60 words at a time.

The write data (**N2Data**) is an array containing values that are written into the module database. The WriteData task places the write data into the output image for transfer to the module. This data is passed from the processor to the module for status and control information for use in other nodes on the network.

Module Data

This object contains the N2 data points to be used for convenience. The points are constantly moved to the Write Data array to be pushed out to the N2 module.



Name: N2Data

Description: This object is used to define the attributes related to the data structures.

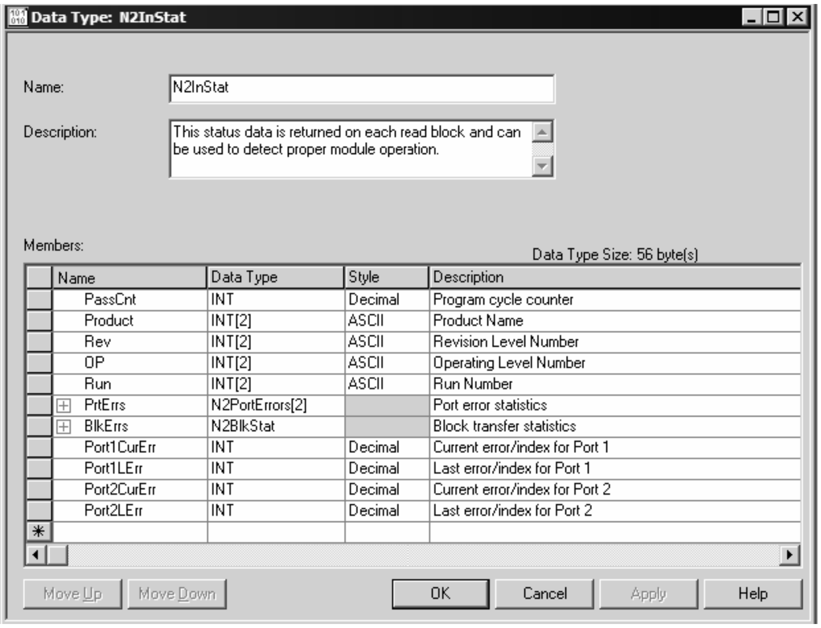
Members:

Data Type Size: 2640

Name	Data Type	Style	Description
AnalogIn	REAL[300]	Float	Analog Inputs
BinaryIn	INT[60]	Decimal	Binary Inputs
AnalogOut	REAL[300]	Float	Analog Outputs
BinaryOut	BOOL[960]	Binary	Binary Outputs

Status Data

This object views the status of the module. The **N2InStat** object shown is updated each time a read block (block 0 & -1) is received by the processor. Use this data to monitor the state of the module at a "real-time rate".



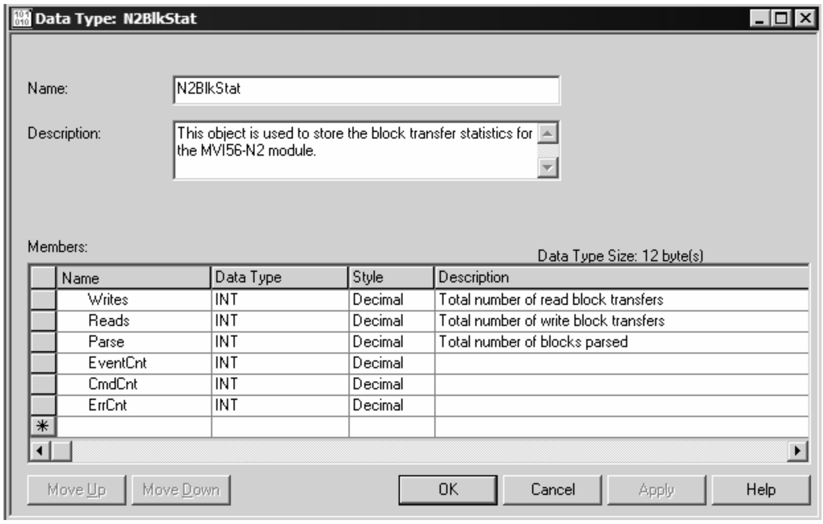
Name: N2InStat

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

Members: Data Type Size: 56 byte(s)

Name	Data Type	Style	Description
PassCnt	INT	Decimal	Program cycle counter
Product	INT[2]	ASCII	Product Name
Rev	INT[2]	ASCII	Revision Level Number
OP	INT[2]	ASCII	Operating Level Number
Run	INT[2]	ASCII	Run Number
⊞ PrtErrs	N2PortErrors[2]		Port error statistics
⊞ BlkErrs	N2BlkStat		Block transfer statistics
Port1CurErr	INT	Decimal	Current error/index for Port 1
Port1LErr	INT	Decimal	Last error/index for Port 1
Port2CurErr	INT	Decimal	Current error/index for Port 2
Port2LErr	INT	Decimal	Last error/index for Port 2

This object contains a structure that includes the status information for the data transfer operations between the processor and the module. The structure of this object is shown in the following example:



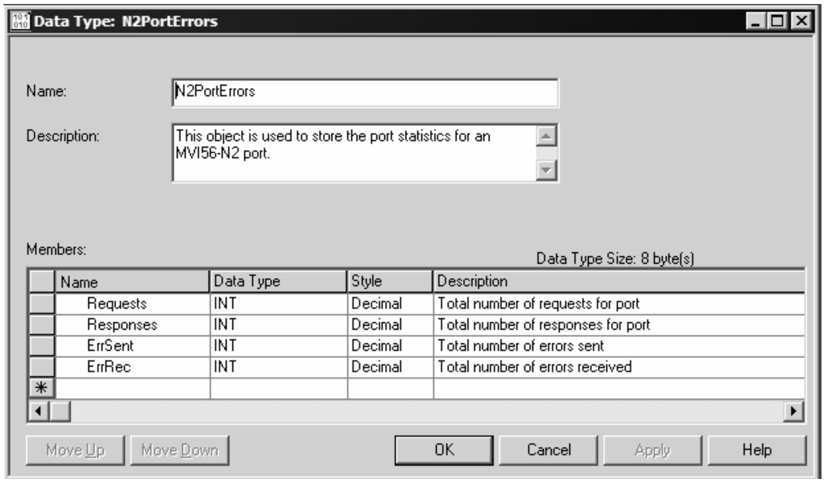
Name: N2BlkStat

Description: This object is used to store the block transfer statistics for the MVI56-N2 module.

Members: Data Type Size: 12 byte(s)

Name	Data Type	Style	Description
Writes	INT	Decimal	Total number of read block transfers
Reads	INT	Decimal	Total number of write block transfers
Parse	INT	Decimal	Total number of blocks parsed
EventCnt	INT	Decimal	
CmdCnt	INT	Decimal	
ErrCnt	INT	Decimal	

Additionally, the status object contains a structure for each of the servers in the module. The structure used for each server is shown in the following example:



Name: N2PortErrors

Description: This object is used to store the port statistics for an MVI56-N2 port.

Members:

Data Type Size: 8 byte(s)

Name	Data Type	Style	Description
Requests	INT	Decimal	Total number of requests for port
Responses	INT	Decimal	Total number of responses for port
ErrSent	INT	Decimal	Total number of errors sent
ErrRec	INT	Decimal	Total number of errors received

Refer to MVI69-N2 Status Data Area for a complete listing of the data stored in status object.

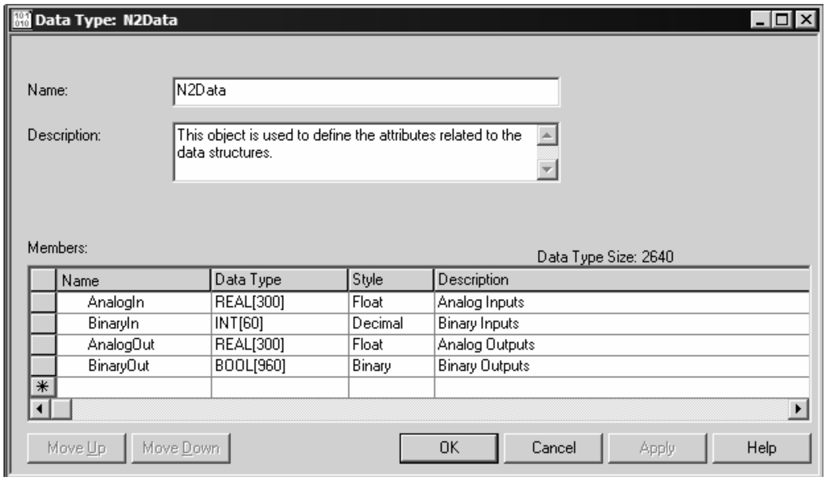
N2 Message Data

This module's program also includes the pass-through mode. The module supports two types of pass-through operation. Block 9001 is for binary output writes from an N2 master. Block 9002 is for analog output writes from an N2 Master.

In this mode, write messages sent to a slave port are passed directly through to the processor. It is the responsibility of the ladder logic to process the message received using this feature. This information is passed from the module to the processor using block identification codes of 9001 and 9002. The following topics explain the data objects involved.

N2 Data Object (N2Data)

These objects hold data transferred from the MVI69-N2 module. The **N2Data** object shown in the example is updated each time a read block number of 9001 or 9002 is received by the processor.



Name: N2Data

Description: This object is used to define the attributes related to the data structures.

Members: Data Type Size: 2640

Name	Data Type	Style	Description
AnalogIn	REAL[300]	Float	Analog Inputs
BinaryIn	INT[60]	Decimal	Binary Inputs
AnalogOut	REAL[300]	Float	Analog Outputs
BinaryOut	BOOL[960]	Binary	Binary Outputs

N2 Data Object

Network Point Type	Network Point Address	Unit	Description	Notes
AI	1	User Defined in Ladder Program	User Defined in Ladder Program	Actual number of points supported is defined in configuration file, for example Analog Input Object Count: 16
AI	…	User Defined in Ladder Program	User Defined in Ladder Program	Actual number of points supported is defined in configuration file. for example Analog Input Object Count: 16
AI	300	User Defined in Ladder Program	User Defined in Ladder Program	Actual number of points supported is defined in configuration file. for example Analog Input Object Count: 16
BI	1	User Defined in Ladder Program	User Defined in Ladder Program	Actual number of points supported is defined in configuration file. for example Binary Input Object Count: 16
BI	...	User Defined in Ladder Program	User Defined in Ladder Program	Actual number of points supported is defined in configuration file. for example Binary Input Object Count: 16
BI	960	User Defined in Ladder Program	User Defined in Ladder Program	Actual number of points supported is defined in configuration file. for example Binary Input Object Count: 16

Network Point Type	Network Point Address	Unit	Description	Notes
AO	1	User Defined in Ladder Program	User Defined in Ladder Program	Actual number of points supported is defined in configuration file. for example Analog Output Object Count: 16
AO	...	User Defined in Ladder Program	User Defined in Ladder Program	Actual number of points supported is defined in configuration file. for example Analog Output Object Count: 16
AO	256	User Defined in Ladder Program	User Defined in Ladder Program	Actual number of points supported is defined in configuration file. for example Analog Output Object Count: 16
BO	1	User Defined in Ladder Program	User Defined in Ladder Program	Actual number of points supported is defined in configuration file. That is, Binary Output Object Count: 16
BO	...	User Defined in Ladder Program	User Defined in Ladder Program	Actual number of points supported is defined in configuration file. for example Analog Output Object Count: 16
BO	960	User Defined in Ladder Program	User Defined in Ladder Program	Actual number of points supported is defined in configuration file. for example Analog Output Object Count: 16

ADF points are not supported. A write to an ADF point will return A. A read from an ADF point will return a status of unreliable and data of zero.

ADI points are not supported. A write to an ADI point will return A. A read from an ADI point will return a status of unreliable and data of zero.

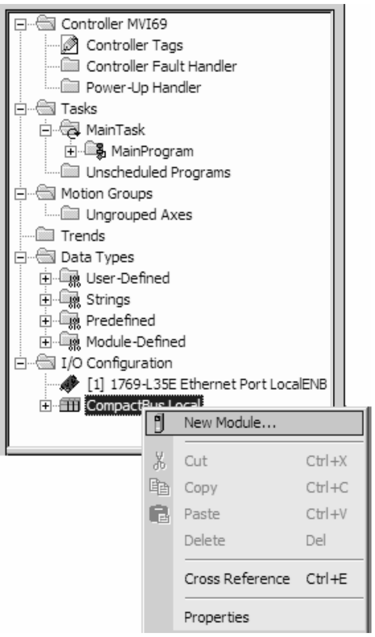
BD points are not supported. A write to an BD point will return A. A read from an BD point will return a status of unreliable and data of zero.

3.2 Adding the Module to an Existing CompactLogix Project

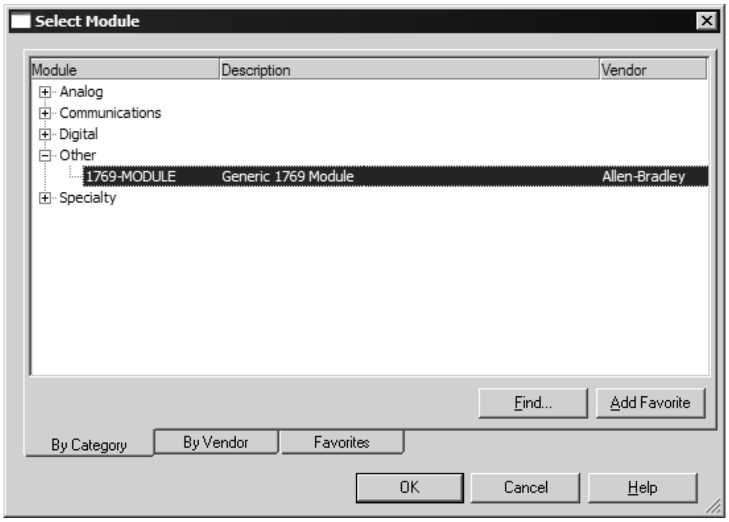
Important: The following steps describe how to install and configure the MVI69-N2 module with RSLogix 5000 version 15 or older. If you are using RSLogix 5000 version 16, please refer to Sample Add-On Instruction Import Procedure.
Important: The MVI69-N2 module has a power supply distance rating of 2 (L43 and L45 installations on first 2 slots of 1769 bus)

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

1 **Add the** MVI69-N2 **module to the project.** Right-click the mouse button on the I/O Configuration option in the Controller Organization window to display a pop-up menu. Select the New Module option from the I/O Configuration menu.

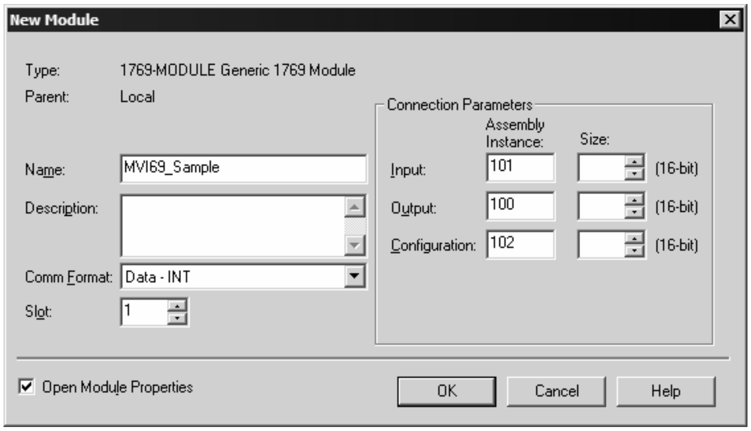


This action opens the following dialog box:



Module	Description	Vendor
⊞ Analog		
⊞ Communications		
⊞ Digital		
⊟ Other		
1769-MODULE	Generic 1769 Module	Allen-Bradley
⊞ Specialty		

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



Type: 1769-MODULE Generic 1769 Module
Parent: Local
Name: MVI69_Sample
Description:
Comm Format: Data - INT
Slot: 1

	Assembly Instance:	Size:
Input:	101	
Output:	100	
Configuration:	102	

☑ Open Module Properties

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

4 Configure the Connection Parameters to match to the Block Transfer Size parameter in the configuration file. Use the values in the table corresponding with the block transfer size you configured.

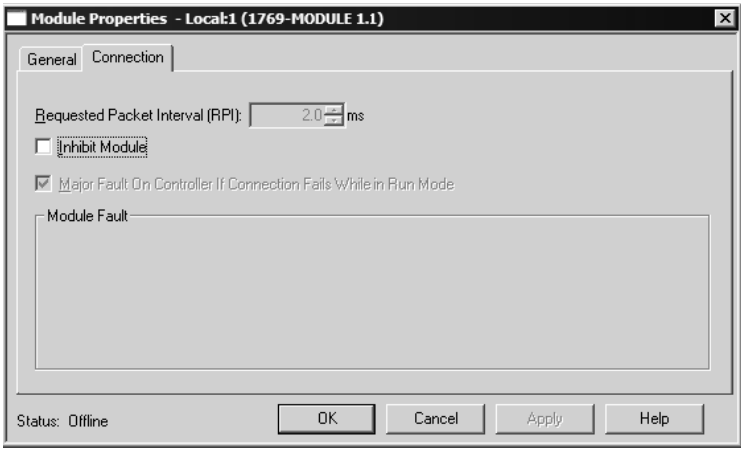
Block Transfer Size = 60	
Field	**Recommended Value**
Type	1769-MODULE Generic 1769 Module
Parent	Local
Name	MVI69
Description	MVI69 Application Module
Comm Format	Data - INT

Block Transfer Size = 60	
Field	**Recommended Value**
Slot	The slot number in the rack where the module is installed
Input Assembly Instance	101
Input Size	62
Output Assembly Instance	100
Output Size	61
Configuration Assembly Instance	102
Configuration Size	0

Block Transfer Size = 120	
Field	**Recommended Value**
Type	1769-MODULE Generic 1769 Module
Parent	Local
Name	MVI69
Description	MVI69 Application Module
Comm Format	Data - INT
Slot	The slot number in the rack where the module is installed
Input Assembly Instance	101
Input Size	122
Output Assembly Instance	100
Output Size	121
Configuration Assembly Instance	102
Configuration Size	0

Block Transfer Size = 240	
Field	**Recommended Value**
Type	1769-MODULE Generic 1769 Module
Parent	Local
Name	MVI69
Description	MVI69 Application Module
Comm Format	Data - INT
Slot	The slot number in the rack where the module is installed
Input Assembly Instance	101
Input Size	242
Output Assembly Instance	100
Output Size	241
Configuration Assembly Instance	102
Configuration Size	0

5 Click **Next** to continue.



Module Properties - Local:1 (1769-MODULE 1.1)

General | Connection

Requested Packet Interval (RPI): 2.0 ms

□ Inhibit Module

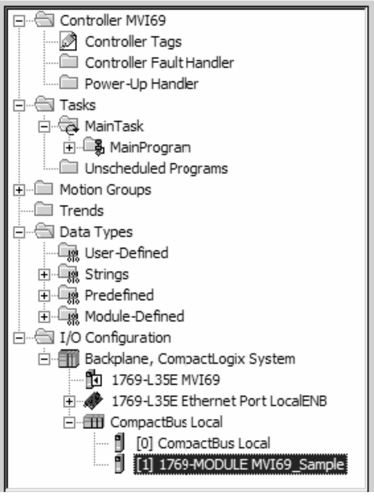
☑ Major Fault On Controller If Connection Fails While in Run Mode

Module Fault

Status: Offline

6 Select the Request Packet Interval value for scanning the I/O on the module. This value represents the minimum frequency the module will handle scheduled events. This value should not be set to less than 1 millisecond. Values between 1 and 10 milliseconds should work with most applications.

7 Save the module. Click OK to dismiss the dialog box. The Controller Organization window now displays the module's presence. The following illustration shows the Controller Organization window:



```
Controller MVI69
    Controller Tags
    Controller Fault Handler
    Power-Up Handler
Tasks
    MainTask
        MainProgram
    Unscheduled Programs
Motion Groups
Trends
Data Types
    User-Defined
    Strings
    Predefined
    Module-Defined
I/O Configuration
    Backplane, CompactLogix System
        1769-L35E MVI69
        1769-L35E Ethernet Port LocalENB
        CompactBus Local
            [0] CompactBus Local
            [1] 1769-MODULE MVI69_Sample
```

8 Copy the Controller Tags from the sample program.

9 Copy the User Defined Data Types from the sample program.

10 Copy the Ladder Rungs from the sample program.

11 Save and Download the new application to the controller and place the processor in run mode.

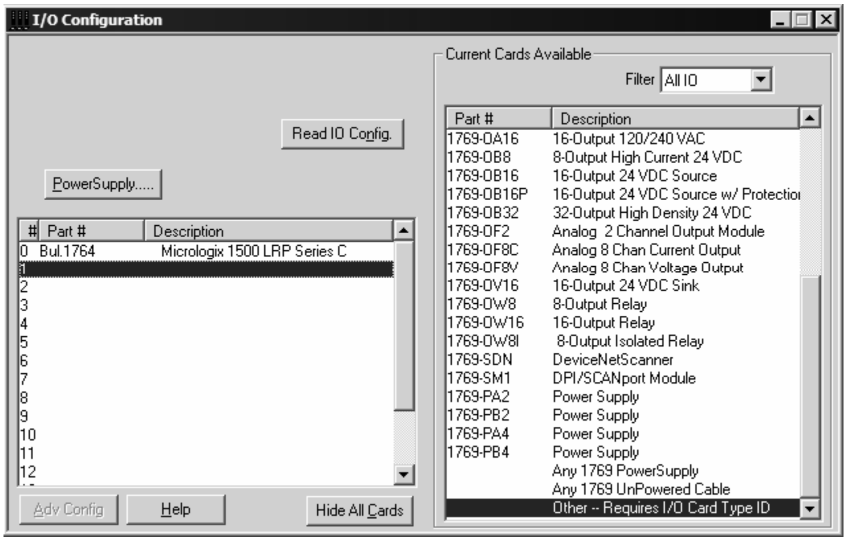
3.3 Adding the Module to an Existing MicroLogix Project

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

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

1 In RSLogix, open your existing application, or start a new application, depending on your requirements.

2 Double-click the I/O Configuration icon located in the Controller folder in the project tree. This action opens the I/O Configuration dialog box.



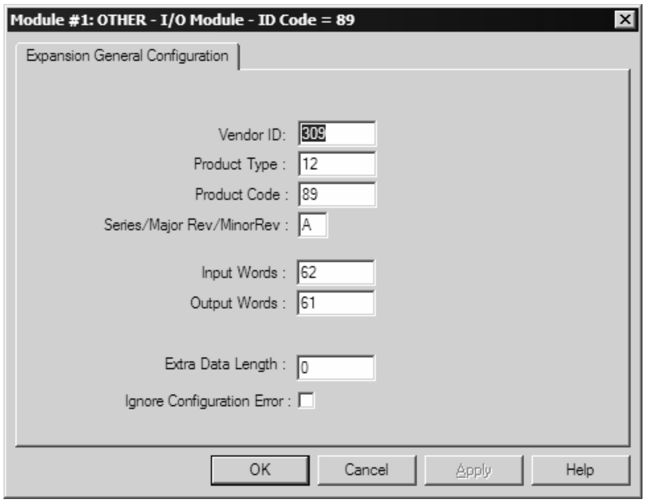
Filter: All IO

#	Part #	Description
0	Bul.1764	Micrologix 1500 LRP Series C
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		

Part #	Description
1769-OA16	16-Output 120/240 VAC
1769-OB8	8-Output High Current 24 VDC
1769-OB16	16-Output 24 VDC Source
1769-OB16P	16-Output 24 VDC Source w/ Protectio
1769-OB32	32-Output High Density 24 VDC
1769-OF2	Analog 2 Channel Output Module
1769-OF8C	Analog 8 Chan Current Output
1769-OF8V	Analog 8 Chan Voltage Output
1769-OV16	16-Output 24 VDC Sink
1769-OW8	8-Output Relay
1769-OW16	16-Output Relay
1769-OW8I	8-Output Isolated Relay
1769-SDN	DeviceNetScanner
1769-SM1	DPI/SCANport Module
1769-PA2	Power Supply
1769-PB2	Power Supply
1769-PA4	Power Supply
1769-PB4	Power Supply
	Any 1769 PowerSupply
	Any 1769 UnPowered Cable
	Other -- Requires I/O Card Type ID

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

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



Module #1: OTHER - I/O Module - ID Code = 89

Expansion General Configuration

Vendor ID: 309
Product Type : 12
Product Code : 89
Series/Major Rev/MinorRev : A

Input Words : 62
Output Words : 61

Extra Data Length : 0
Ignore Configuration Error : □

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

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

5 Click **Next** to continue.

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

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

Download the new application to the controller and place the processor in run mode. If you encounter errors, refer to **Diagnostics and Troubleshooting** (page 43) for information on how to connect to the module's Config/Debug port to use its troubleshooting features.

4 Diagnostics and Troubleshooting

In This Chapter

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

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

4.1 Reading Status Data from the Module

The MVI69-N2 module returns a 30-word Status Data Block that may be used to determine the module's operating status. This data is located in the module's database in registers 6000 through 6029.

This data is transferred to the CompactLogix or MicroLogix processor every scan when Binary Output and Analog Output data is not present (Blocks 0 and -1).

The Configuration/Debug port provides the following functionality:

- Full view of the module's configuration data
- View of the module's status data
- Complete display of the module's internal database (registers 0 to 10000)
- Version Information
- Control over the module (warm boot, cold boot, transfer configuration)

4.1.1 Required Hardware

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

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

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

4.1.2 The Configuration/Debug Menu

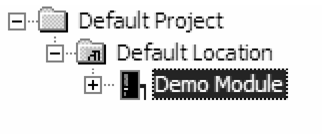
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

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

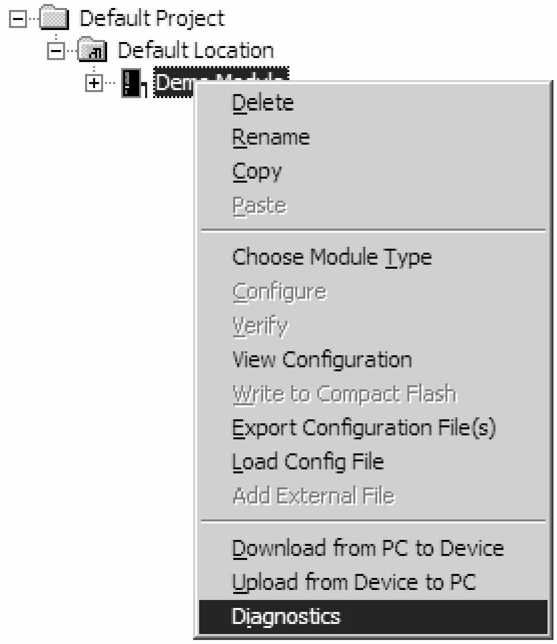
Using the Diagnostic Window in ProSoft Configuration Builder

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

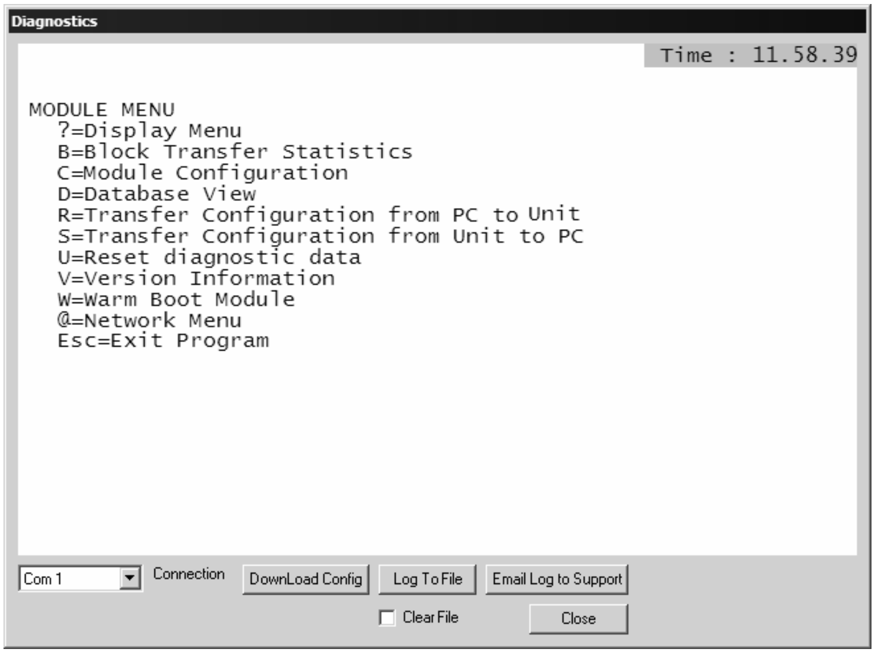
1 Start PCB program with the application file to be tested. Right click over the module icon.



2 On the shortcut menu, choose Diagnostics.



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



```
                                                        Time : 11.58.39

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

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

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

1 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.

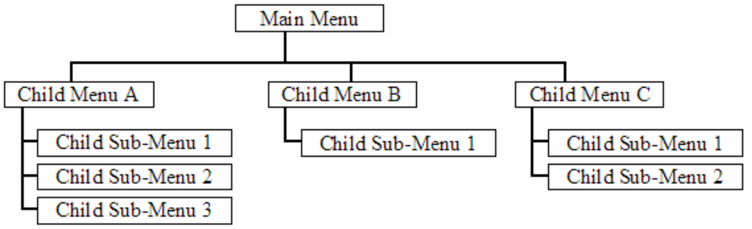
2 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

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

Navigation

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

The organization of the menu structure is represented in simplified form in the following illustration:



The remainder of this section shows you the menus available for this module, and briefly discusses the commands available to you.

Keystrokes

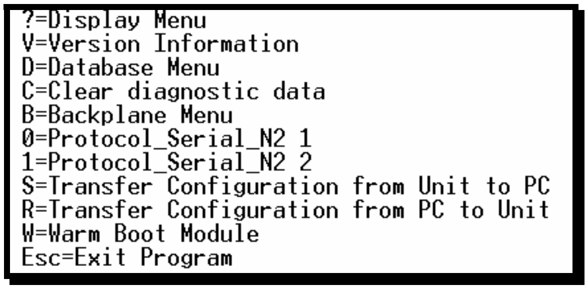
The keyboard commands on these menus are almost always non-case sensitive. You can enter most commands in lower case or capital letters.

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

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

4.1.3 Main menu

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



```
?=Display Menu
V=Version Information
D=Database Menu
C=Clear diagnostic data
B=Backplane Menu
0=Protocol_Serial_N2 1
1=Protocol_Serial_N2 2
S=Transfer Configuration from Unit to PC
R=Transfer Configuration from PC to Unit
W=Warm Boot Module
Esc=Exit Program
```

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

Viewing Version Information

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

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

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

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

Opening the Database Menu

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

Resetting diagnostic data

Press **[C]** to reset the status counters for the client and servers in the module.

Opening the Backplane Menu

Press **[B]** from the Main Menu to view the Backplane Data Exchange List. Use this command to display the configuration and statistics of the backplane data transfer operations.

Tip: Repeat this command at one-second intervals to determine the number of blocks transferred each second.

Opening the Protocol_Serial_N2 Menu

Press **[0]** or **[1]** from the Main Menu to open the Protocol_Serial_N2 menu for N2 Ports 1 and 2.

Use this command to view communication status and statistics for the selected port. This information can be useful for trouble-shooting communication problems.

Sending the Configuration File

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

Receiving the Configuration File

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

Warm Booting the Module

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

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

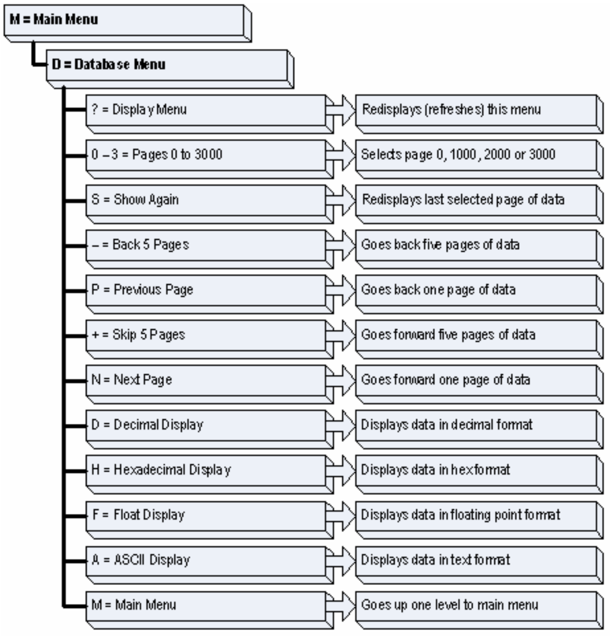
Exiting the Program

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

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

4.1.4 Database View Menu

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


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

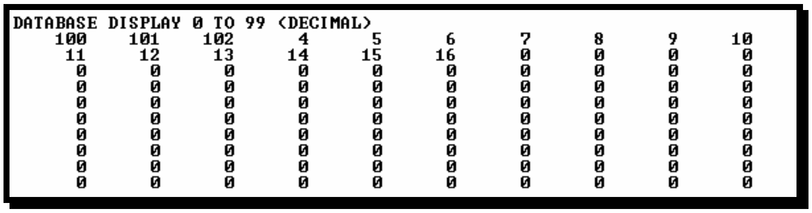
Viewing Register Pages

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

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

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

Displaying the Current Page of Registers Again



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

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

Moving Back Through 5 Pages of Registers

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

Viewing the Previous 100 Registers of Data

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

Skipping 500 Registers of Data

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

Viewing the Next 100 Registers of Data

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

Viewing Data in Decimal Format

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

Viewing Data in Hexadecimal Format

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

Viewing Data in Floating Point Format

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

Viewing Data in ASCII (Text) Format

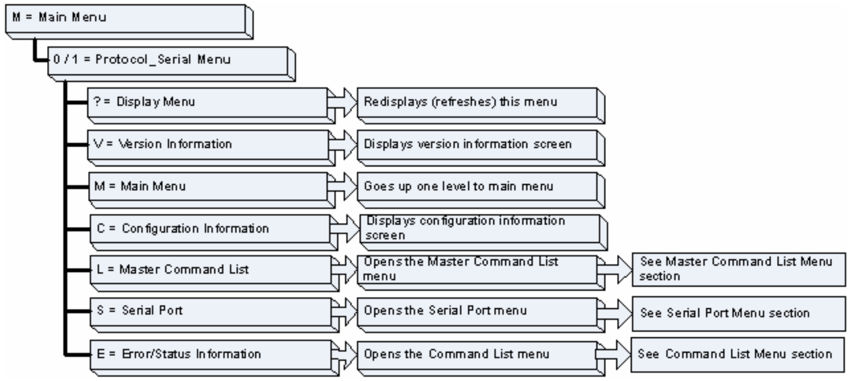
Press **[A]** to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Returning to the Main Menu

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

4.1.5 Protocol Serial N2 Menu

Press **[0]** or **[1]** to view protocol serial information for ports 1 and 2, respectively. Use this command to view a variety of error and status screens for the port. Press **[?]** to view a list of commands available on this menu.


M = Main Menu
0 / 1 = Protocol_Serial Menu
? = Display Menu
Redisplays (refreshes) this menu
V = Version Information
Displays version information screen
M = Main Menu
Goes up one level to main menu
C = Configuration Information
Displays configuration information screen
L = Master Command List
Opens the Master Command List menu
See Master Command List Menu section
S = Serial Port
Opens the Serial Port menu
See Serial Port Menu section
E = Error/Status Information
Opens the Command List menu
See Command List Menu section

Redisplaying the Menu

Press **[?]** to display the current menu. Use this command when you are looking at a screen of data, and want to view the menu choices available to you.

Viewing Version Information

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

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

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

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

Returning to the Main Menu

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

Viewing Configuration Information

Press **[C]** to view configuration information for the selected port, protocol, driver or device.

Opening the Command List Menu

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

Opening the Serial Port Menu

Press **[S]** to open the Serial Port menu. Use this command to view and change additional serial port driver settings.

Viewing Error and Status Data

Press **[E]** to display the error/status data for the module.

4.2 LED Status Indicators

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

Module	Color	Status	Indication
CFG	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
P1	Green	On	Data being transferred between master and port 1.
		Off	No data
P2	Green	On	Data being transferred between master and port 2.
		Off	No data
APP Status	Amber	Off	The MVI69-N2 module program has recognized a communication error.
		On	The MVI69-N2 is working normally.
BP ACT	Amber	On	The LED is on when the module is performing a write operation on the backplane.
		Off	The LED is off when the module is performing a read operation on the backplane. Under normal operation, the LED should blink rapidly on and off.
OK	Red/ Green	Off	The card is not receiving any power and is not securely plugged into the rack.
		Green	The module is operating normally.
		Red	The program has detected an error or is being configured. If the LED remains red for over 10 seconds, the program has probably halted. Remove the card from the rack and re-insert the card to restart the module's program.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or battery is not present. Allow battery to charge by keeping module plugged into rack for 24 hours. If BAT LED still does not go off, contact ProSoft Technology, as this is not a user serviceable item.

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

4.2.1 Clearing a Fault Condition

Typically, if the OK LED on the front of the module becomes illuminated red for over ten seconds, a hardware problem has been detected in the module or the program has exited. To attempt to clear the condition:

1. Turn off the power to the rack
2. Remove the card from the rack
3. Make certain all jumpers are set correctly
4. Re-insert the card in the rack and turn the power back on
5. Verify the configuration data being transferred to the module from the CompactLogix or MicroLogix processor

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

4.2.2 Troubleshooting

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

Processor Errors

Problem Description	Steps to take
Processor Fault	Verify that the module is plugged into the slot that has been configured for the module. Verify that the slot in the rack configuration has been set up correctly in the ladder logic.
Processor I/O LED flashes	This indicates a problem with backplane communications. Verify that all modules in the rack are configured in the ladder logic. Module has a power supply distance rating of 2 on Compact Logix. The module must be within 2 slots of the power supply on Compact Logix, or that the MicroLogix backplane can supply the 800ma required for the module.

Module Errors

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

5 Reference

In This Chapter

5.1 Product Specifications

The MVI69 Johnson Controls N2 Communication Module allows CompactLogix I/O compatible processors to interface easily with other Johnson Controls N2 master devices.

The MVI69-N2 module acts as an input/output module between the Johnson Controls Metasys network and the CompactLogix backplane. The module acts as a slave receiving commands from a master device. The data transfer from the CompactLogix processor is asynchronous from the actions on the Johnson Controls Metasys network. An internal database in the module exchanges data between the processor and the Johnson Controls Metasys Master (NCM, N-30, NAE, NIE).

The MVI69-N2 module acts as an input/output module between the Johnson Controls Metasys network and the CompactLogix backplane. The module acts as a slave receiving commands from a master device. The data transfer from the CompactLogix processor is asynchronous from the actions on the Johnson Controls Metasys network. An internal database in the module exchanges data between the processor and the Johnson Controls Metasys Master (NCM, N-30, NAE, NIE).

5.1.1 General Specifications

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

5.1.2 Hardware Specifications

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

5.1.3 Functional Specifications

Some of the general specifications include:

- Support for the storage and transfer of internal database registers to/from the CompactLogix processor's controller tags
- Two ports to emulate a Johnson Controls N2 slave
- Supports the following N2 objects:
  - Binary Input: Up to 960 points / 60 words
  - Analog Input: Up to 300 points / 600 words
  - Binary Output: Up to 960 points / 60 words
  - Analog Output: Up to 256 points / 512 words
- Supported Commands/Sub-commands
  - 0/4: Poll Message No Acknowledge
  - 0/5: Poll Message with Acknowledge
  - 0/9: Status Update
  - 1/1: Read Analog Input Attributes
  - 1/2: Read Binary Input Attributes
  - 1/3: Read Analog Output Attributes
  - 1/4: Read Binary Output Attributes
  - 2/1: Write Analog Input Attributes
  - 2/2: Write Binary Input Attributes
  - 2/3: Write Analog Output Attributes

- 2/4: Write Binary Output Attributes
- 7/2/3: Override Analog Output
- 7/2/4: Override Binary Output
- F: Identify Device Type

- The following commands are recognized, and acknowledged, but are not communicated in any way to the CompactLogix, and do not return any data:
  - 0/0: Time Update
  - 0/8: Warm Start
  - All other commands return a Bad Command Error Code
- Configurable through the configuration file for the following:
  - Slave Address (assignable individually for Port 1 and 2)
  - Analog Input Object Count
  - Binary Input Object Count
  - Analog Output Object Count
  - Binary Output Object Count
- Warning and Alarming functions performed on Analog Input and Binary Input data types
- Change of State Response buffering
- Communication status error codes and statistics returned per port
- Communication parameter: Baud rate 9600 bps

Memory mapping is pre-assigned to optimize data access and to ease implementation

A port configured as a Johnson Controls N2 slave permits a remote master to interact with data contained in the module. This data is derived from the CompactLogix processor.

5.2 Functional Overview

This section provides an overview of how the MVI69-N2 module transfers data using the N2 protocol. You should understand the important concepts in this chapter before you begin installing and configuring the module.

5.2.1 General Concepts

The following discussion explains several concepts that are important for understanding the operation of the MVI69-N2 module.

Module Power Up

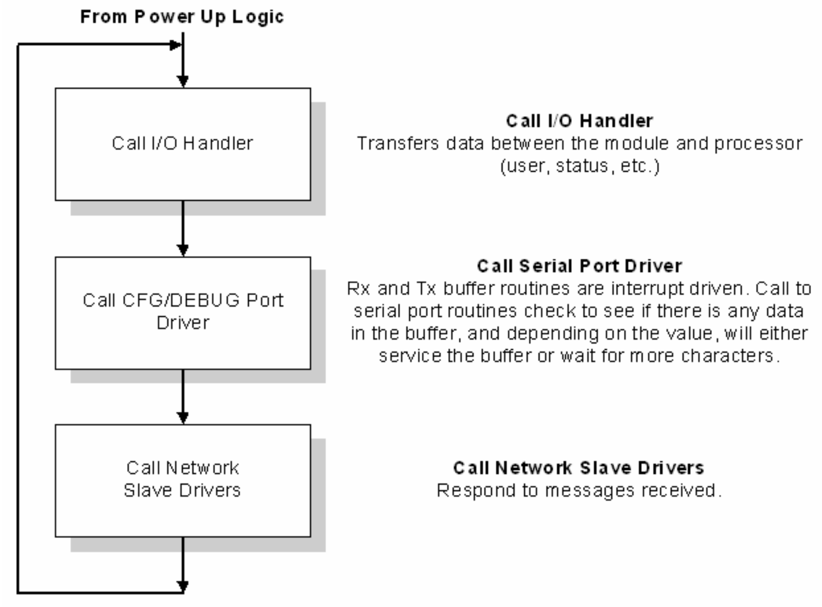
On power up the module begins performing the following logical functions:

1. Initialize hardware components:
   - Initialize CompactLogix or MicroLogix backplane driver.
   - Test and clear all RAM.
   - Initialize the serial communication ports.
2. Initialize module register space.
3. Enable slave driver on selected ports.

After the module is configured, the module begins receiving and transmitting messages with Johnson Controls N2 devices on the network.

Main Logic Loop

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


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

CompactLogix or MicroLogix Processor Not in Run

Whenever the module detects that the processor has gone out of the Run mode (that is, Fault or PGM), the Johnson Controls N2 ports can be shut down as prescribed in the user configuration. When the processor is returned to a running state, the module resumes communications on the network.

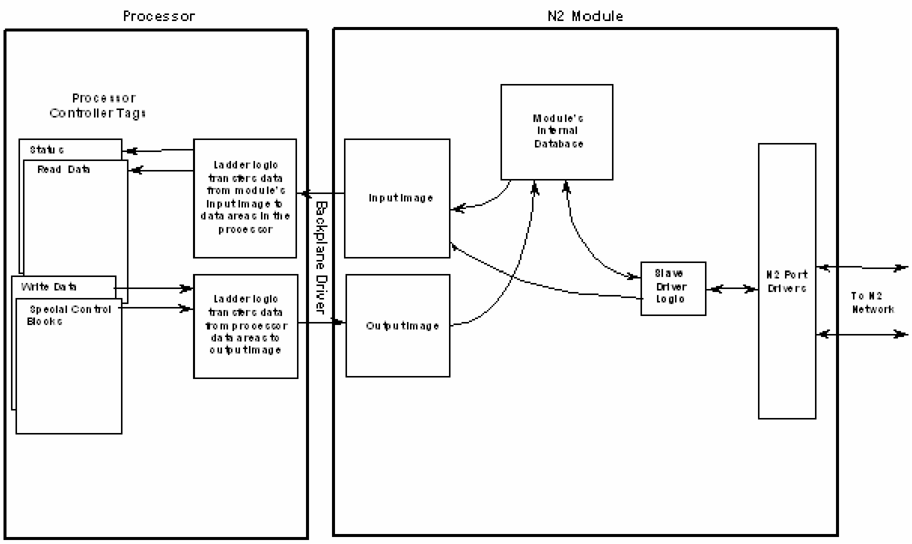
Backplane Data Transfer

The MVI69-N2 module communicates directly over the CompactLogix or MicroLogix backplane. Data is paged between the module and the CompactLogix or MicroLogix 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 the user for the module and the communication load on the module. Typical updates are in the range of 2 to 10 milliseconds.

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 *n* words, where *n* = 60, 120 or 240.

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 *n* words, where *n* = 60, 120 or 240.

The following illustration shows the data transfer method used to move data between the CompactLogix or MicroLogix processor, the MVI69-N2 module, and the Johnson Controls N2 network.


Processor
N2 Module
Processor Controller Tags
Status
Read Data
Write Data
Special Control Blocks
Ladder logic transfers data from module's input image to data areas in the processor
Ladder logic transfers data from processor data areas to output image
Backplane Driver
Input Image
Output Image
Module's Internal Database
Slave Driver Logic
N2 Port Drivers
To N2 Network

As shown in the diagram, all data transferred between the module and the processor over the backplane is through the input and output images. Ladder logic must be written in the CompactLogix or MicroLogix 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 database.

MVI69-N2 Database Map

Start Address	End Address	Data
0	59	Binary Inputs
200	799	Analog Inputs
5000	5011	Error / Status Table

Data contained in this database is paged through the input and output images by coordination of the CompactLogix or MicroLogix ladder logic and the MVI69-N2 module's program. Up to 242 words of data can be transferred from the module to the processor at a time. Up to 241 words of data can be transferred from the processor to the module. Each image has a defined structure depending on the data content and the function of the data transfer described in the following topics.

Normal Data Transfer

Normal data transfer includes the paging of the user data found in the module's internal database and the Status Data. These data are transferred through read (input image) and write (output image) blocks. Refer to the Module Configuration section for a description of the data objects used with the blocks. The structure and function of each block is discussed in the following topics.

Read Block

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

Offset	Description	Length
0	Read Block ID (0 or -1)	1
1	Write Block ID	1
2	Program Scan Counter	1
3 to 4	Product Name	2
5 to 6	Product Version	2
7 to 8	Operating System	2
9 to 10	Run Number	2
11 to 17	Port 1 Error Status	7
18 to 24	Port 2 Error Status	7
25 to 30	Data Transfer Status	6
31	Port 1 Current Error/Index	1
32	Port 1 Last Error/Index	1
33	Port 2 Current Error/Index	1
34	Port 2 Last Error/Index	1
35 to n	Not Used	20

where

n = 60, 120, or 240 depending on the Block Transfer Size parameter (refer to the configuration file).

The Read Block ID is an index value used to alert the ladder logic to the presence of new input data. A read block ID of 0 and -1 transfers status data, while a value of 9001 and 9002 indicates an output write has been issued by the N2 master. The number of data words per transfer depends on the configured Block Transfer Size parameter in the configuration file (possible values are 60, 120, or 240). The sample ladder works only for block size of 60, and it is recommended to leave this parameter as set in the default configuration file.

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

R0W1→R-1W2→R0W1→R-1W2→R0W1→R-1W2→R0W1→

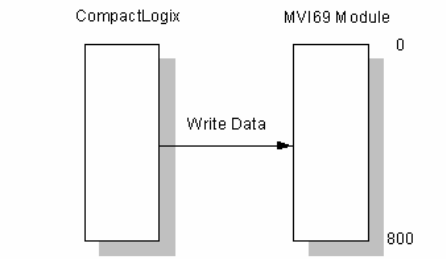
This sequence will continue until interrupted by other write block numbers sent by the controller or by a command request from a node on the N2 network or operator control through the module's Configuration/Debug port.

The following example shows a typical backplane communication application.

The backplane parameters are configured as follows:

```
Read Register Start:    0
Read Register Count:    0
Write Register Start:   0
Write Register Count: 800
```

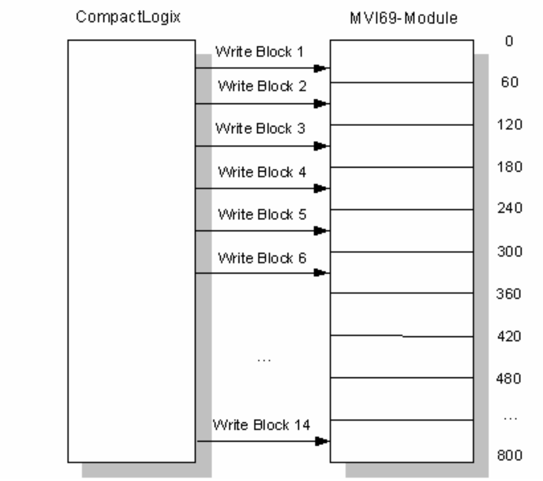
The backplane communication would be configured as follows:



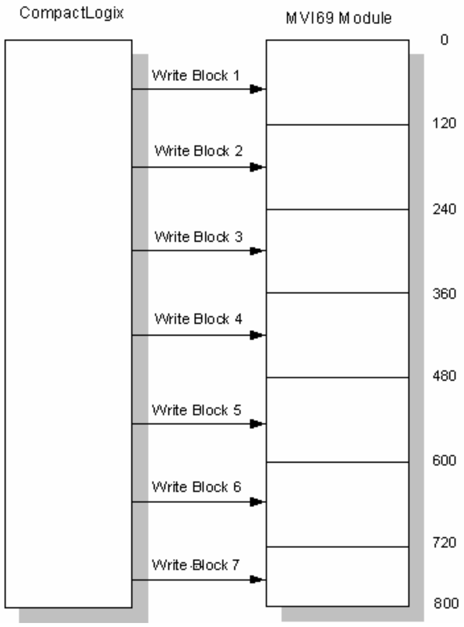
No data transfer takes place from the module to the processor except for pass-through data. Database address 0 to 800 will continuously be transferred from the processor to the module.

The Block Transfer Size parameter basically configures how the Read Data and Write Data areas are broken down into data blocks (60, 120, or 240).

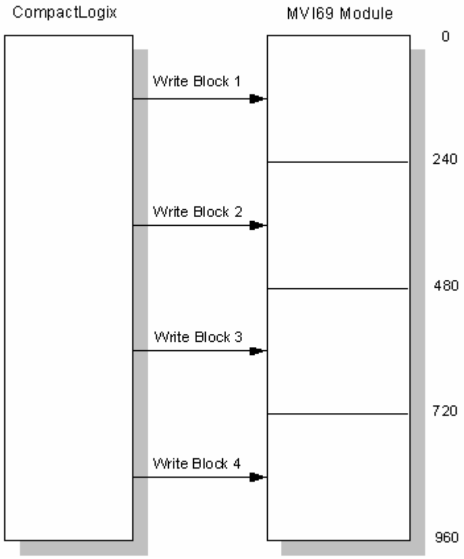
If Block Transfer Size = 60:



If Block Transfer Size = 120:



If Block Transfer Size = 240:



Write Block

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

Offset	Description	Length
0	Write Block ID	1
1 to n	Write Data	n

where *n* = 60, 120, or 240 depending on the Block Transfer Size parameter (refer to the configuration file).

The Write Block ID is an index value used to determine the location in the module's database where the data will be placed. Each transfer can move up to 240 words (block offsets 1 to 241) of data.

Warm Boot

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

Offset	Description	Length
0	9998	1
1 to n	Spare	n

Cold Boot

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

Offset	Description	Length
0	9999	1
1 to n	Spare	n

Pass-Through Control Blocks

The module will pass blocks with identification codes of 9001 and 9002 to the processor for each received function 7/2/4, and 7/2/3 command respectively. The structure of the pass-through control blocks are shown in the following tables.

Block 9001

Offset	Description	Length
0	9001	1
1	Binary Point Offset	1
2	Binary Value	1
3 to n	Spare	n

Block 9002

Offset	Description	Length
0	9002	1
1	Analog Point Offset	1
2	Analog Value	2
3 to n	Spare	n

The ladder logic should copy the received data and control the processor as expected by the master device. The processor must respond to the pass-through control block with the following format:

Offset	Description	Length
0	9001 and 9002	1
1 to n	Spare	n

This informs the module that the command has been processed and can be cleared from the pass-through queue.

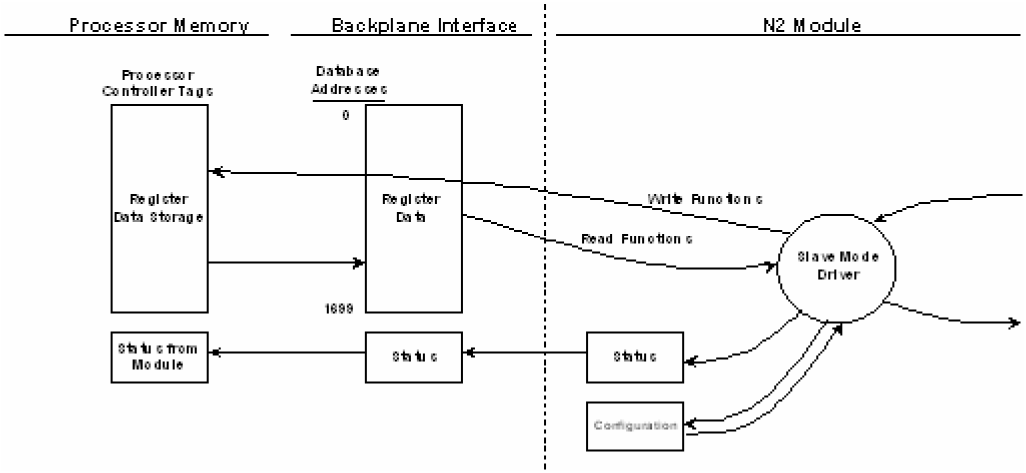
5.2.2 Data Flow Between the MVI69-N2 Module and the CompactLogix or MicroLogix Processor

The following section describes the flow of data between the two pieces of hardware (CompactLogix or MicroLogix processor and MVI69-N2 module). Each port on the module is configured to emulate a Johnson Controls N2 slave device. The operation of each port is dependent on this configuration.

Slave Driver

The Slave Driver Mode allows the MVI69-N2 module to respond to data read and write commands issued by a master on the Johnson Controls N2 network. The following flowchart describes the flow of data into and out of the module. The Module Setup section provides a list of the parameters that must be defined for a slave port.

All write requests will be passed directly to the processor. The following illustration shows the data flow for a slave port.



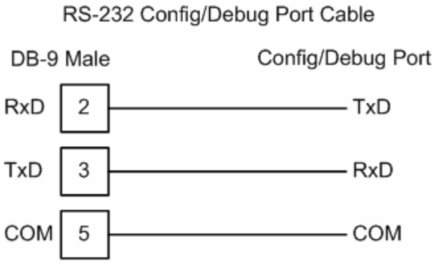
5.3 Cable Connections

The application ports on the MVI69-N2 module support RS-232, RS-422, and RS-485 interfaces. Please inspect the module to ensure that the jumpers are set correctly to correspond with the type of interface you are using.

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

5.3.1 RS-232 Configuration/Debug Port

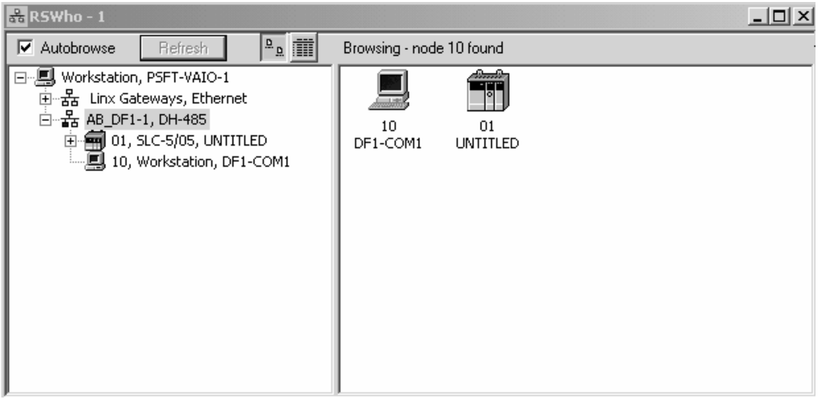
This port is physically an RJ45 connection. An RJ45 to DB-9 adapter cable is included with the module. This port permits a PC based terminal emulation program to view configuration and status data in the module and to control the module. The cable for communications on this port is shown in the following diagram:



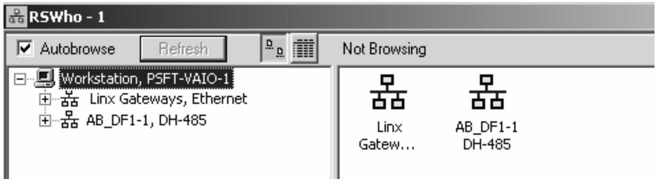
Disabling the RSLinx Driver for the Com Port on the PC

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

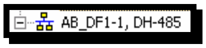
1 Open RSLinx and go to Communications>RSWho
2 Make sure that you are not actively browsing using the driver that you wish to stop. The following shows an actively browsed network:



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

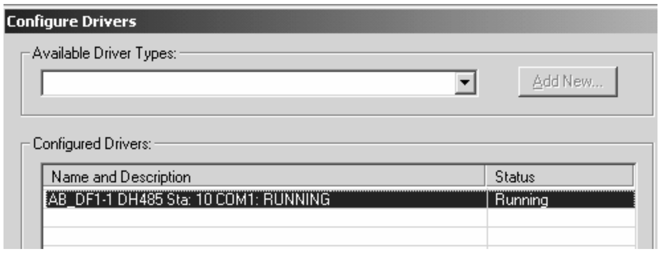


Branches are displayed or hidden by clicking on the ⊞ or the ⊟ icons.



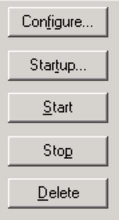
4 When you have verified that the driver is not being browsed, go to

Communications>Configure Drivers
You may see something like this:

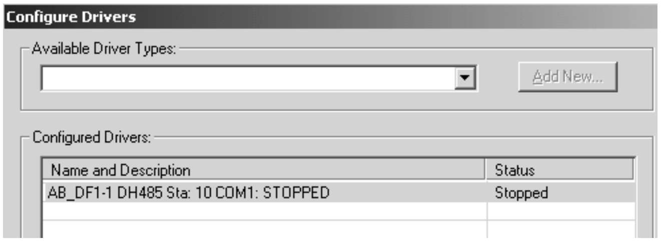


Name and Description	Status
AB_DF1-1 DH485 Sta: 10 COM1: RUNNING	Running

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



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



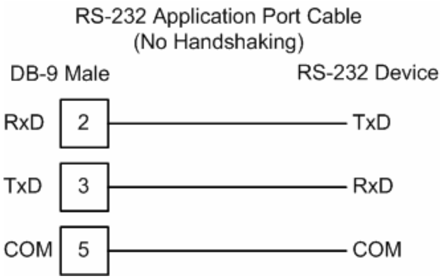
Name and Description	Status
AB_DF1-1 DH485 Sta: 10 COM1: STOPPED	Stopped

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

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

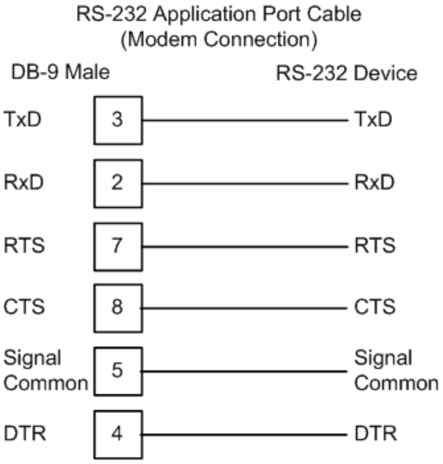
5.3.2 RS-232

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



RS-232: Modem Connection

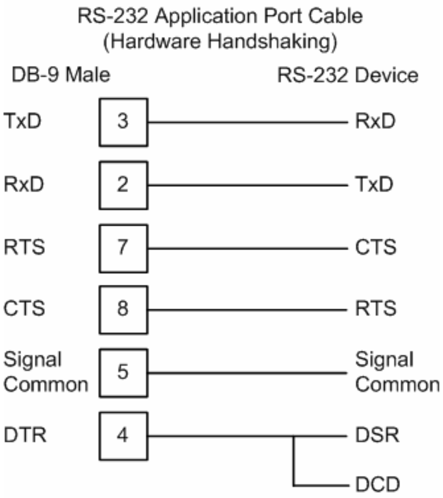
This type of connection is required between the module and a modem or other communication device.



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

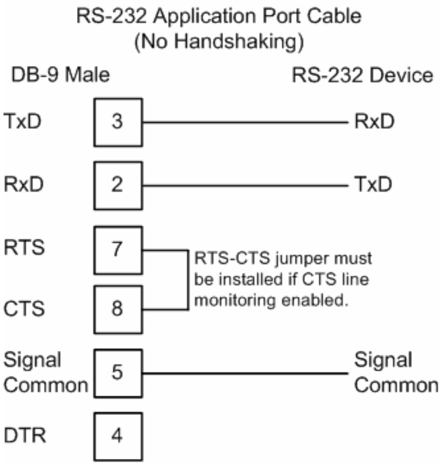
RS-232: Null Modem Connection (Hardware Handshaking)

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



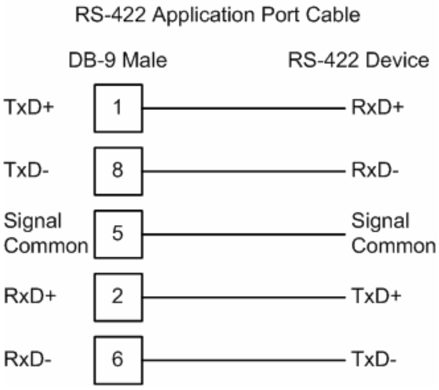
RS-232: Null Modem Connection (No Hardware Handshaking)

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


RS-232 Application Port Cable
(No Handshaking)
DB-9 Male
RS-232 Device
TxD
3
RxD
RxD
2
TxD
RTS
7
CTS
8
RTS-CTS jumper must be installed if CTS line monitoring enabled.
Signal Common
5
Signal Common
DTR
4

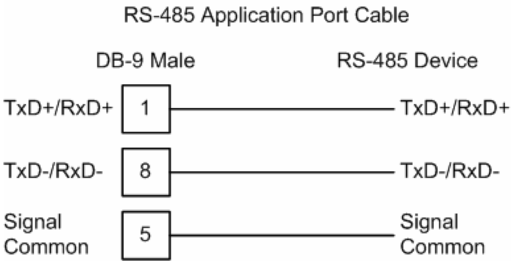
Note: If the port is configured with the "Use CTS Line" set to 'Y', then a jumper is required between the RTS and the CTS line on the module connection.

5.3.3 RS-422



5.3.4 RS-485

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

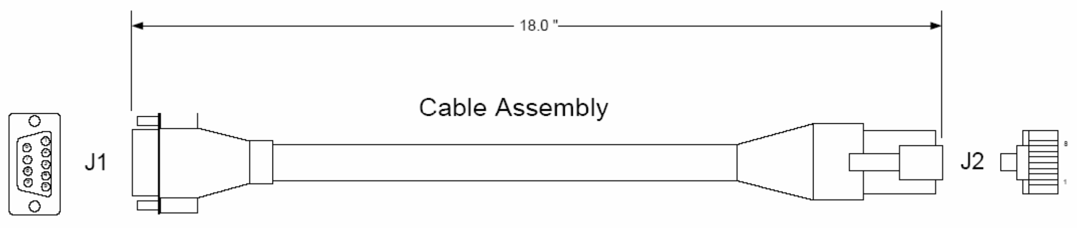


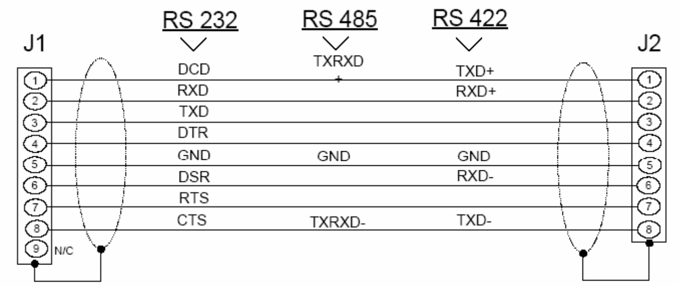
Note: Terminating resistors are generally not required on the RS-485 network, unless you are experiencing communication problems that can be attributed to signal echoes or reflections. In this case, install a 120 ohm terminating resistor on the RS-485 line.

RS-485 and RS-422 Tip

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

5.3.5 DB9 to RJ45 Adaptor (Cable 14)





Wiring Diagram

5.4 MVI69-N2 Status Data Area

This section contains a listing of the data contained in the MVI69-N2 status data object.

Offset	Content	Description
2	Program scan counter	This value is incremented each time a complete program cycle occurs in the module.
3 to 4	Product Name	
	Product Code as ASCII string	This register contains the product code of "N2"
5 to 6	Revision level as ASCII string	This register contains the product version for the current software.
7 to 9	Operating system level as ASCII string	This register contains the month and year values for the program operating system.
9 to 10	Run number as ASCII string	This register contains the run number value for the current software.
11 to 17	Number of requests - Port 1	Contains the number of port messages sent out of the port.
	Number of responses - Port 1	Contains the total number of messages received on the port.
	Number of errors sent - Port 1	Contains the total number of message errors sent out of the port.
	Number of errors received - Port 1	Contains the total number of message errors received on the port.
18 to 24	Number of requests - Port 2	Contains the total number of messages sent out the port.

Offset	Content	Description
	Number of responses - Port 2	Contains the total number of messages received on the port.
	Number of errors sent - Port 2	Contains the total number of message errors sent out the port.
	Number of errors received - Port 2	Contains the total number of message errors received on the port.
25 to 30	Backplane Transfer Writes	Contains the total number of write blocks transferred from the processor to the module.
	Backplane Transfer Reads	Contains the total number of read blocks transferred from the module to the processor.
	Backplane Blocks Parsed	
	Backplane Transfer Errors	Contains the total number of block errors recognized by the module.
31	Current Error - Port 1	For a slave port, this field contains the value of the current error code returned. For a master port, this field contains the index of the currently executing command.
32	Last Error - Port 1	For a slave port, this field contains the value of the last error code returned. For a master port, this field contains the index of the command with the error.
33	Current Error - Port 2	For a slave port, this field contains the value of the current error code returned. For a master port, this field contains the index of the current executing command.
34	Last Error - Port 2	For a slave port, this field contains the value of the last error code returned. For a master port, this field contains the index of the command with an error.

A data file should be allocated in the ladder logic to hold this block of data. The size of the block of data is 35 words.

5.5 N2 Error Codes

The Slave Error Table is a 20 word block. The location of the Error Table is determined by the Slave Error Table Pointer parameter in the Configuration Block. The structure of the data block is as follows:

Word	Description
Port 1	
0	Current port status
1	Last error condition
2	Total Messages to this slave
3	Total Msg responses from this slave
4	Total Msgs seen by this slave
Port 2	
5	Current port status
6	Last error condition
7	Total Messages to this slave
8	Total Msg responses from this slave

Word	Description
9	Total Msgs seen by this slave
System Information	
10-11	Product Name (ASCII)
12-13	Revision (ASCII)
14	Operating System Rev(ASCII)
15	Production Run Number (ASCII)
16-19	Spare

5.5.1 Current Port Error Status

This value represents the current value of the error code for the port. This value will only be valid if the port is configured as a Slave. The possible values are described in the following section.

Code	Description
0	All OK
1	Bad Command
2	Checksum Error
3	Recv buffer overflow
5	Command Data Error
16	Data Register Addr out of range
17	Field Data Error
18	Command Rejected
20	Unit not Warm Started
255	TX Timeout – RTS/CTS jumper missing

5.5.2 Last Error Code

This value is the last error code transmitted to the master by this slave port.

5.5.3 Total Messages to This Slave

This value represents the total number of messages that have matched this slaves address on this port, whether the slave actually determined them to be good (worthy of response) or not.

5.5.4 Total Message Responses From This Slave

This value represents the number of good (non-error) responses that the slave has sent to the master on this port. The presumption is that if the slave is responding, the message was good.

5.5.5 Total Messages Seen By This Slave

This value represents the total number of commands seen by the slave on this port, regardless of the slave address.

Note: All counters in the Slave Error Table will rollover to 0 after reaching 65535

5.5.6 Product Name

These two words represent the product name of the module in an ASCII representation. In the case of the N2 product, the letters "N2" should be displayed when placing the programming software in the ASCII data representation mode.

5.5.7 Revision

These two words represent the product revision level of the firmware in an ASCII representation. An example of the data displayed would be ‘1.00’ when placing the programming software in the ASCII data representation mode.

5.5.8 Operating System Revision

These two words represent the module’s internal operating system revision level in an ASCII representation.

5.5.9 Production Run Number

This number represents the ‘batch’ number that your particular chip belongs to in an ASCII representation.

6 Support, Service & Warranty

In This Chapter

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

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

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

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

6.1 How to Contact Us: Technical Support

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

Asia Pacific

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

Languages spoken include: Chinese, English

Europe (location in Toulouse, France)

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

Languages spoken include: French, English

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

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

Languages spoken include: English, Spanish

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

Brasil (location in Sao Paulo)

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

Languages spoken include: Portuguese, English

6.2 Return Material Authorization (RMA) Policies and Conditions

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

6.2.1 All Product Returns:

a) In order to return a Product for repair, exchange or otherwise, the Customer must obtain a Returned Material Authorization (RMA) number from ProSoft and comply with ProSoft shipping instructions.
b) In the event that the Customer experiences a problem with the Product for any reason, Customer should contact ProSoft Technical Support at one of the telephone numbers listed above (page 73). A Technical Support Engineer will request that you perform several tests in an attempt to isolate the problem. If after completing these tests, the Product is found to be the source of the problem, we will issue an RMA.
c) All returned Products must be shipped freight prepaid, in the original shipping container or equivalent, to the location specified by ProSoft, and be accompanied by proof of purchase and receipt date. The RMA number is to be prominently marked on the outside of the shipping box. Customer agrees to insure the Product or assume the risk of loss or damage in transit. Products shipped to ProSoft using a shipment method other than that specified by ProSoft or shipped without an RMA number will be returned to the Customer, freight collect. Contact ProSoft Technical Support for further information.
d) A 10% restocking fee applies to all warranty credit returns whereby a Customer has an application change, ordered too many, does not need, etc.

6.2.2 Procedures for Return of Units Under Warranty:

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

a) A replacement module will be shipped and invoiced. A purchase order will be required.
b) Credit for a product under warranty will be issued upon receipt of authorized product by ProSoft at designated location referenced on the Return Material Authorization.

6.2.3 Procedures for Return of Units Out of Warranty:

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

The following is a list of non-repairable units:

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

6.2.4 Purchasing Warranty Extension:

a) ProSoft's standard warranty period is three (3) years from the date of shipment as detailed in "Limited Warranty (page 76)". The Warranty Period may be extended at the time of equipment purchase for an additional charge, as follows:

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

6.3 LIMITED WARRANTY

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

6.3.1 What Is Covered By This Warranty

a) *Warranty On New Products*: ProSoft warrants, to the original purchaser, that the Product that is the subject of the sale will (1) conform to and perform in accordance with published specifications prepared, approved and issued by ProSoft, and (2) will be free from defects in material or workmanship; provided these warranties only cover Product that is sold as new. This Warranty expires three years from the date of shipment (the "Warranty Period"). If the Customer discovers within the Warranty Period a failure of the Product to conform to specifications, or a defect in material or workmanship of the Product, the Customer must promptly notify ProSoft by fax, email or telephone. In no event may that notification be received by ProSoft later than 39 months. Within a reasonable time after notification, ProSoft will correct any failure of the Product to conform to specifications or any defect in material or workmanship of the Product, with either new or used replacement parts. Such repair, including both parts and labor, will be performed at ProSoft's expense. All warranty service will be performed at service centers designated by ProSoft.

b) *Warranty On Services*: Materials and labor performed by ProSoft to repair a verified malfunction or defect are warranteed in the terms specified above for new Product, provided said warranty will be for the period remaining on the original new equipment warranty or, if the original warranty is no longer in effect, for a period of 90 days from the date of repair.

6.3.2 What Is Not Covered By This Warranty

a) ProSoft makes no representation or warranty, expressed or implied, that the operation of software purchased from ProSoft will be uninterrupted or error free or that the functions contained in the software will meet or satisfy the purchaser's intended use or requirements; the Customer assumes complete responsibility for decisions made or actions taken based on information obtained using ProSoft software.

b) This Warranty does not cover the failure of the Product to perform specified functions, or any other non-conformance, defects, losses or damages caused by or attributable to any of the following: (i) shipping; (ii) improper installation or other failure of Customer to adhere to ProSoft's specifications or instructions; (iii) unauthorized repair or maintenance; (iv) attachments, equipment, options, parts, software, or user-created programming (including, but not limited to, programs developed with any IEC 61131-3, "C" or any variant of "C" programming languages) not furnished by ProSoft; (v) use of the Product for purposes other than those for which it was designed; (vi) any other abuse, misapplication, neglect or misuse by the Customer; (vii) accident, improper testing or causes external to the Product such as, but not limited to, exposure to extremes of temperature or humidity, power failure or power surges; or (viii) disasters such as fire, flood, earthquake, wind and lightning.

c) The information in this Agreement is subject to change without notice. ProSoft shall not be liable for technical or editorial errors or omissions made herein; nor for incidental or consequential damages resulting from the furnishing, performance or use of this material. The user guide included with your original product purchase from ProSoft contains information protected by copyright. No part of the guide may be duplicated or reproduced in any form without prior written consent from ProSoft.

6.3.3 Disclaimer Regarding High Risk Activities

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

6.3.4 Intellectual Property Indemnity

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

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

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

6.3.5 Disclaimer of all Other Warranties

The Warranty set forth in What Is Covered By This Warranty (page 76) are in lieu of all other warranties, express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

*6.3.6 Limitation of Remedies ***

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

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

6.3.7 Time Limit for Bringing Suit

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

6.3.8 No Other Warranties

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

6.3.9 Allocation of Risks

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

6.3.10 Controlling Law and Severability

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

Index