ProSoft TECHNOLOGY

Where Automation Connects.

inRAx®

MVI69-DFCM

CompactLogix or MicroLogix Platform

DF1 Interface Module

11/3/2008

USER MANUAL

Please Read This Notice

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

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

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

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

Battery Life Advisory

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

Note: The battery is not user replaceable.

Your Feedback Please

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

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

MVI69-DFCM User Manual
11/3/2008

ProSoft® Product Documentation

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

Contents

Guide to the MVI69-DFCM 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 60) Diagnostics and Troubleshooting (page 45)	This section describes how to verify communications with the network. Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview Glossary	→	Reference (page 63) Functional Overview (page 65) Product Specifications (page 63)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 107)	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

Installing the MVI69-DFCM module requires a reasonable working knowledge of the Rockwell Automation hardware, the MVI69-DFCM 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-DFCM 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-DFCM module. The module requires 800mA of available power.

Important: The MVI69-DFCM 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-DFCM 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-DFCM 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-DFCM 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-DFCM Module	MVI69-DFCM	DF1 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-DFCM module.

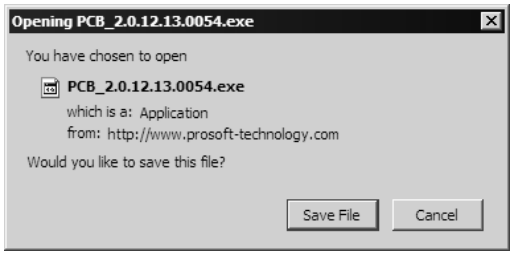
If any of these components are missing, please contact ProSoft Technology Support for replacement parts.

1.3 Install ProSoft Configuration Builder Software

You must install the ProSoft Configuration Builder (PCB) software in order to configure the MVI69-DFCM 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.



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-DFCM module.

To install ProSoft Configuration Builder from the CD-ROM

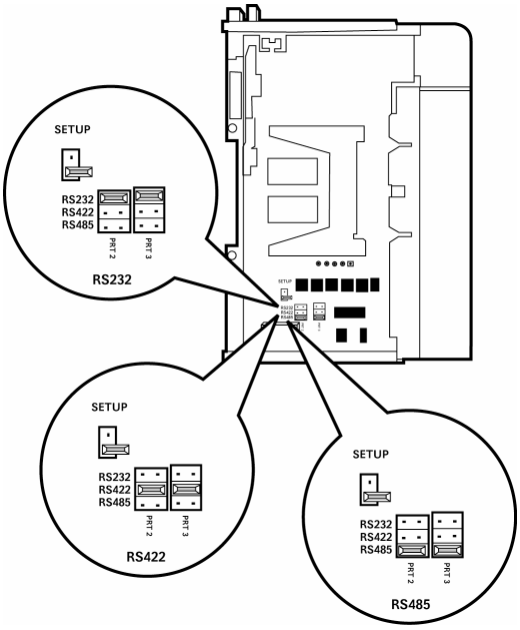
1. Insert the ProSoft Solutions CD-ROM into the CD drive of your PC. Wait for the startup screen to appear.
2. On the startup screen, click *Product Documentation*. This action opens an explorer window.

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.

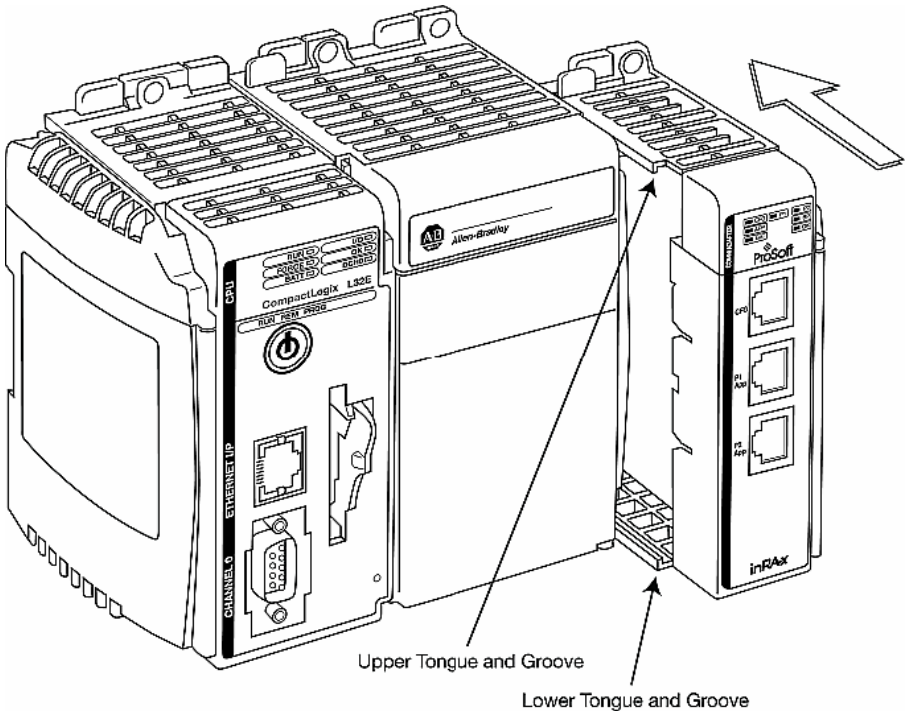
1.5 Install the Module in the Rack

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.

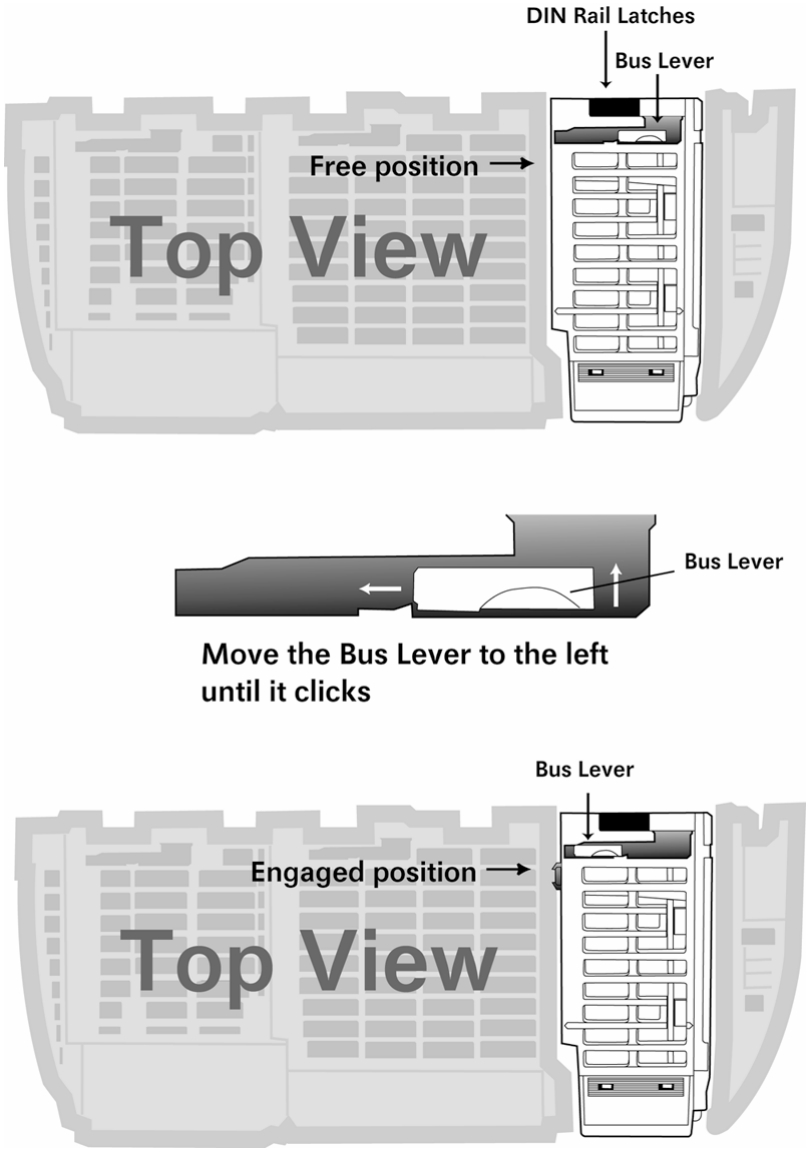
Warning: This module is not hot-swappable! Always remove power from the rack before inserting or removing this module, or damage may result to the module, the processor, or other connected devices.

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.



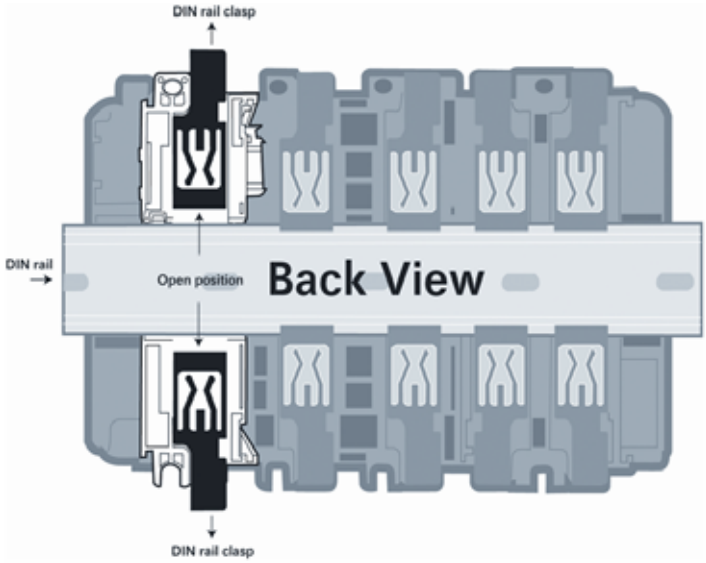
2 Move the module back along the tongue-and-groove slots until the bus connectors on the MVI69 module and the adjacent module line up with each other.

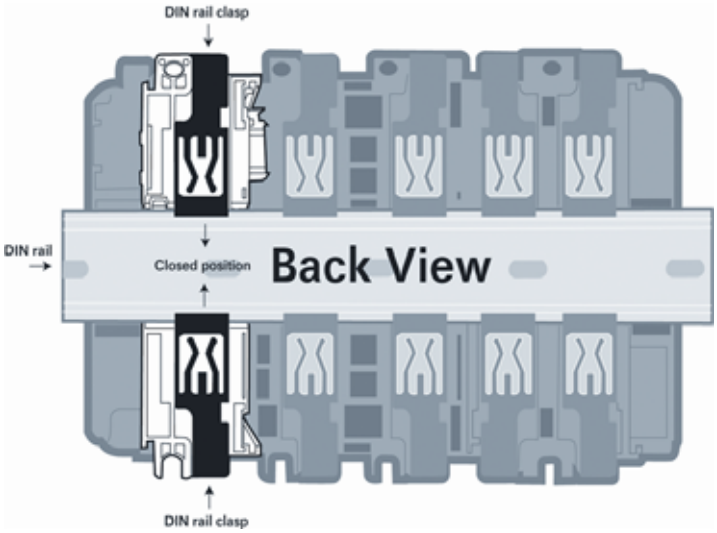
3 Push the module's bus lever back slightly to clear the positioning tab and move it firmly to the left until it clicks. Ensure that it is locked firmly in place.



4 Close all DIN rail latches.

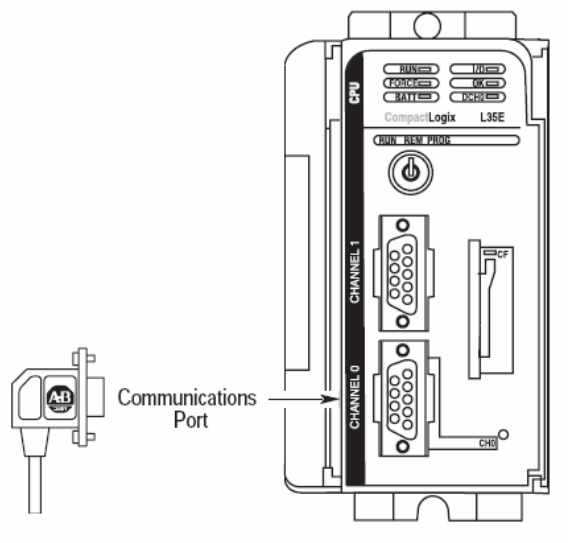
5 Press the DIN rail mounting area of the controller against the DIN rail. The latches will momentarily open and lock into place.



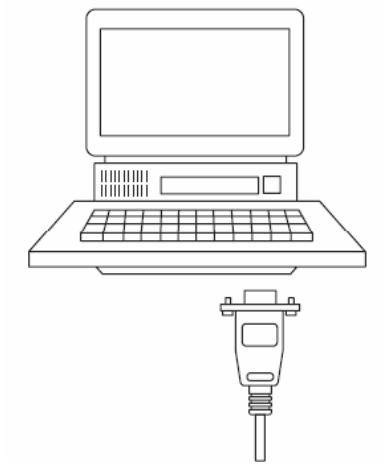


1.6 Connect your PC to the Processor

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



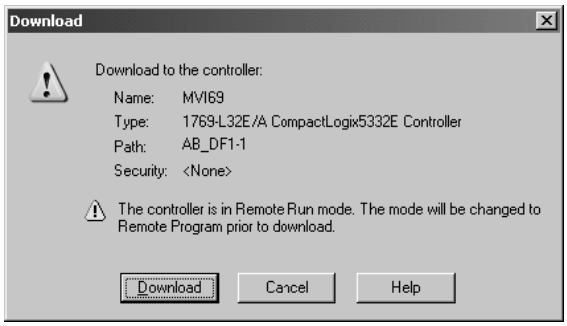
2 Connect the straight connector end of the cable to the serial port on your computer.



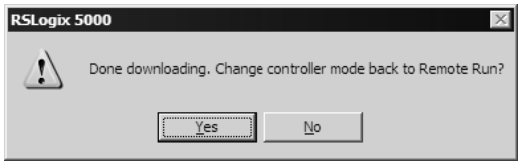
1.7 Download the Sample Program to the Processor

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.



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

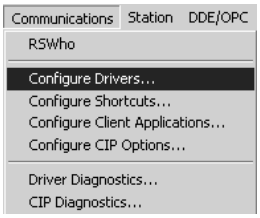


Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

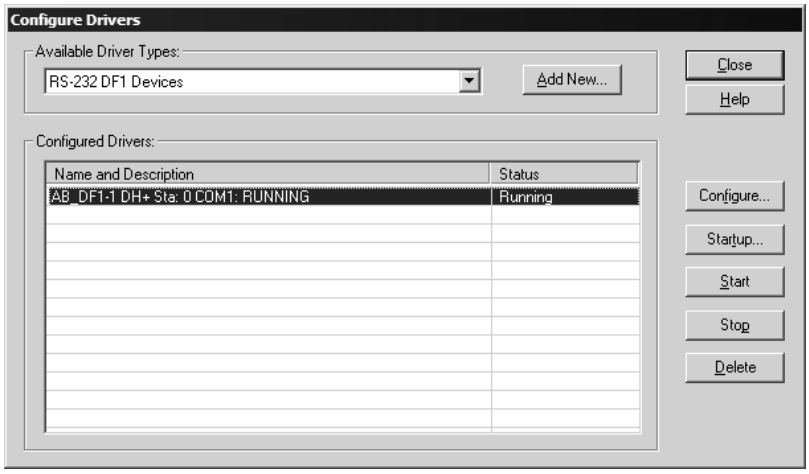
1.7.1 Configuring RSLinx

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.

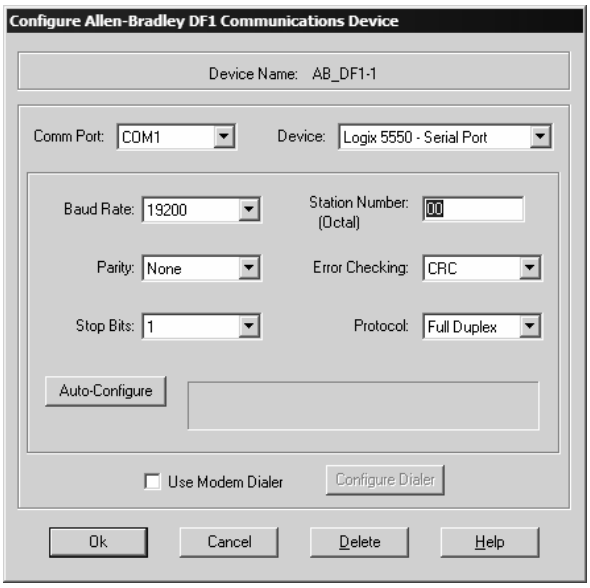


This action opens the Configure Drivers dialog box.



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

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



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

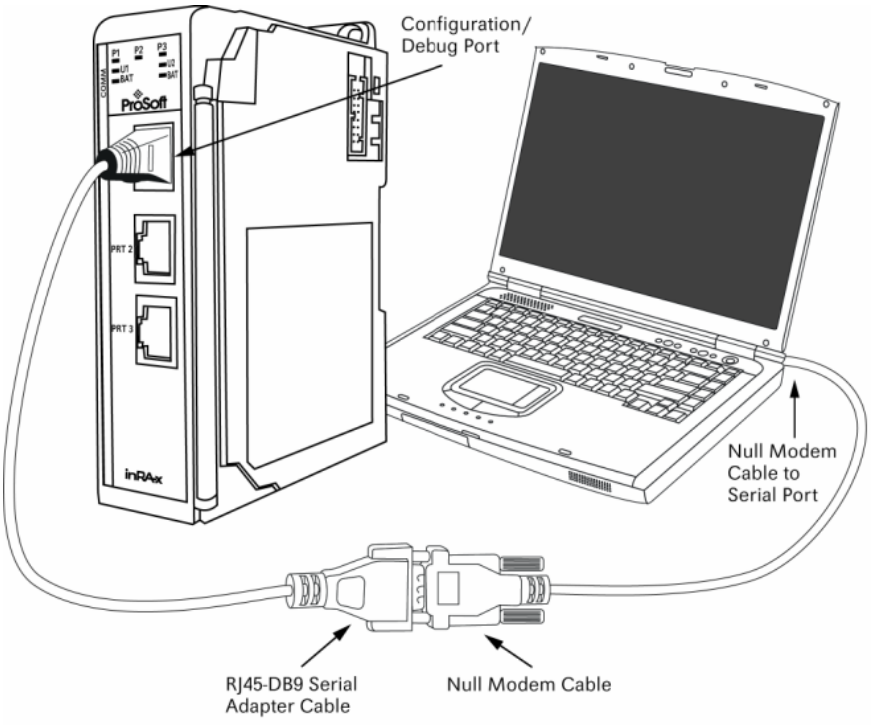
5 When you see the message "Auto Configuration Successful", click the OK button to dismiss the dialog box.

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

1.8 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-DFCM Module

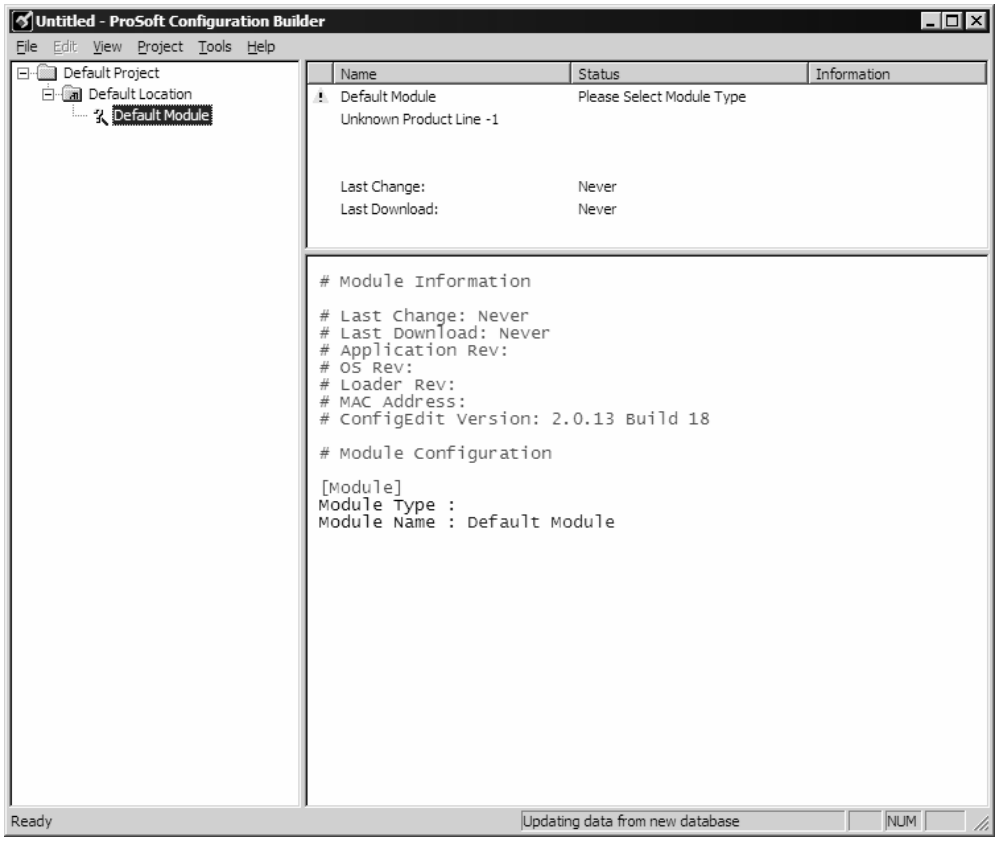
In This Chapter

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

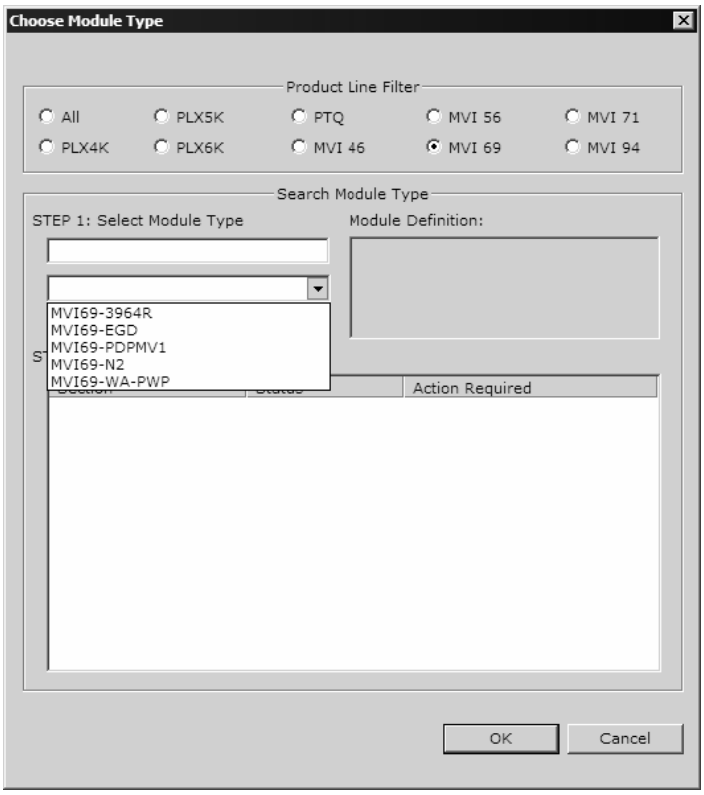
To begin, start ProSoft Configuration Builder. If you have used other Windows configuration tools before, you will find the screen layout familiar. ProSoft Configuration Builder's window consists of a tree view on the left, an information pane and a configuration pane on the right side of the window. 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.



Your first task is to add the MVI69-DFCM 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.



3 In the Product Line Filter area of the dialog box, select MVI69. In the Select Module Type dropdown list, select MVI69-DFCM, 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:

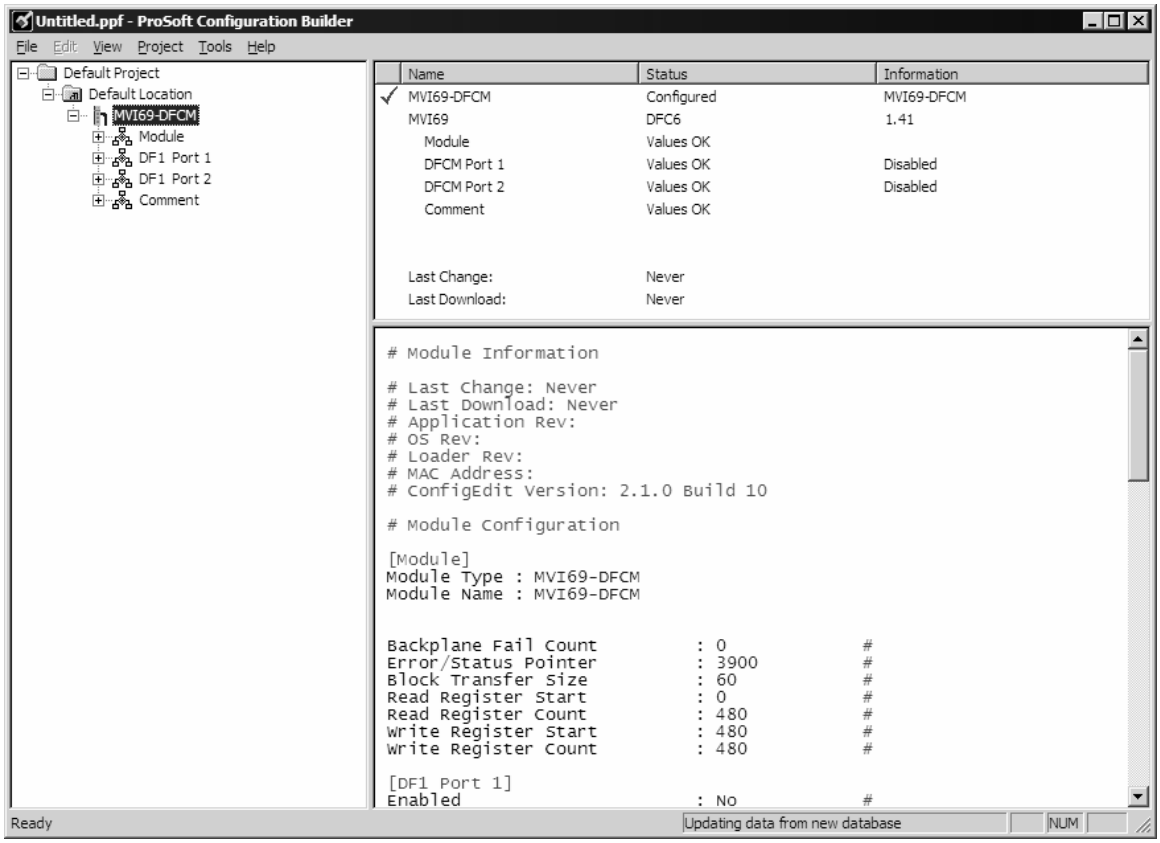
1 Right-click the Location folder and choose Add Module. A new module icon appears.

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-DFCM module to the project.



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

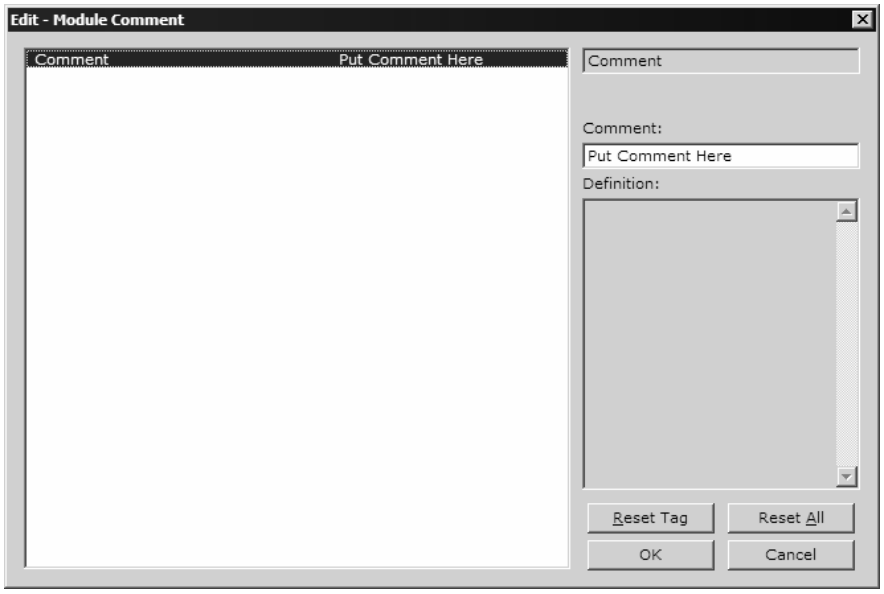
To configure module parameters

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.



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 [Module]

This section provides the module with a unique name, identifies the method of failure for the communications for the module if the CompactLogix or MicroLogix is not in run, and describes how to initialize the module upon startup.

The following example shows a sample [Module] section:

```
[Module]
Module Name: Test Example of DFCM Communication Module
```

Modify each of the parameters based on the needs of your application.

2.2.1 Module Name

0 to 80 characters

This parameter assigns a name to the module that can be viewed using the configuration/debug port. Use this parameter to identify the module and the configuration file.

2.2.2 Backplane Fail Count

0 to 65535

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

2.2.3 Error / Status Pointer

-1 to 3999

Starting register location in virtual database for the error/status table. If a value of -1 is entered, the error/status data will not be placed in the database. All other valid values determine the starting location of the data. This data area includes the module version information.

2.2.4 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.

2.2.5 Read Register Start

0 to 4999

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 4999.

2.2.6 Read Register Count

0 to 5000

This parameter specifies the number of registers to be transferred from the module to the processor. Valid entry for this parameter is 0 to 5000.

2.2.7 Write Register Start

0 to 4999

This parameter specifies the starting register in the module where the data will be transferred from the processor to the module. Valid range for this parameter is 0 to 4999.

2.2.8 Read Register Count

0 to 5000

This parameter specifies the number of registers to be transferred from the module to the processor. Valid entry for this parameter is 0 to 5000.

2.3 [DF1 Port x]

The following shows a sample DF1 Port x section of the configuration file.

```
[DF1 Port x]
Enabled               : Yes    # Y=Use port, N=Do not use port
RS Interface          : 0      # 0=RS-232, 1=RS-485, 2=RS-422
Type                  : Master # M=Master, S=Slave
Local Station ID      : 0      # DF1 node address
Protocol              : F      # F=Full-Duplex, H=Half-Duplex
Termination Type      : CRC    # B=BCC, C=CRC
Baud Rate             : 19200  # Baud rate for Port 110-38400
Parity                : None   # N=None,O=Odd,E=Even,M=Mark,S=Space
Data Bits             : 8      # 5, 6, 7 or 8
Stop Bits             : 1      # 1 or 2
Min Response Delay    : 0      # 0-65535 mSec before sending response msg
RTS On                : 0      # 0-65536 mSec before message
RTS Off               : 1      # 0-65536 mSec after message
Use CTS Line          : No     # Use CTS modem control line (Y/N)
Response Timeout      : 1000   # Response message timeout (0-65535 mSec)
Retry Count           : 3      # Response failure retry count
ENQ Delay             : 10     # 0-65535 mSec before DLE-ENQ sent
Minimum Command Delay : 10     # Minimum number of msec's between commands
Error Delay Counter   : 100    # 0-65535 Command cycle count if error
Command Control Reg   : -1     # Cmd control start DB Reg (-1=ignore)
First File            : 7      # First file number for SLC simulation
File Size             : 200    # Number of elements in each file
File Offset           : 0      # Database offset for first file element
```

Modify each of the parameters as follows based on the needs of your application:

2.3.1 Enabled

This parameter specifies if the port will be used. If the parameter is set to No, the port is disabled. If the parameter is set to Yes, the port is enabled.

2.3.2 Type

```
Type                    : Master #M=Master, S=Slave
```

This parameter defines if the port will emulate a master or a slave device. Enter Master if the port is to emulate a master device or Slave if the port is to emulate a slave device.

2.3.3 Local Station ID

This parameter specifies the local station ID for all DF1 messages sent from this master port. A value of 255 is not permitted as this is the broadcast address. Enter a value in the range of 0 to 254.

2.3.4 Protocol

```
Protocol                : Full   #F=Full-Duplex, H=Half-Duplex
```

This parameter specifies the DF1 protocol to be used on the port. Enter Full for full-duplex communications or Half for half-duplex communications.

2.3.5 Termination Type

```
Termination Type        : CRC    #B=BCC, C=CRC
```

This parameter specifies error checking for all DF1 messages. Enter CRC or BCC.

2.3.6 Baud Rate

This is the baud rate to be used on the port. Enter the baud rate as a value. For example, to select 19K baud, enter 19200.

2.3.7 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.

2.3.8 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.

2.3.9 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.

2.3.10 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.

2.3.11 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.

2.3.12 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.

2.3.13 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).

2.3.14 Response Timeout

Number of milliseconds to wait for response to command. The value is set depending upon the communication network used and the expected response time of the slowest device on the network. Valid values are 0 to 5000 milliseconds.

2.3.15 Retry Count

0 to 10

This parameter specifies the number of times a command will be retried if it fails.

2.3.16 ENQ Delay

```
ENQ Delay              : 0      #0-65535 milliseconds before DLE-ENQ sent
```

This parameter specifies the number of milliseconds to wait after a DLE-ACK is received from a slave using half-duplex mode before the DLE-ENQ request is made for data. Enter a value in the range of 0 to 65535 milliseconds.

2.3.17 Minimum Command Delay

This parameter specifies the number of milliseconds to wait between issuing each command. This delay value is not applied to retries.

2.3.18 Error Delay Count

```
Error Delay Count      : 100    #0-65535 Command cycle count if error
```

This parameter specifies the number of polls to be skipped on the slave before trying to re-establish communications. After the slave fails to respond, the master will skip commands to be sent to the slave the number of times entered in this parameter. Enter a value in the range of 0 to 65535.

2.3.19 Command Error Pointer

```
Command Error Pointer  : 3000   #Cmd Error list data (-1=ignore)
```

This parameter sets the address in the internal database where the command error data will be placed. If the value is set to -1, the data will not be transferred to the database. Enter a value from 0 to 4999.

2.3.20 Slave List Pointer

```
Slave List Pointer     : 3100   #Slave status list data (-1=ignore)
```

This parameter specifies the starting address in the virtual database where the 256 slave status values will be written. If the parameter is set to -1, the slave data will not be placed in the database. Enter a value in the range of -1 to 4743.

2.3.21 First File

```
First File             : 7      #First file number for SLC simulation
```

This parameter is used when a request for a file is received on the communication port. This field is required when responding to PLC5 and SLC DF1 commands. Use this parameter to define the virtual file(s) to be simulated on the module. Enter a value in the range of 0 to 100.

2.3.22 File Size

Range 1 to 1000

This parameter specifies the size of each file to be simulated on the module. All files simulated are defined to have the same assigned size.

2.3.23 File Offset

This parameter sets the database register location of the first element in the first file simulated in the module. All offsets in the first file and subsequent files will be computed using the address specified. Enter a value in the range of 0 to 4999.

2.4 [DFCM Port x Commands]

This section defines the commands to be issued from the module to server devices on the network. These commands can be used for data collection and/or control of devices on the network.

```
[DF1 Port 1 Commands]
# The file contains examples for a SLC 5/03 processor.
#
# LOCATION     :
# DATE         : 06/24/99
# CONFIGURED BY: RAR
# MODIFIED     :
#   07/23/99 -- Set to read more data file types.
#
START
#      1       2       3      4    5        6    7    8    9   10   11
#        Internal     Poll       Swap     Node Func File File  Elm  Sub
# Enable  Address Interval Count Code  Address Code Type    #    #  Elm
       1       10        0    10    0       10  501    N    7    0
       1     1500        0    10    0       10  502    N    7    0    0
       1       10        0    10    0       10  509    N    7   10
END
```

2.4.1 Command List Overview

In order to interface the module with other nodes, you must construct a command list of up to 100 user-defined commands. The commands in the list specify the device to be addressed, the function to be performed (read or write), the data area in the device to interface with, and the registers in the internal database to be associated with the device data. The command list is processed from top (command #0) to bottom. A poll interval parameter is associated with each command to specify a minimum delay time in tenths of a second between the issuance of a command. If the user specifies a value of 10 for the parameter, the command is executed no more frequently than every (1) second.

Write commands have a special feature, as they can be set to execute only if the data in the write command changes. If the register data values in the command have not changed since the command was last issued, the command will not be executed. If the data in the command has changed since the command was last issued, the command is executed. Use of this feature can lighten the load on the network. In order to implement this feature; set the enable code for the command to a value of 2.

The module supports numerous commands. This permits the module to interface with a wide variety of devices.

The commands take the following parameters:

Enable: This field defines whether or not the command is to be executed and under what conditions. A **0** indicates that the command is disabled and will not be executed in the normal polling sequence. A **1** indicates that the command is executed each scan of the command list if the Poll Interval Time is set to zero. If the Poll Interval Time is set, the command is executed when the internal timer expires. A **2** indicates that the command only executes if the internal data associated with the command changes. This value is valid for write commands only.

Internal Address: This field specifies the database address in the module's internal database to be associated with the command. If the command is not a read function, the data received in the response message is placed at the specified location. If the command is a write function, data used in the command is sourced from the specified data area. Valid values are 0 to 3999.

Poll Interval: This parameter specifies the minimum interval to execute continuous commands (Enable code of 1). The parameter is entered in $1/10^{th}$ of a second. Therefore, if a value of .50 is entered for a command, the command executes no more frequently that every 5 seconds. Valid values are 0 to 65535.

Count: This parameter specifies the number of registers or digital points to be associated with the command.

Swap Code: This parameter defines if the data received from the server is to be ordered differently than that received from the server device. This parameter is helpful when dealing with floating point or other multi-register values, as there is no standard method of storage of these data types in server devices. This parameter can be set to order the register data received in an order useful by other applications. The following defines the values and operations:

Swap Code	Description
0	None - No Change is made in the byte ordering (1234 = 1234)
1	Words - The words are swapped (1234=3412)
2	Words & Bytes - The words are swapped then the bytes in each word are swapped (1234=4321)
3	Bytes - The bytes in each word are swapped (1234=2143)

The words should be swapped only when using an even number of words.

Node Address: The address of the device being addressed by the command.

Function Code: These parameters specify the function to be executed by the command. Refer to DF1 Command Set For ProSoft Technology Communication Modules (page 93) for more information on constructing commands for the module.

2.5 [DF1 Port x Override Data File Maps]

```
[DF1 Port 1 OVERRIDE DATA FILE MAPS]
#      DB     File     First     Word
# Address   Number   Element   Length
START
END
```

Group	File	Register	Content	Description
Port 1 Override File Maps		8000 to 8003	File Map #1	This set of registers contains the first override file map for the slave port.
		8004 to 8007	File Map #2	This set of registers contains the second override file map for the slave port.
		-		
		8196 to 8199	Command # 50	This set of registers contains the last override file map for the slave port.

2.5.1 P1 and P2 Override File Mappings

This feature allows a write command to be re-directed to the module's ReadData area. If the DF1 master has a fixed address to write to, the module's WriteData area with the delivered data would be overwritten on the next scan. This feature requires that the port be configured as a slave.

Note: A Slave ID setting of 255 will respond to all commands sent on the network. If this setting is used in a multipoint network, only writes should be used on the network from the DF1 master.

2.6 Changing parameters during operation

A copy of the module's configuration data is mapped in the module's database as displayed in the following table. These values are initialized when the module first receives its configuration from the configuration file.

Module Register Address	Functional Modes Affected	Name	Description
5000 to 5009	Data Transfer	General Module Configuration	This section of the configuration data contains the module configuration data that defines the data transfer between the module and the CompactLogix or MicroLogix processor.
5010 to 5039 and 5040 to 5069	Master and Slave	Port Configuration	These sections define the characteristics of each of the DFCM serial communication ports on the module. These parameters must be set correctly for proper module operation.
5200 to 6399 and 6400 to 7599	Master	Master Command List	If the module's Master Mode functionality is to be supported on a port, the Master Command List must be set up.

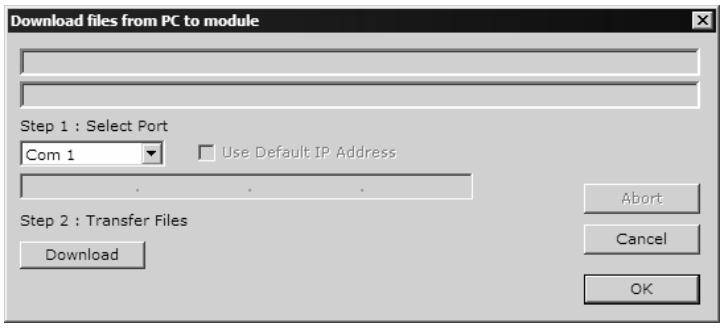
The configuration file is located in the module as well as on the CD and web site. With a new module, ProSoft recommends that you download the configuration file from the module for editing.

2.7 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.

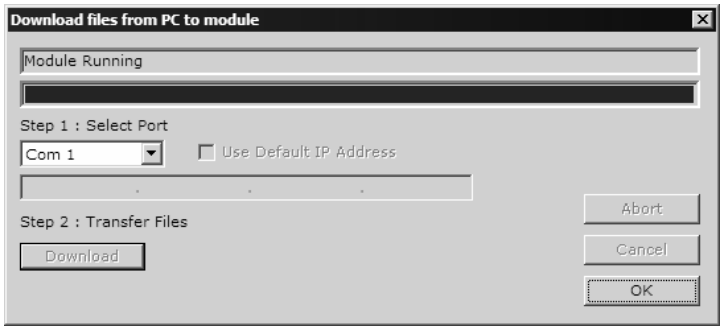
To Download the Project File

1. In the tree view in ProSoft Configuration Builder, click once to select the MVI69-DFCM 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.



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*".



3 Ladder Logic

In This Chapter

Ladder logic is required for application of the MVI69-DFCM 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 (DFCMModuleDef)

All data related to the MVI69-DFCM 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 table.

Name	Data Type	Description
BlockTransferSize	INT	60, 120 or 240
ReadData	INT[720]	Data read from MVI69-DFCM
WriteData	INT[720]	Data written to MVI69-DFCM
BP	DFCMBackplane	Variables for BP logic
ModuleStatus	DFCM_STATUS	Module Status
BlockRequest	DFCMBlockRequest	Bits to request special blocks
ReadClock	DFCMClock	Time read from MVI69-DFCM
WriteClock	DFCMClock	Time written to MVI69-DFCM
CommandControl	DFCMCommandControlPorts	Used for Command Control operation
EventCommand	DFCMEventCommandPorts	Used for Event Command operation
SlavePollingControl	DFCMSlavePollingControlPorts	Slave polling control
SlaveStatus	DFCMSlaveStatusPorts	Status codes for each slave

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 Status Object (DFCM_STATUS)

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

Name	Data Type	Description
Pass_Cnt	INT	Program cycle counter
Prod	SINT[4]	Product Name
Rev	SINT[4]	Revision Level Number
Op	SINT[4]	Operating Level Number
Run	SINT[4]	Run Number
PortErr	DFCM_PORT_ERROR[2]	Port error statistics
BlkStats	DFCM_BLK_STATS	
Port1_CurErr	INT	Port 1 current error
Port1_LastErr	INT	Port 1 current error
Port2_CurErr	INT	Port 2 current error
Port2_LastErr	INT	Port 2 current error

Refer to the Reference chapter for a complete listing of the data stored in this object.

3.1.2 User Data Objects

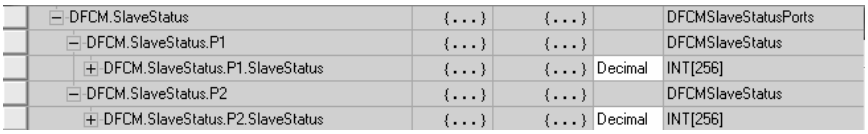
These objects hold data to be transferred between the processor and the MVI69-DFCM module. The user data is the read and write data transferred between the processor and the module as "pages" of data up to 60, 120, or 240 words long depending on the Block Transfer Size parameter.

The read data array should be dimensioned according to the Read Data Count parameter in the configuration file. The ReadData task is responsible for placing the data received into the proper position in the read data array. Use this data for status and control in the ladder logic of the processor.

The write data array should be dimensioned according to the Read Data Count parameter in the configuration file. . The WriteData task is responsible for placing the write data into the output image for transfer to the module.

3.1.3 Slave Polling Control and Status

Two arrays are allocated in the processor to hold the polling status of each slave on the master ports. This status data can be used to determine which slaves are currently active on the port, are in communication error or have their polling suspended and disabled. Ladder logic in the processor can be written to monitor and control the status of each slave on a master port. The objects used are displayed below:



3.1.4 DFCM Slave Polling Control (DFCMSlavePollingControl)

This user-defined data type sends the Enable/Disable blocks to control the slave polling with blocks 3000, 3001, 3100 or 3101.

Name	Data Type	Description
NumberOfSlaves	INT	
SlaveID	INT[60]	

3.1.5 Event Command (DFCMEventCommand)

This data type issues event commands to the modules using blocks 1000 and 2000.

Name	Data Type	Description
InternalAddress	INT	
Count	INT	
Swap	INT	
NodeAddress	INT	
FunctionCode	INT	
Parameter	INT[4]	

3.1.6 Command Control (DFCMCommandControl)

This data type issues command control commands to the module using blocks 5000 to 5006 or 5100 to 5106.

Name	Data Type	Description
CommandIndex	INT[6]	
CommandsAdded	INT	

3.1.7 Clock (DFCMClock)

Reading or writing date and time is performed by filling in the date and time and passing between the processor and module. See data type below.

Name	Data Type	Description
Year	INT	
Month	INT	
Day	INT	

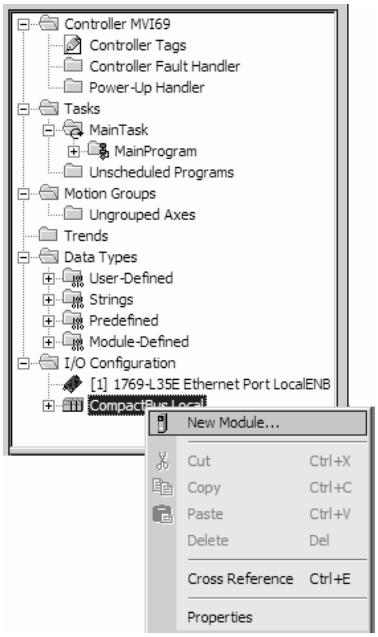
Name	Data Type	Description
Hour	INT	
Minute	INT	
Second	INT	
Reserved1	INT	
Reserved2	INT	
Reserved3	INT	

3.2 Adding the Module to an Existing CompactLogix Project

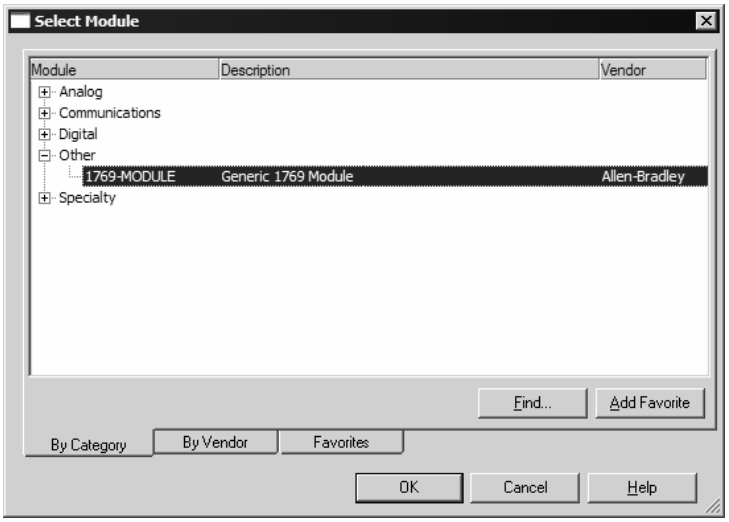
Important: The MVI69-DFCM module has a power supply distance rating of 2 (L43 and L45 installations on first 2 slots of 1769 bus)

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.

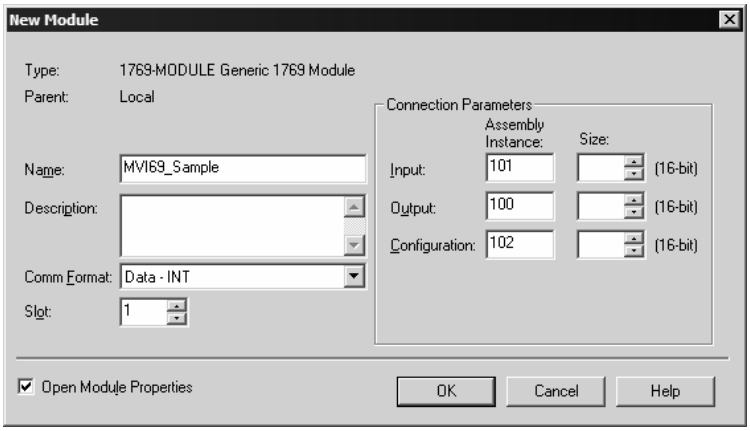
1 **Add the** MVI69-DFCM **module to the project.** Right-click the mouse button on the I/O Configuration option in the Controller Organization window to display a pop-up menu. Select the New Module option from the I/O Configuration menu.



This action opens the following dialog box:



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



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

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.

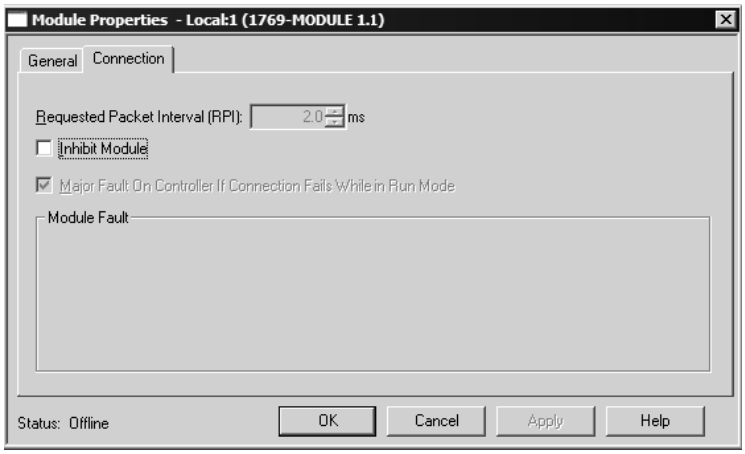
Block Transfer Size = 60	
Field	**Recommended Value**
Type	1769-MODULE Generic 1769 Module
Parent	Local
Name	MVI69
Description	MVI69 Application Module
Comm Format	Data - INT

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

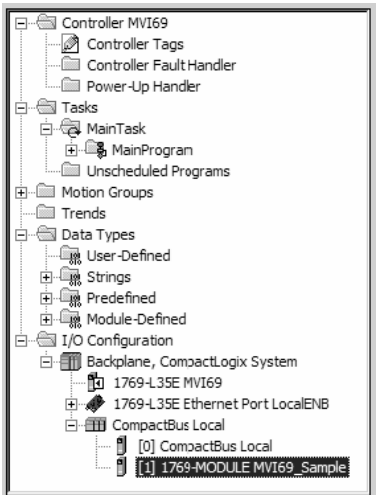
Block Transfer Size = 240	
Field	**Recommended Value**
Type	1769-MODULE Generic 1769 Module
Parent	Local
Name	MVI69
Description	MVI69 Application Module
Comm Format	Data - INT
Slot	The slot number in the rack where the module is installed
Input Assembly Instance	101
Input Size	242
Output Assembly Instance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

5 Click **Next** to continue.



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:



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.

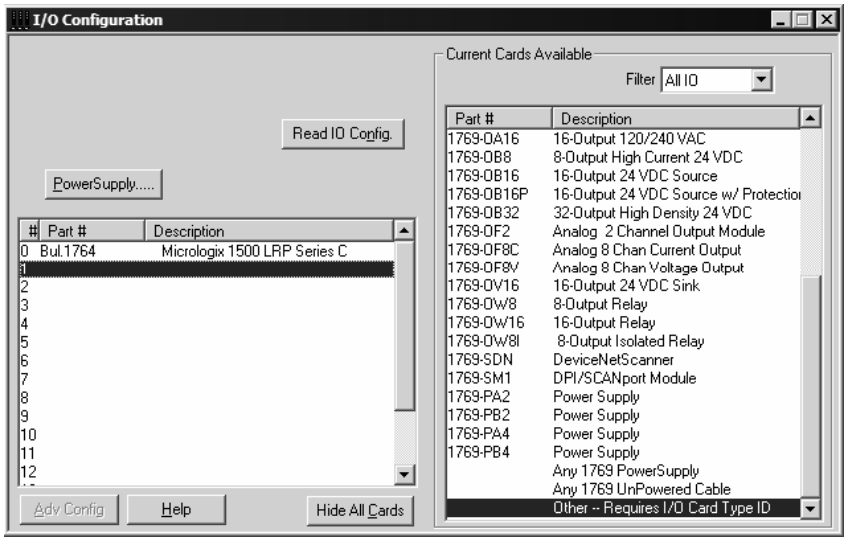
11 Save and Download the new application to the controller and place the processor in run mode.

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.

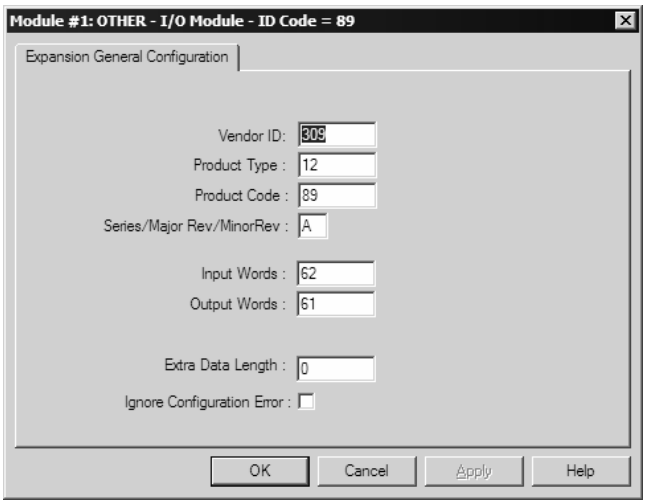
The first step in setting up the processor ladder file is to define the I/O type module to the system. Start RSLogix 500, and follow these steps:

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.



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

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



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

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

5 Click **Next** to continue.

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

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

Download the new application to the controller and place the processor in run mode. If you encounter errors, refer to **Diagnostics and Troubleshooting** (page 45) 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-DFCM module returns a 29-word Status Data block that can be used to determine the module's operating status. This data is located in the module's database at registers 7600 to 7628 and at the location specified in the configuration. This data is transferred to the CompactLogix or MicroLogix processor continuously with each read block.

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

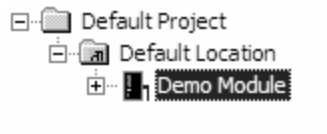
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

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.

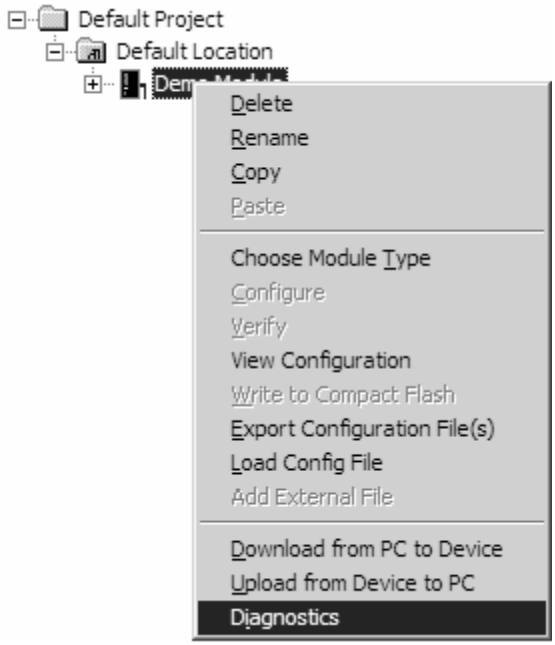
Using the Diagnostic Window in ProSoft Configuration Builder

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

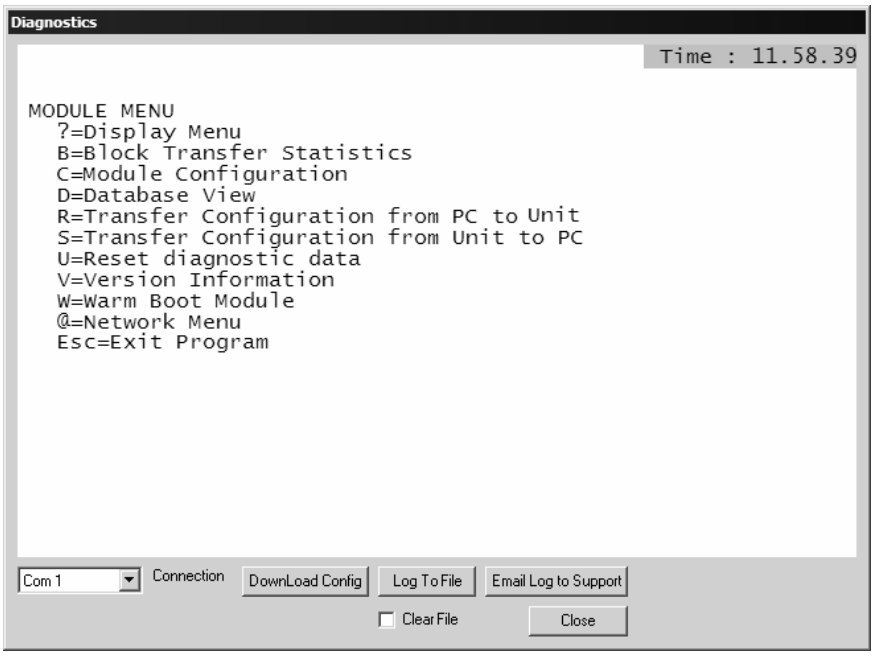
1 Start PCB program with the application file to be tested. Right click over the module icon.



2 On the shortcut menu, choose Diagnostics.



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



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:

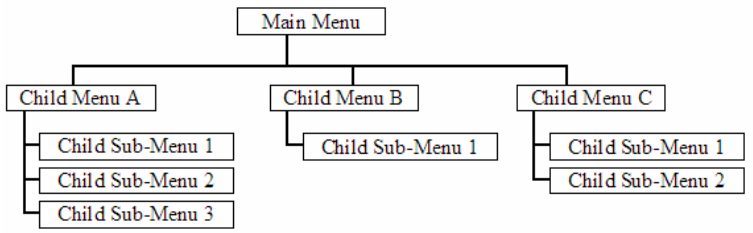
1 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.
2 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

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:

```
DF1 MASTER/SLAVE COMMUNICATION MODULE
  ?=Display Menu
  A=Data Analyzer
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  Master Command Errors : E=Port 1   F=Port 2
  Master Command List   : I=Port 1   J=Port 2
  Slave Status List     : O=Port 1   P=Port 2
  V=Version Information
  W=Warm Boot Module
  Y=Transfer Module Cfg to Processor
  Communication Status : 1=Port 1   2=Port 2
  Port Configuration   : 6=Port 1   7=Port 2

  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.

Opening the Data Analyzer Menu

Press **[A]** to open the Data Analyzer Menu. Use this command to view all bytes of data transferred on each port. Both the transmitted and received data bytes are displayed. Refer to Data Analyzer for more information about this menu.

Important: When in analyzer mode, program execution will slow down. Only use this tool during a troubleshooting session. Before disconnecting from the Config/Debug port, please press [S] to stop the data analyzer, and then press [M] to return to the main menu. This action will allow the module to resume its normal high speed operating mode.

Viewing Block Transfer Statistics

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

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

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

Viewing Module Configuration

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

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

Opening the Database Menu

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

Opening the Command Error List Menu

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

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.

Viewing the Slave Status List (Port 1 and 2)

Press **[O]** (port 1) or **[P]** (port 2) to view the 256 slave status values associated with the ports. The slave status values are defined as follows:

0 = slave is not used,

1 = slave being actively polled,

2 = slave suspended and

3 = slave disabled.

Viewing Version Information

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

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

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

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

Warm Booting the Module

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

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

Transferring Module Configuration to the Processor

Press **[Y]** to transfer the module's configuration data to the processor. Ladder logic is required in the processor to receive and implement the updated configuration. You will be prompted to confirm the transfer.

If the operation is not successful, an error code will be returned.

Code	Description
0	Transfer successful
-1	Error transferring module configuration data (block -9000)
-2	Error transferring device definition data (blocks -9100 to -9103)
-3	Error transferring master command list data (blocks -6000 to -6007)

After successful data transfer, the module will perform a warm-boot operation to read in the new data.

Viewing Communication Status

Press **[1]** to view the communication status and statistics of the DFCM Network for the module's node address. This command is useful for troubleshooting purposes.

Viewing Port Configuration

Press **[6]** or **[7]** from the Main Menu to view configuration information for ports 1 and 2.

Use this command to display detailed configuration information for the selected port.

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.

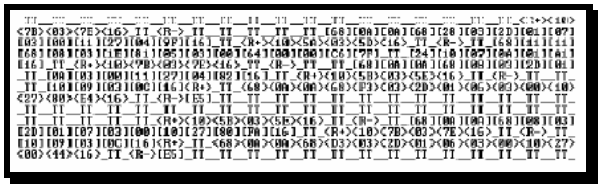
4.1.4 Data Analyzer

The data analyzer mode allows you to view all bytes of data transferred on each port. Both the transmitted and received data bytes are displayed. Use of this feature is limited without a thorough understanding of the protocol.

Note: The Port selection commands on the Data Analyzer menu differs very slightly in different modules, but the functionality is basically the same. Use the illustration above as a general guide only. Refer to the actual data analyzer menu on your module for the specific port commands to use.
Important: When in analyzer mode, program execution will slow down. Only use this tool during a troubleshooting session. Before disconnecting from the Config/Debug port, please press [S] to stop the data analyzer, and then press [M] to return to the main menu. This action will allow the module to resume its normal high speed operating mode.

Analyzing Data for the first application port

Press **[1]** to display I/O data for the first application port in the Data Analyzer. The following illustration shows an example of the Data Analyzer output.



Analyzing Data for the second application port

Press **[2]** to display I/O data for the second application port in the Data Analyzer.

Displaying Timing Marks in the Data Analyzer

You can display timing marks for a variety of intervals in the data analyzer screen. These timing marks can help you determine communication-timing characteristics.

Key	Interval
[5]	1 milliseconds ticks
[6]	5 milliseconds ticks
[7]	10 milliseconds ticks
[8]	50 milliseconds ticks
[9]	100 milliseconds ticks
[0]	Turn off timing marks

Removing Timing Marks in the Data Analyzer

Press **[0]** to turn off timing marks in the Data Analyzer screen.

Viewing Data in Hexadecimal Format

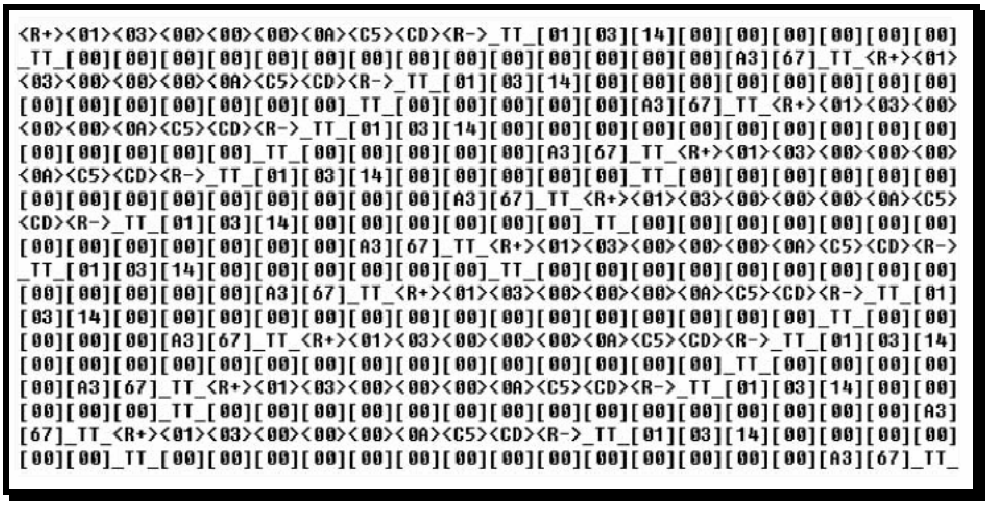
Press **[H]** to display the data on the current page in hexadecimal format.

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.

Starting the Data Analyzer

Press **[B]** to start the data analyzer. After the key is pressed, all data transmitted and received on the currently selected port will be displayed. The following illustration shows an example.



The Data Analyzer displays the following special characters:

Character	Definition
[ ]	Data enclosed in these characters represent data received on the port.
< >	Data enclosed in these characters represent data transmitted on the port.
<R+>	These characters are inserted when the RTS line is driven high on the port.
<R->	These characters are inserted when the RTS line is dropped low on the port.
<CS>	These characters are displayed when the CTS line is recognized high.
_TT_	These characters are displayed when the timing mark interval has been reached. This parameter is user defined.

Stopping the Data Analyzer

Press **[S]** to stop the data analyzer. Use this option to freeze the display so the data can be analyzed. To restart the analyzer, press **[B]**.

Important: When in analyzer mode, program execution will slow down. Only use this tool during a troubleshooting session. Before disconnecting from the Config/Debug port, please press [S] to stop the data analyzer, and then press [M] to return to the main menu. This action will allow the module to resume its normal high speed operating mode.

Returning to the Main Menu

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

4.1.5 Data Analyzer Tips

From the main menu, press **[A]** for the "Data Analyzer". You should see the following text appear on the screen:

```
Data Analyzer Mode Selected
```

After the "Data Analyzer" mode has been selected, press **[?]** to view the Data Analyzer menu. You will see the following menu:

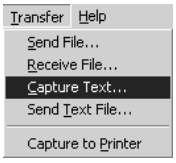
```
DATA ANALYZER VIEW MENU
 ?=Display Menu
 1=Select Port 1
 2=Select Port 2
 5=1 mSec Ticks
 6=5 mSec Ticks
 7=10 mSec Ticks
 8=50 mSec Ticks
 9=100 mSec Ticks
 0=No mSec Ticks
 H=Hex Format
 A=ASCII Format
 B=Start
 S=Stop
 M=Main Menu

 Port = 1, Format=HEX, Tick=10
```

From this menu, you can select the "Port", the "format", and the "ticks" that you can display the data in.

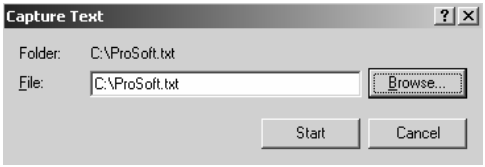
For most applications, HEX is the best format to view the data, and this does include ASCII based messages (because some characters will not display on HyperTerminal and by capturing the data in HEX, we can figure out what the corresponding ASCII characters are supposed to be).

The Tick value is a timing mark. The module will print a _TT for every xx milliseconds of no data on the line. Usually 10milliseconds is the best value to start with.

After you have selected the Port, Format, and Tick, we are now ready to start a capture of this data. The easiest way to do so is to go up to the top of you HyperTerminal window, and do a **Transfer / Capture Text** as shown below:



After selecting the above option, the following window will appear:



Next name the file, and select a directory to store the file in. In this example, we are creating a file ProSoft.txt and storing this file on our root C: drive. After you have done this, press the Start button.

Now you have everything that shows up on the HyperTerminal screen being logged to a file called ProSoft.txt. This is the file that you will then be able to email to ProSoft Technical Support to assist with issues on the communications network.

To begin the display of the communications data, you will then want to press 'B' to tell the module to start printing the communications traffic out on the debug port of the module. After you have pressed 'B', you should see something like the following:

```
[03][00][04][00][05][00][06][00][07][00][08][00][09][FB][B7]_TT__TT_<R+><01><02>
<00><00><00><0A><F8><0D><R->_TT__TT__TT_[01][02][02][00][00][B9][B8]_TT__TT_<R+>
<01><03><00><00><00><0A><C5><CD><R->_TT__TT_[01][03][14][00][00][00][01][00]_TT_
[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][CD][51]_TT__TT_<R+>
<01><01><00><00><00><A0><3C><72><R->_TT__TT_[01][01][14][00][00][01][00][02]_TT_
[00][03][00][04][00][05][00][06][00][07][00][08][00][09][00][B7][52]_TT__TT_<R+>
<01><04><00><00><00><0A><70><0D><R->_TT__TT_[01][04][14][00][00][00][01][00]_TT_
[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][FB][B7]_TT__TT_<R+>
<01><02><00><00><00><0A><F8><0D><R->_TT__TT__TT_[01][02][02][00][00][B9][B8]_TT_
_TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT__TT_[01][03][14][00][00][00][01]
[00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][CD][51]_TT_
_TT_<R+><01><01><00><00><00><A0><3C><72><R->_TT__TT__TT_[01][01][14][00][00][01]
[00][02]_TT_[00][03][00][04][00][05][00][06][00][07][00][08][00][09][00][B7][52]
_TT__TT_<R+><01><04><00><00><00><0A><70><0D><R->_TT__TT_[01][04][14][00][00][00]
[01][00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][FB][B7]
_TT__TT_<R+><01><02><00><00><00><0A><F8><0D><R->_TT__TT_[01][02][02][00][00][B9]
[B8]_TT__TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT__TT_[01][03][14][00][00]
[00][01][00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][CD]
[51]_TT__TT_<R+><01><01><00><00><00><A0><3C><72><R->_TT__TT__TT_[01][01][14][00]
[00][01][00][02]_TT_[00][03][00][04][00][05][00][06][00][07][00][08][00][09][00]
[B7][52]_TT__TT_<R+><01><04><00><00><00><0A><70><0D><R->_TT__TT_[01][04][14][00]
[00][00][01][00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09]
[FB][B7]_TT__TT_<R+><01><02><00><00><00><0A><F8><0D><R->_TT__TT__TT_[01][02][02]
[00][00][B9][B8]_TT__TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT__TT_
```

The <R+> means that the module is transitioning the communications line to a transmit state.

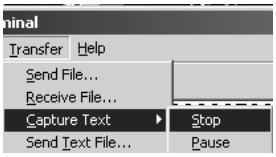
All characters shown in <> brackets are characters being sent out by the module.

The <R-> shows when the module is done transmitting data, and is now ready to receive information back.

And finally, all characters shown in the [ ] brackets is information being received from another device by the module.

After taking a minute or two of traffic capture, you will now want to stop the "Data Analyzer". To do so, press the 'S' key, and you will then see the scrolling of the data stop.

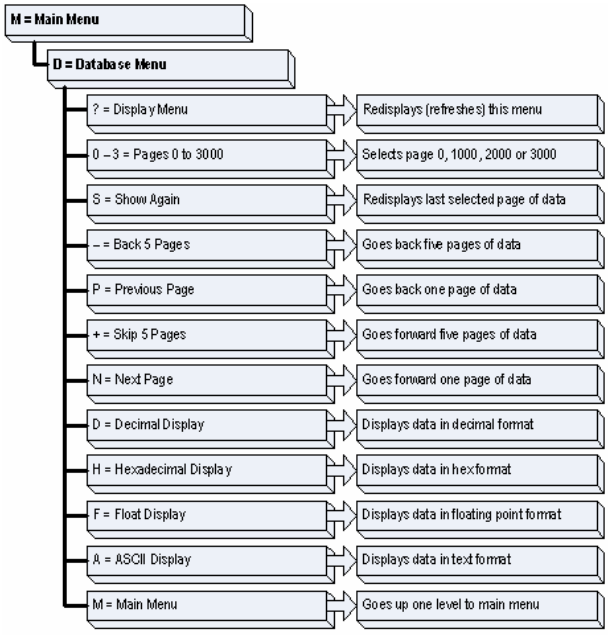
When you have captured the data you want to save, open the Transfer menu and choose Capture Text. On the secondary menu, choose Stop.



You have now captured, and saved the file to your PC. This file can now be used in analyzing the communications traffic on the line, and assist in determining communication errors.

4.1.6 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.



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

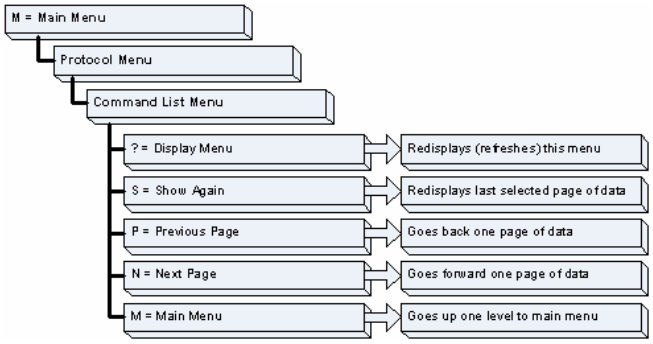
Press **[A]** to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Returning to the Main Menu

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

4.1.7 Master Command Error List Menu

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



Redisplaying the Current Page

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

Viewing the Previous 20 Commands

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

Viewing the Previous Page of Commands

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

Viewing the Next 20 Commands

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

Viewing the Next Page of Commands

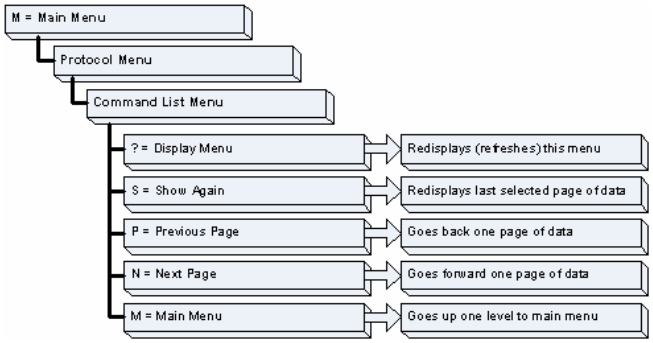
Press **[N]** to display the next page of commands.

Returning to the Main Menu

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

4.1.8 Master Command List Menu

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



Redisplaying the Current Page

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

Viewing the Previous 50 Commands

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

Viewing the Previous Page of Commands

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

Viewing the Next 50 Commands

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

Viewing the Next Page of Commands

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

Returning to the Main Menu

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

4.2 LED Status Indicators

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

Module	Color	Status	Indication
CFG	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
P1	Green	On	Data is being transferred between the module and the DFCM network on Port 1.
		Off	No data is being transferred on the port.
P2	Green	On	Data is being transferred between the module and the DFCM network on Port 2.
		Off	No data is being transferred on the port.
APP	Amber	On	The MVI69-DFCM module program has recognized a communication error on one of its ports.
		Off	The MVI69-DFCM is functioning 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.

During module configuration, the OK LED will be red and the APP and BP ACT LEDs will be on.

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

4.2.1 Clearing a Fault Condition

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

To clear the condition, follow these steps:

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

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

4.2.2 Troubleshooting

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

Processor Errors

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

Module Errors

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

5 Reference

In This Chapter

5.1 Product Specifications

The MVI69 DF1 Master/Slave Communication Module is a CompactLogix I/O compatible module that allows CompactLogix I/O compatible processors to interface easily with DF1 protocol compatible devices and hosts.

Many host SCADA applications support the DF1 protocol, while devices commonly supporting the protocol include Rockwell Automation PLCs, power monitoring equipment, as well as several other third-party devices in the marketplace.

The module has two serial ports supporting the DF1 protocol, with each port user-configurable to act as a master or as a slave. Data transfer between the module and the CompactLogix processor is asynchronous to the DF1 network, with the module's internal database being used to exchange data between the processor and the DF1 network.

The MVI69 DF1 Master/Slave Communications module allows CompactLogix processors to interface easily with DF1 protocol-compatible devices and hosts. Many host SCADA applications support the DF1 protocol, while devices commonly supporting the protocol include Rockwell Automation PLCs, power monitoring equipment, as well as several other third-party devices in the marketplace.

The module has two serial ports supporting the DF1 protocol, with each port user-configurable to act as a master or as a slave. Data transfer between the module and the CompactLogix processor is asynchronous to the DF1 network, with the module's internal database being used to exchange data between the processor and the DF1 network.

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

DF1 ports

- Full and half duplex modes supported
- CRC and BCC error checking
- Full hardware handshaking control provides radio, modem and multi-drop support
- User-definable module memory usage, supporting the storage and transfer of up to 4000 registers to/from the control processor
- 125 word read and write command lengths supported
- Floating point data movement supported

DF1 Master Protocol Specifications

The ports on the DF1 module can be individually configured as Master ports. When configured in master mode, the DFCM module is capable of reading and writing data to remote DF1 devices, enabling the CompactLogix platform to act as a SCADA sub-master.

- Command List: Up to 100 commands per Master port, each fully-configurable for function, slave address, register to/from addressing and word/byte count
- Status Data: Error codes available on an individual command basis. In addition, a slave status list is maintained per active master port
- Polling of Command List: User-configurable polling of commands, including disabled, continuous, and on change of data (write only)

DF1 Slave Protocol Specifications

The module accepts DF1 commands from an attached DF1 master unit. When in slave mode, the module can accept DF1 commands from a master to read/write data stored in the module's internal registers. This data can be derived from other DF1 slave devices on the network through a master port or from the processor and is easily transferred to the processor's data registers.

Tested Hardware Connections

Several hardware connections have been tested by ProSoft or have been customer field tested. The following physical connections have been tested successfully:

- RA Panel view (Full Duplex point-point, DFCM as slave)
- RA Processors (Full/Half duplex, DFCM as either master or slave)
- RA Power Monitors (485 Half-Duplex, DFCM as Master)

5.2 Functional Overview

This section provides an overview of how the MVI69-DFCM module transfers data using the DFCM 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-DFCM module.

Module Power Up

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

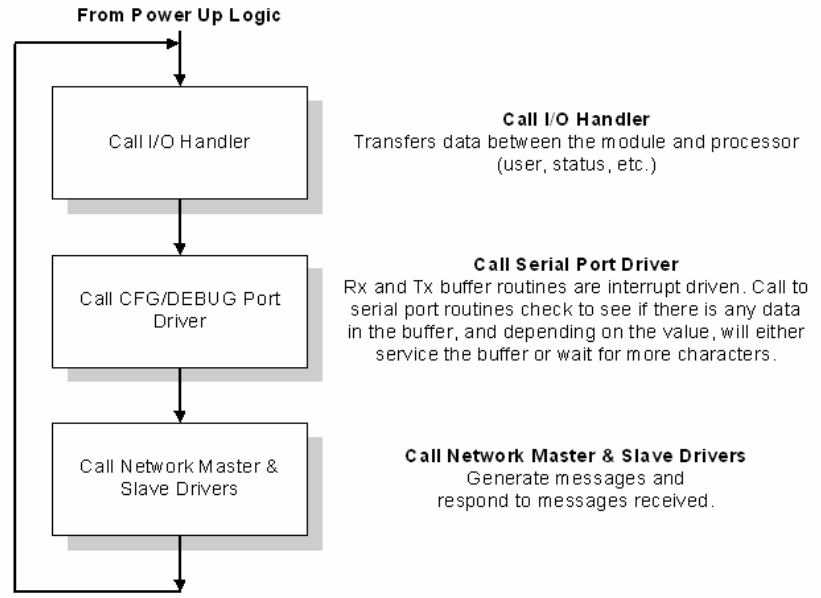
1. Initialize hardware components
2. Initialize CompactLogix or MicroLogix backplane driver
   - Test and Clear all RAM
   - Initialize the serial communication ports
   - Read module configuration from the Compact Flash

3 Initialize Module Register space
4 Enable Slave Driver on selected ports
5 Enable Master Driver on selected ports

After this initialization procedure is complete, the module will begin communicating with other nodes on the network, depending on the configuration.

Main Logic Loop

Upon completing the power up configuration process, the module enters an infinite loop that performs the following functions:



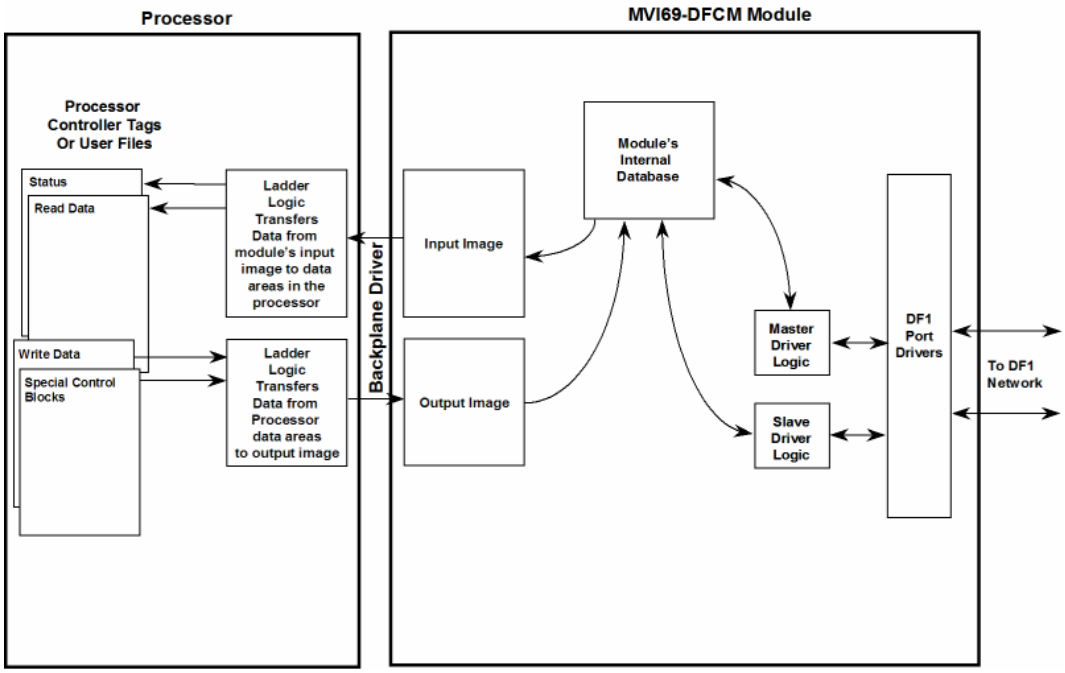
Backplane Data Transfer

The MVI69-DFCM 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 by the user for the module and the communication load on the module. Typical updates are in the range of 2 to 10 milliseconds.

The data is paged between the processor and the module using input and output image blocks. You can configure the size of the blocks using the Block Transfer Size parameter in the configuration file. You can configure blocks of 60, 120, or 240 words of data depending on the number of words allowed for your own application.

This bi-directional transference of data is accomplished by the module filling in data in the module's input image to send to the processor. Data in the input image is placed in the Controller Tags in the processor by the ladder logic. The input image for the module may be set to 62, 122, or 242 words depending on the block transfer size parameter set in the configuration file.

The processor inserts data to the module's output image to transfer to the module. The module's program extracts the data and places it in the module's internal database. The output image for the module may be set to 61, 121, or 241 words depending on the block transfer size parameter set in the configuration file.



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

All data transferred between the module and the processor over the backplane is through the input and output images. Ladder logic must be written in the CompactLogix 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. The following illustration shows the layout of the database:

Module's Internal Database Structure

5000 registers for user data	Register Data	0 4999
3000 words of configuration and status data	Status and Config	5000 7999

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-DFCM 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. The read and write block identification codes in each data block determine the function to be performed or the content of the data block. The block identification codes used by the module are listed below.

Block Range	Descriptions
-1	Status Block
0	Status Block
1 to 999	Read or write data
1000	Event Port 1
2000	Event Port 2
3000 to 3001	Port 1 slave polling control
3002 to 3006	Port 1 slave status
3100 to 3101	Port 2 slave polling control
3102 to 3106	Port 2 slave status
5000 to 5006	Port 1 command control
5100 to 5106	Port 2 command control
9972	Set module time using received time
9973	Pass module time to processor
9998	Warm-boot control block
9999	Cold-boot control block

Each image has a defined structure depending on the data content and the function of the data transfer.

5.2.2 Normal Data Transfer

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

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	1
1	Write Block ID	1
2 to (n+1)	Read Data	n

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 determine the location of where the data will be placed in the CompactLogix or MicroLogix processor controller tag array of module read data. 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 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 three read and two write blocks are used with the application, the sequence will be as follows:

R1W1→R2W2→R3W1→R1W2→R2W1→R3W2→R1W1→

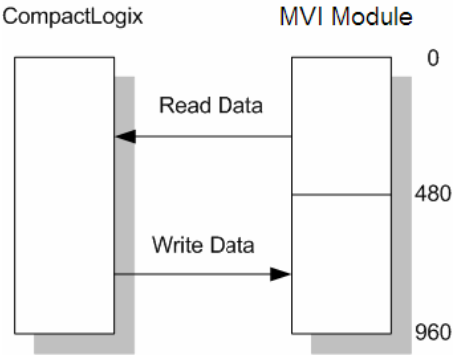
This sequence will continue until interrupted by other write block numbers sent by the controller or by a command request from a node on the DFCM network or operator control through the module's Configuration/Debug port.

The following example shows a typical backplane communication application.

Assume that the backplane parameters are configured as follows:

```
Read Register Start:    0
Read Register Count:  480
Write Register Start: 480
Write Register Count: 480
```

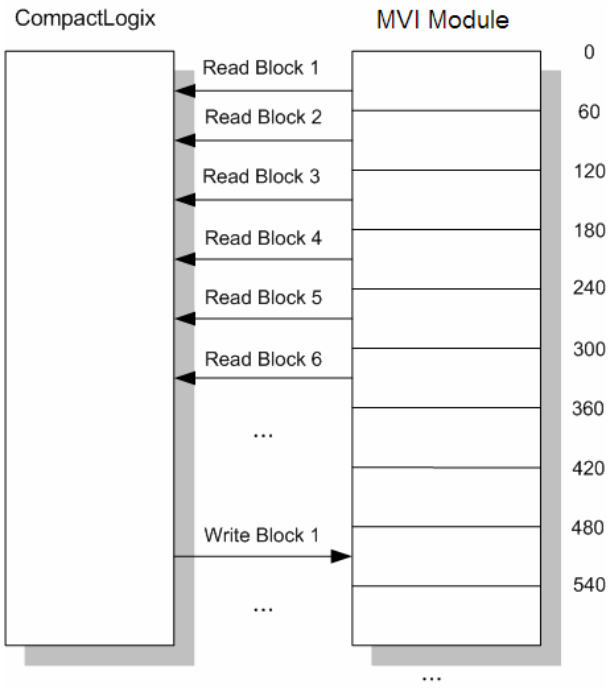
The backplane communication would be configured as follows:



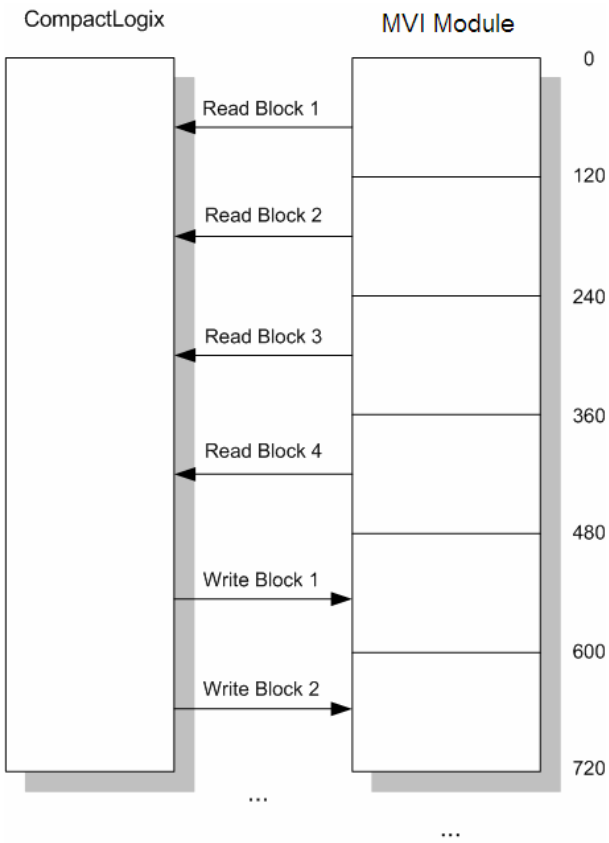
Database address 0 to 479 will be continuously transferred from the module to the processor. Database address 480 to 959 will continuously be transferred from the processor to the module.

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).

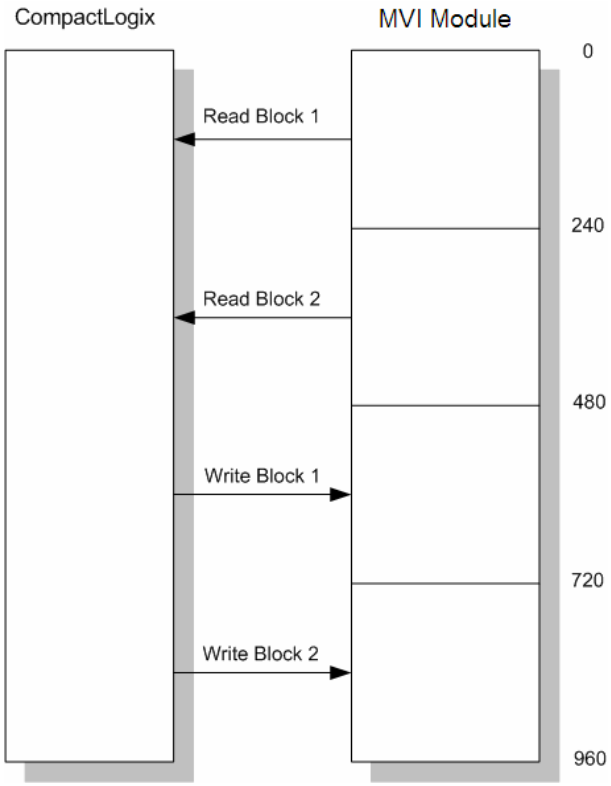
If Block Transfer Size = 60:



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 and source the input (monitored) data to be used by the remote client. The structure of the output image used to transfer this data is shown in the following table.

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 200 words (block offsets 1 to 200) of data.

5.2.3 Special Blocks

Slave Status Blocks

Slave status blocks send status information of each slave device on a master port. Slaves attached to the master port can have one of the following states:

State	Description
0	The slave is inactive and not defined in the command list for the master port.
1	The slave is actively being polled or controlled by the master port and communications is successful.
2	The master port has failed to communicate with the slave device. Communications with the slave is suspended for a user defined period based on the scanning of the command list.
3	Communications with the slave has been disabled by the ladder logic. No communication will occur with the slave until this state is cleared by the ladder logic.

Slaves are defined to the system when the module initializes the master command list. Each slave defined will be set to a state of one in this initial step. If the master port fails to communicate with a slave device (retry count expired on a command), the master will set the state of the slave to a value of 2 in the status table. This suspends communication with the slave device for a user specified scan count (**Error Delay Count** parameter in the configuration file). Each time a command in the list is scanned that has the address of a suspended slave, the delay counter value will be decremented. When the value reaches zero, the slave state will be set to one.

In order to read the slave status table, you should refer to the sample ladder logic. The ladder logic must send a special block to the module to request the data. Each port has a specific set of blocks to request the data as follows:

Block ID	Description
3002	Request status for slaves 0 to 59 for Port 1
3003	Request status for slaves 60 to 119 for Port 1
3004	Request status for slaves 120 to 179 for Port 1
3005	Request status for slaves 180 to 239 for Port 1
3006	Request status for slaves 240 to 255 for Port 1
3102	Request status for slaves 0 to 59 for Port 2
3103	Request status for slaves 60 to 119 for Port 2
3104	Request status for slaves 120 to 179 for Port 2
3105	Request status for slaves 180 to 239 for Port 2
3106	Request status for slaves 240 to 255 for Port 2

The format of these blocks is as shown below:

Write Block: Request Slave Status

Offset	Description	Length
0	3002 to 3006 or 3102 to 3106	1
1 to n	Spare	n

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

The module will recognize the request by receiving the special write block code and respond with a read block with the following format:

Read Block: Read Slave Status

Offset	Description	Length
0	3002 to 3006 or 3102 to 3106	1
1	Write Block ID	1
2 to 61	Slave Poll Status Data	60
62 to n	Spare (if present)	

The sample ladder logic shows how to override the value in the slave status table to disable slaves (state value of 3) by sending a special block of data from the processor to the slave. Port 1 slaves are disabled using block 3000, and Port 2 slaves are disabled using block 3100. Each block contains the slave node addresses to disable. The structure of the block is displayed below:

Write Block: Disable Slaves

Offset	Description	Length
0	3000 or 3100	1
1	Number of slaves in block	1
2 to 61	Slave indexes	60
62 to (n+1)	Spare	

n=120, or 240 (if configured)

The module will respond with a block with the same identification code received and indicate the number of slaves acted on with the block. The format of this response block is displayed below:

Read Block: Disable Slaves

Offset	Description	Length
0	3000 or 3100	1
1	Write Block ID	1
2	Number of slaves processed	1
3 to (n+1)	Spare	

n=60, 120, or 240 (if configured)

The sample ladder logic explains how to override the value in the slave status table to enable the slave (state value of 1) by sending a special block. Port 1 slaves are enabled using block 3001, and Port 2 slaves are enabled using block 3101. Each block contains the slave node addresses to enable. The format of the block is displayed below:

Write Block: Enable Slaves

Offset	Description	Length
0	3001 or 3101	1
1	Number of slaves in block	1
2	Slave indexes	1
3 to n	Spare	

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

The module will respond with a block with the same identification code received and indicate the number of slaves acted on with the block. The format of this response block is displayed below:

Read Block: Enable Slaves

Offset	Description	Length
0	3001 or 3101	1
1	Write Block ID	1
2	Number of slaves processed	1
3 to n	Spare	

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

Important: The slaves are enabled by default. Therefore, this block should only be used after Block 3000 or 3001 to re-enable the slaves.

5.2.4 Command Control Blocks

Command control blocks are special blocks used to control the module or request special data from the module. The current version of the software supports five command control blocks: event command control, command control, transfer time, warm boot and cold boot.

Event Command

Event command control blocks send DFCM commands directly from the ladder logic to one of the master ports. The format for these blocks is displayed below:

Write Block - Event Command

Offset	Description	Length
0	1000 or 2000	1
1	Internal DB Address	1
2	Point Count	1
3	Swap Code	1
4	Node Address	1
5	Function Code	1
6	Parameter #1	1
7	Parameter #2	1
8	Parameter #3	1
9	Parameter #4	1
10 to n	Spare	

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

The block number defines the DFCM port to be considered. Block 1000 commands are directed to Port 1, and block 2000 commands are directed to Port 2. The parameters passed with the block construct the command. The **Internal DB Address** parameter specifies the module's database location to associate with the command. The **Point Count** parameter defines the number of registers for the command. The **Swap Code** changes the word or byte order. The **Node Address** parameter defines the device on the DFCM network to consider. The **Function Code** parameter is one of those defined in the ProSoft DFCM Command Set documentation. The parameter fields in the block should be completed as required by the selected function code. Each command has its own set of parameters. When the block is received, the module will process it and place the command in the command queue. The module will respond to each event command block with a read block with the following format:

Read Block - Event Command

Offset	Description	Length
0	1000 or 2000	1
1	Write Block ID	1
2	0=Fail, 1=Success	1
3 to n	Spare	

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

Word two of the block can be used by the ladder logic to determine if the command was added to the command queue of the module. The command will only fail if the command queue for the port is full (100 commands for each queue) or the command requested is invalid.

Command Control

Command control blocks place commands in the command list into the command queue. Each port has a command queue of up to 100 commands. The module services commands in the queue before the master command list. This gives high priority to commands in the queue. Commands placed in the queue through this mechanism must be defined in the master command list. Under normal command list execution, the module will only execute commands with the Enable parameter set to one or two. If the value is set to zero, the command is skipped. Commands may be placed in the command list with an Enable parameter set to zero. These commands can then be executed using the command control blocks.

One to six commands can be placed in the command queue with a single request. The following table describes the format for this block.

Write Block - Command Control

Offset	Description	Length
0	5001 to 5006 or 5101-5106	1
1	Command index	1
2	Command index	1

Offset	Description	Length
3	Command index	1
4	Command index	1
5	Command index	1
6	Command index	1
7 to n	Spare	

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

Blocks in the range of 5001 to 5006 are used for Port 1, and blocks in the range of 5101 to 5106 are used for Port 2. The last digit in the block code defines the number of commands to process in the block. For example, a block code of 5003 contains 3 command indexes that are to be used with Port 1. The Command index parameters in the block have a range of 0 to 99 and correspond to the master command list entries.

The module responds to a command control block with a block containing the number of commands added to the command queue for the port. The following table describes the format for this block.

Read Block - Command Control

Offset	Description	Length
0	5000 to 5006 or 5100 to 5106	1
1	Write Block ID	1
2	Number of commands added to command queue	1
3 to (n+1)	Spare	

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

Set Module Time Using Processor Time

This block can be used to update the module's internal clock (date and time).

Write Block: Set Module Time

Offset	Description	Length
0	9972	1
1	Year (0 to 9999)	1
2	Month (1 to 12)	1
3	Day (1 to 31)	1
4	Hour (0 to 23)	1
5	Minutes (0 to 59)	1
6	Seconds (0 to 59)	1
7 to n	Spare	

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

Set Module Time Response

Read Block: Set Module Time

Offset	Description	Length
0	9972	1
1	Write Block ID	1
2 to (n+1)	Spare	

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

Get Module Time for Processor Time

Write Block: Get Module Time

Offset	Description	Length
0	9973	1
1 to n	Spare	

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

Read Block: Write Module Time

Offset	Description	Length
0	9973	1
1	Write Block ID	1
2	Year (0 to 9999)	1
3	Month (1 to 12)	1
4	Day (1 to 31)	1
5	Hour (0 to 23)	1
6	Minutes (0 to 59)	1
7	Seconds (0 to 59)	1
8 to n	Spare	

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

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. The structure of the control block is shown below:

Offset	Description	Length
0	9998	1
1 to n	Spare	247

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

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 below:

Offset	Description	Length
0	9999	1
1 to n	Spare	247

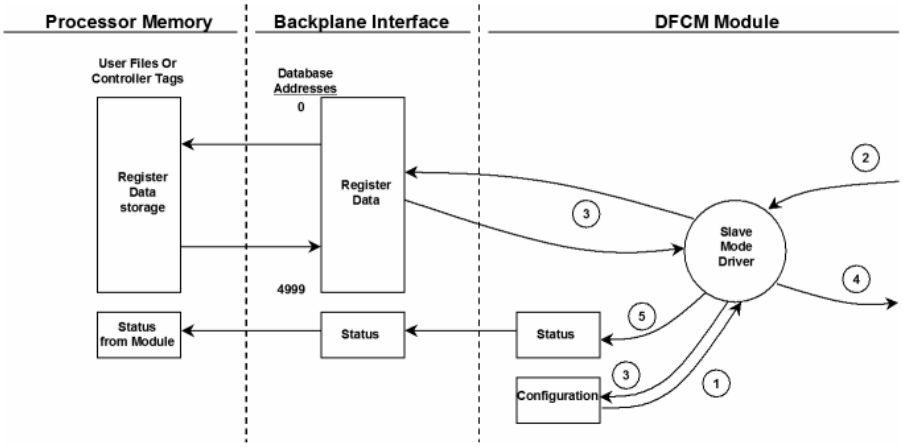
n=60, 120, or 240 depending on what is entered in the Block Transfer Size parameter (refer to the configuration file).

5.2.5 Data Flow between MVI69-DFCM Module and CompactLogix or MicroLogix Processor

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

Slave Driver Mode

The Slave Driver Mode allows the MVI69-DFCM module to respond to data read and write commands issued by a master on the DFCM network. The following flow chart and associated table describe the flow of data into and out of the module.



Step	Description
1	The DFCM slave port driver receives the configuration information from the internal Compact Flash disk. This information configures the serial port and define the slave node characteristics. The module simulates N-files to permit remote access of the database. Each file has a configurable length of 60, 120, or 240-word registers.

Step	Description
2	A Host device, such as the Rockwell Automation PLC or an HMI application issues a read or write command to the module's node address. The port driver qualifies the message before accepting it into the module.
3	After the module accepts the command, the data is immediately transferred to or from the internal database in the module. If the command is a read command, the data is read out of the database and a response message is built. If the command is a write command, the data is written directly into the database and a response message is built.
4	After the data processing has been completed in Step 3, the response is issued to the originating master node.
5	Counters are available in the Status Block that permit the ladder logic program to determine the level of activity of the Slave Driver.

Review the **Module Set Up** section for a complete list of the parameters that must be defined for a slave port. The slave driver supports the following DFCM command set:

Basic Command Set Functions

Function Code	Command	Function	Definition	PLC5	SLC500 & MicroLogix	Power-monitor II	ControlLogix
1	0x00	N/A	Protected Write	X			X
2	0x01	N/A	Unprotected Read	X	X		X
3	0x02	N/A	Protected Bit Write	X			X
4	0x05	N/A	Unprotected Bit Write	X			X
5	0x08	N/A	Unprotected Write	X	X		X

PLC-5 Command Set Functions

Function Code	Command	Function	Definition	PLC5	SLC500 & MicroLogix	Power-monitor II	ControlLogix
100	0x0F	0x00	Word Range Write (Binary Address)	X			X
101	0x0F	0x01	Word Range Read (Binary Address)	X			X
102	0x0F	0x26	Read-Modify-Write (Binary Address)	X			X
150	0x0F	0x00	Word Range Write (ASCII Address)	X			X
151	0x0F	0x01	Word Range Read (ASCII Address)	X			X
152	0x0F	0x26	Read-Modify-Write (ASCII Address)	X			X

SLC-500 Command Set Functions

Function Code	Command	Function	Definition	PLC5	SLC500 & MicroLogix	Power-monitor II	ControlLogix
501	0x0F	0xA1	Protected Typed Logical Read With Two Address Fields		X		X
502	0x0F	0XA2	Protected Typed Logical Read With Three Address Fields		X	X	X
509	0x0F	0XA9	Protected Typed Logical Write With Two Address Fields		X		X

Function Code	Command	Function	Definition	PLC5	SLC500 & MicroLogix	Power-monitor II	ControlLogix
510	0x0F	0XAA	Protected Typed Logical Write With Three Address Fields		X	X	X
511	0x0F	0XAB	Protected Typed Logical Write With Mask (Three Address Fields)		X		X

Each command list record has the same general format. The first part of the record contains the information relating to the communication module and the second part contains information required to interface to the DF1 slave device.

The PLC-5 and SLC-500 command set require the use of files. These files are emulated in the module. The module defines these files each as containing 200-word registers that overlay the internal database. The following table shows the relationship of the files to the user data area of the internal database:

File	→	Database Register
N7:0	→	0
N8:0	→	200
N9:0	→	400
N10:0	→	600
N11:0	→	800
N12:0	→	1000
N13:0	→	1200
N14:0	→	1400
N15:0	→	1600
N16:0	→	1800
N17:0	→	2000
N18:0	→	2200
N18:0	→	2400
N20:0	→	2600
N21:0	→	2800
N22:0	→	3000
N23:0	→	3200
N24:0	→	3400
N25:0	→	3600
N26:0	→	3800
N27:0	→	4000
N28:0	→	4200
N29:0	→	4400
N30:0	→	4600
N31:0	→	4800
N32:0	→	5000

Note: The way these files are emulated depends of the *First File* and *File Size* parameters. The previous example shows using the *First File* parameter set to 7 and the *File Size* parameter set to 200.

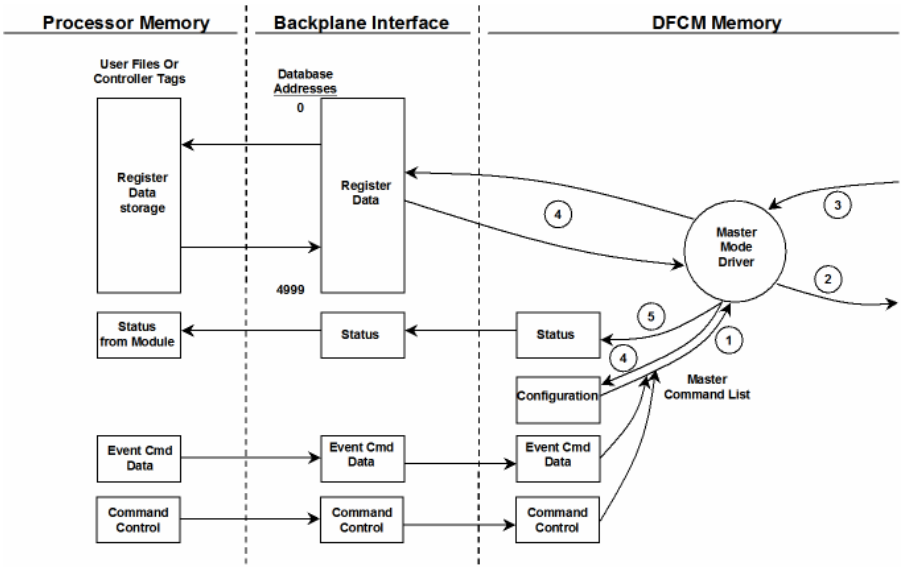
In order to retrieve data from the modules database register 200, the remote master would issue a command using the address N8:0. In order to interface with database base register 405, the remote master would use the address N9:5. The following table outlines the complete file emulation for the module:

Register Range	File Start	File End	Content	Size
0 to 4999	N7:0	N31:199	User Data	5000
5000 to 5009	N32:0	N32:9	Backplane Configuration	10
5010 to 5039	N32:10	N32:39	Port 1 Setup	30
5040 to 5069	N32:40	N32:69	Port 2 Setup	30
5070 to 5199	N32:70	N32:199	Reserved	130
5200 to 6399	N33:0	N38:199	Port 1 Commands	1200
6400 to 7599	N39:0	N44:199	Port 2 Commands	1200
7600 to 7700	N45:0	N45:199	Misc. Status Data	200
7800 to 7999	N46:0	N46:199	Command Control	200
8000 to 9999	N47:0	N56:199	Reserved	2000

All the data in the module is available to a remote host. This permits the host device to remotely configure the module and view the status data.

Master Driver Mode

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



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

The Reference chapter describes the structure and content of each command. Care must be taken in constructing each command in the list for predictable operation of the module. If two commands write to the same internal database address of the module, the results will not be as desired. All commands containing invalid data will be ignored by the module. The following table describes the functions supported by the module and the format of each command:

	Module Information Data ←					→ Device Information Data					
Col #	1	2	3	4	5	6	7	8	9	10	11
Function Code	Enable Code	Internal Address	Poll Interval Time	Count	Swap Code	Node Address	Function Code	Function Parameters			
FC 1	Code	Register	Seconds	Count	Code	Node	1	Word Address			
FC 2	Code	Register	Seconds	Count	Code	Node	2	Word Address			
FC 3	Code	Register	Seconds	Count	0	Node	3	Word Address			
FC 4	Code	Register	Seconds	Count	0	Node	4	Word Address			
FC 5	Code	Register	Seconds	Count	Code	Node	5	Word Address			
FC 100	Code	Register	Seconds	Count	Code	Node	100	File Number	Element	Sub-Element	
FC 101	Code	Register	Seconds	Count	Code	Node	101	File Number	Element	Sub-Element	
FC 102	Code	Register	Seconds	Count	0	Node	102	File Number	Element	Sub-Element	
FC 501	Code	Register	Seconds	Count	Code	Node	501	File Type	File Number	Element	
FC 502	Code	Register	Seconds	Count	Code	Node	502	File Type	File Number	Element	Sub-Element
FC 509	Code	Register	Seconds	Count	Code	Node	509	File Type	File Number	Element	
FC 510	Code	Register	Seconds	Count	Code	Node	510	File Type	File Number	Element	Sub-Element
FC 511	Code	Register	Seconds	Count	0	Node	511	File Type	File Number	Element	Sub-Element

Node Address = Destination Address for Message

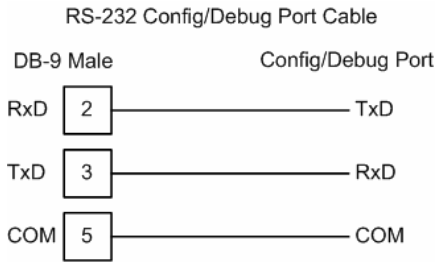
5.3 Cable Connections

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

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

5.3.1 RS-232 Configuration/Debug Port

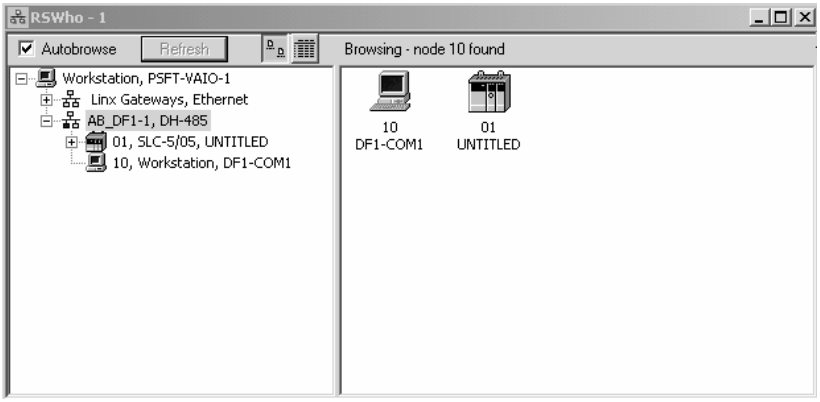
This port is physically an RJ45 connection. An RJ45 to DB-9 adapter cable is included with the module. This port permits a PC based terminal emulation program to view configuration and status data in the module and to control the module. The cable for communications on this port is shown in the following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



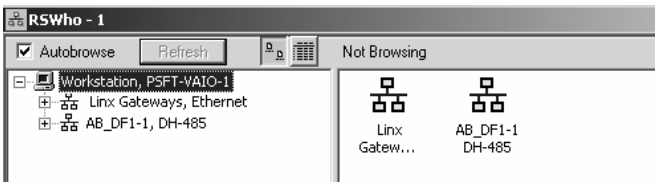
Disabling the RSLinx Driver for the Com Port on the PC

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

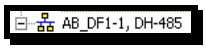
1. Open RSLinx and go to Communications>RSWho
2. Make sure that you are not actively browsing using the driver that you wish to stop. The following shows an actively browsed network:



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

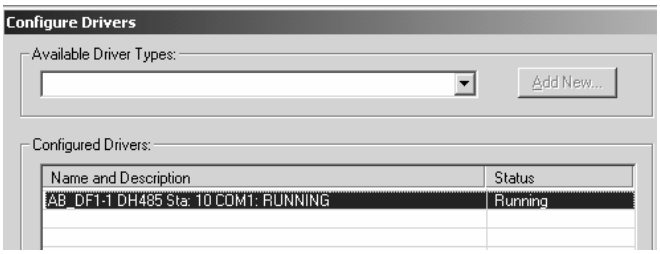


Branches are displayed or hidden by clicking on the ⊞ or the ⊟ icons.

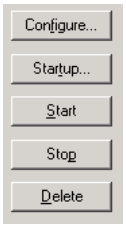


4 When you have verified that the driver is not being browsed, go to

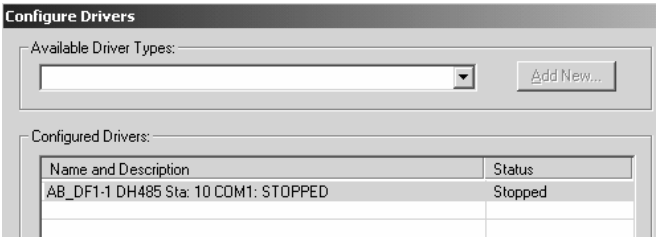
Communications>Configure Drivers
You may see something like this:



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



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

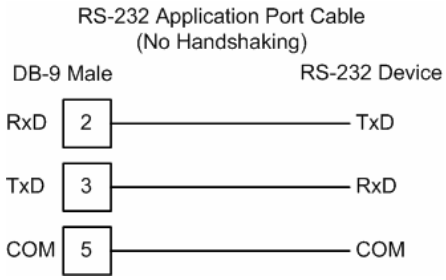


6 Upon seeing this, you may now use that com port to connect to the debug port of the module.

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

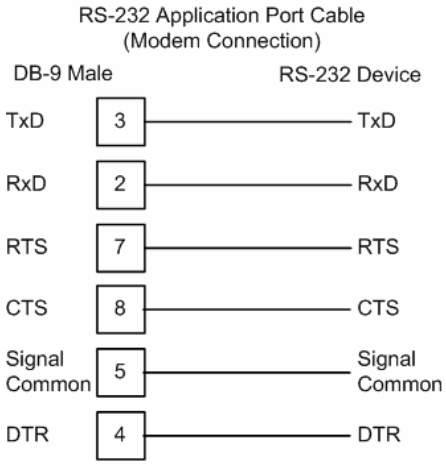
5.3.2 RS-232

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



RS-232: Modem Connection

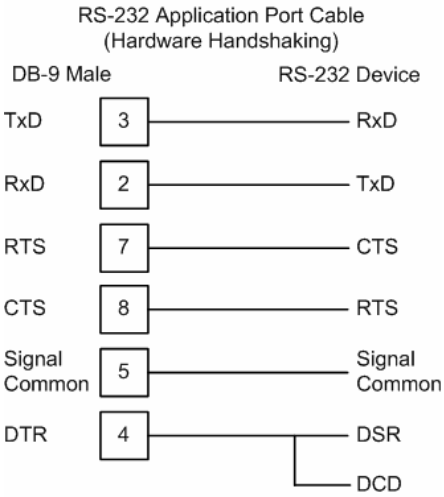
This type of connection is required between the module and a modem or other communication device.



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

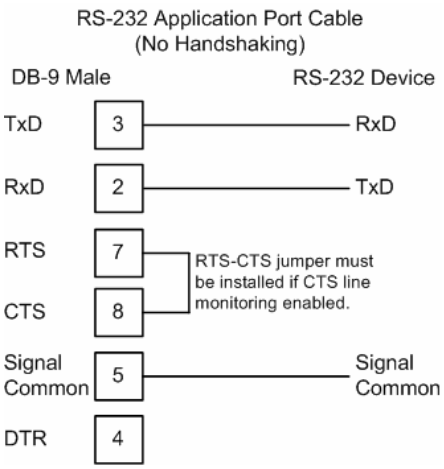
RS-232: Null Modem Connection (Hardware Handshaking)

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



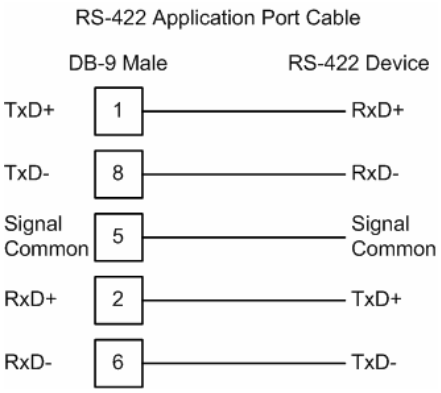
RS-232: Null Modem Connection (No Hardware Handshaking)

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



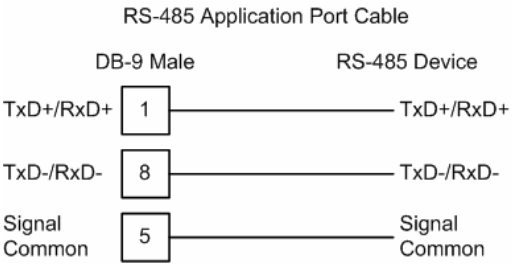
Note: If the port is configured with the "Use CTS Line" set to 'Y', then a jumper is required between the RTS and the CTS line on the module connection.

5.3.3 RS-422



5.3.4 RS-485

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

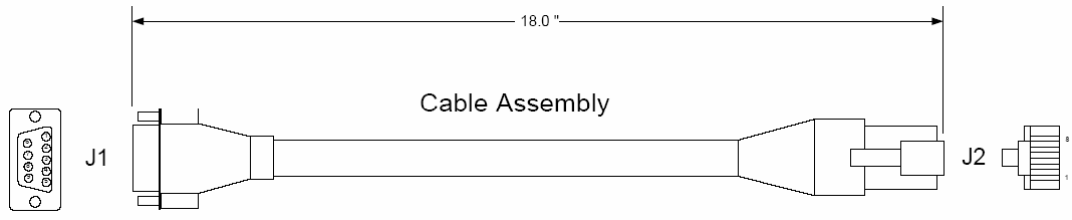


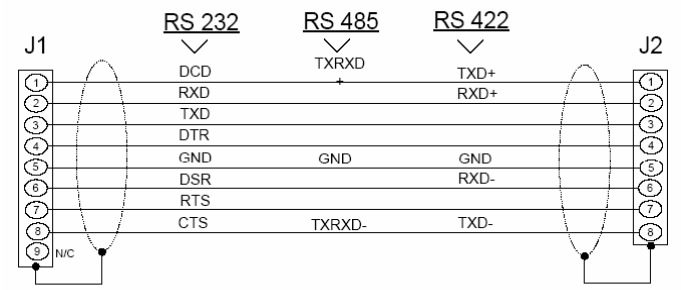
Note: Terminating resistors are generally not required on the RS-485 network, unless you are experiencing communication problems that can be attributed to signal echoes or reflections. In this case, install a 120 ohm terminating resistor on the RS-485 line.

RS-485 and RS-422 Tip

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

5.3.5 DB9 to RJ45 Adaptor (Cable 14)





Wiring Diagram

5.4 DFCM Database Definition

This section contains a listing of the internal database of the MVI69-DFCM module. This information can be used to interface other devices to the data contained in the module.

Register Range	Content	Size
0 to 4999	User Data	5000
5000 to 5009	Backplane Configuration	10
5010 to 5039	Port 1 Setup	30
5040 to 5069	Port 2 Setup	30
5070 to 5199	Reserved	130
5200 to 6399	Port 1 Commands	1200
6400 to 7599	Port 2 Commands	1200
7600 to 7700	Misc. Status Data	200
7800 to 7999	Command Control	200
8000 to 9999	Reserved	2000

The User Data area holds data collected from other nodes on the network (master read commands) or data received from the processor (write blocks). Additionally, this data area is used as a data source for the processor (read blocks) or other nodes on the network (write commands).

Refer to Status Data for descriptions of:

- miscellaneous status data
- configuration data areas
- command control

5.5 Status Data Definition

This section contains a description of the members present in the **DFCMStatus** object. This data is transferred from the module to the processor as part of each read block.

5.5.1 Status Data Block Structure

Offset	Content	Description
7600	Program Scan Count	This value is incremented each time a complete program cycle occurs in the module.
7601 to 7602	Product Code	These two registers contain the product code of "DFCM"
7603 to 7604	Product Version	These two registers contain the product version for the current running software.
7605 to 7606	Operating System	These two registers contain the month and year values for the program operating system.
7607 to 7608	Run Number	These two registers contain the run number value for the currently running software.
7609	Port 1 Command List Requests	This field contains the number of requests made from this port to slave devices on the network.
7610	Port 1 Command List Response	This field contains the number of slave response messages received on the port.
7611	Port 1 Command List Errors	This field contains the number of command errors processed on the port. These errors could be due to a bad response or command.
7612	Port 1 Requests	This field contains the total number of messages sent out of the port.
7613	Port 1 Responses	This field contains the total number of messages received on the port.
7614	Port 1 Errors Sent	This field contains the total number of message errors sent out of the port.
7615	Port 1 Errors Received	This field contains the total number of message errors received on the port.
7616	Port 2 Command List Requests	This field contains the number of requests made from this port to slave devices on the network.
7617	Port 2 Command List Response	This field contains the number of slave response messages received on the port.
7618	Port 2 Command List Errors	This field contains the number of command errors processed on the port. These errors could be due to a bad response or command.
7619	Port 2 Requests	This field contains the total number of messages sent out the port.
7620	Port 2 Responses	This field contains the total number of messages received on the port.
7621	Port 2 Errors Sent	This field contains the total number of message errors sent out of the port.

Offset	Content	Description
7622	Port 2 Errors Received	This field contains the total number of message errors received on the port.
7623	Read Block Count	This field contains the total number of read blocks transferred from the module to the processor.
7624	Write Block Count	This field contains the total number of write blocks transferred from the processor to the module.
7625	Parse Block Count	This field contains the total number of blocks successfully parsed that were received from the processor.
7626	Command Event Block Count	This field contains the total number of command event blocks received from the processor.
7627	Command Block Count	This field contains the total number of command blocks received from the processor.
7628	Error Block Count	This field contains the total number of block errors recognized by the module.
7629	Port 1 Current Error	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.
7630	Port 1 Last Error	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.
7631	Port 2 Current Error	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.
7632	Port 2 Last Error	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.

5.6 DFCM Command Control

Command Control data is received from other nodes on the network that can control the MVI69-DFCM module. Specific values are written to regions of this block to control the module. Currently, the module is programmed to handle the receipt of the following requests: warm boot and cold boot.

The remote node controls the module by writing one of the following values to register 7800:

Value	Description
9998	Warm boot the module
9999	Cold boot the module

5.7 Error Codes

The module error codes are listed in this section. Error codes returned from the command list process are stored in the command list error memory region. A word is allocated for each command in the memory area. The error codes are formatted in the word as follows: The least-significant byte of the word contains the extended status code and the most-significant byte contains the status code.

Use the error codes returned for each command in the list to determine the success or failure of the command. If the command fails, use the error code to determine the cause of failure.

Note: The Module Specific error codes (not DF1 compliant) are returned from within the module and never returned from an attached DF1 slave device. These are error codes that are part of the DF1 protocol or are extended codes unique to this module. The standard DF1 error codes can be found in the DF1 Protocol and Command Set Reference Manual (Publication 1770-6.5.16) from Rockwell Automation. The most common errors for the DF1 protocol are shown in the following tables:

5.7.1 Local STS Error Codes

Code (Int)	Code (Hex)	Description
0	0x0000	Success, no error
256	0x0100	DST node is out of buffer space
512	0x0200	Cannot guarantee delivery (Link Layer)
768	0x0300	Duplicate token holder detected
1024	0x0400	Local port is disconnected
1280	0x0500	Application layer timed out waiting for response
1536	0x0600	Duplicate node detected
1792	0x0700	Station is offline
2048	0x0800	Hardware fault

5.7.2 Remote STS Error Codes

Code (Int)	Code (Hex)	Description
0	0x0000	Success, no error
4096	0x1000	Illegal command or format
8192	0x2000	Host has a problem and will not communicate
12288	0x3000	Remote node host is missing, disconnected or shut down
16384	0x4000	Host could not complete function due to hardware fault
20480	0x5000	Addressing problem or memory protect rungs
24576	0x6000	Function not allowed due to command protection selection
26872	0x7000	Processor is in Program mode
-32768	0x8000	Compatibility mode file missing or communication zone problem
-28672	0x9000	Remote node cannot buffer command
-24576	0xA000	Wait ACK (1775-KA buffer full)
-20480	0xB000	Remote node problem due to download

Code (Int)	Code (Hex)	Description
-16384	0xC000	Wait ACK (1775-KA buffer full)
-12288	0xD000	Not used
-8192	0xE000	Not used
	0xF0nn	Error code in the EXT STS byte (nn contains EXT error code)

5.7.3 Errors When EXT STS Is Present

Code (Int)	Code (Hex)	Description
-4096	0xF000	Not used
-4095	0xF001	A field has an illegal value
-4094	0xF002	Less levels specified in address than minimum for any address
-4093	0xF003	More levels specified in address than system supports
-4092	0xF004	Symbol not found
-4091	0xF005	Symbol is of improper format
-4090	0xF006	Address does not point to something usable
-4089	0xF007	File is wrong size
-4088	0xF008	Cannot complete request
-4087	0xF009	Data or file is too large
-4086	0xF00A	Transaction size plus word address is too large
-4085	0xF00B	Access denied, improper privilege
-4084	0xF00C	Condition cannot be generated - resource is not available
-4083	0xF00D	Condition already exists - resource is already available
-4082	0xF00E	Command cannot be executed
-4081	0xF00F	Histogram overflow
-4080	0xF010	No access
-4079	0xF011	Illegal data type
-4078	0xF012	Invalid parameter or invalid data
-4077	0xF013	Address reference exists to deleted area
-4076	0xF014	Command execution failure for unknown reason
-4075	0xF015	Data conversion error
-4074	0xF016	Scanner not able to communicate with 1771 rack adapter
-4073	0xF017	Type mismatch
-4072	0xF018	1171 module response was not valid
-4071	0xF019	Duplicate label
-4070	0xF01A	File is open; another node owns it
-4069	0xF01B	Another node is the program owner
-4068	0xF01C	Reserved
-4067	0xF01D	Reserved
-4066	0xF01E	Data table element protection violation
-4065	0xF01F	Temporary internal problem

5.7.4 Module Specific Error (not DFCM Compliant)

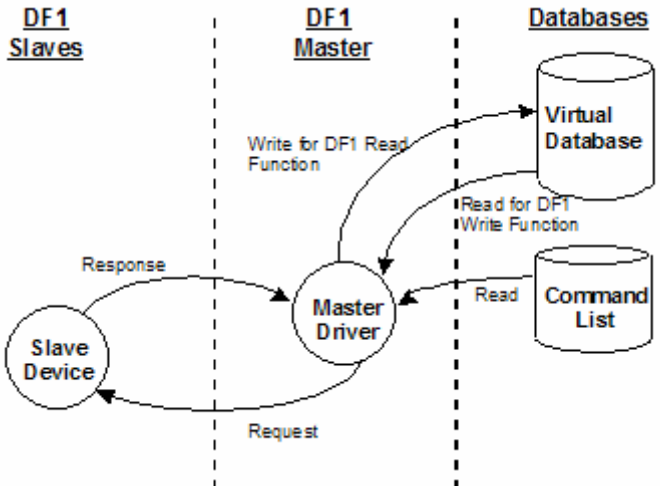
Code (Int)	Code (Hex)	Description
-1	0xFFFF	CTS modem control line not set before transmit
-2	0xFFFE	Timeout while transmitting message
-10	0xFFF6	Timeout waiting for DLE-ACK after request
-11	0xFFF5	Timeout waiting for response after request
-12	0xFFF4	Reply data does not match requested byte count
-20	0xFFEC	DLE-NAK received after request
-21	0xFFEB	DLE-NAK sent after response

5.8 DF1 Command Set For ProSoft Technology Communication Modules

5.8.1 Introduction

This document contains a complete description of the command set required to communicate with DF1 protocol devices using a ProSoft communication module. ProSoft communication modules that contain a virtual DF1 master device use this command set to control and monitor data in DF1 protocol devices. These include Rockwell Automation PLC, SLC, MicroLogix and ControlLogix controllers and field devices supporting the DF1 protocol. ProSoft supports the DF1 protocol on both the serial and network interface. The network interface requires the use of the port service address 0xAF12 as specified in the ControlNet Specification. Rockwell Automation supports this feature in the ControlLogix 5550, PLC5 xx/E and SLC 5/05 processors.

The ProSoft modules contain a virtual database that is defined by the user. This database is used as the source for write commands and the destination for read commands issued on the virtual DF1 master devices. The module interfaces data contained in remote DF1 slave devices to the virtual database using the DF1 master. User commands are issued out of the DF1 master from a command list. These commands gather or control data in the DF1 slave devices. The following illustration shows the relationships discussed above:



Each command issued from the DF1 master contains a field that indicates the location in the virtual database to be associated with the command. Care must be taken when designing a system to be sure the read and write data regions for the database do not overlap for a single device. The read area of one device can overlap the write section of another device to transfer the data from one slave device to another.

5.8.2 Command Function Codes

This section describes DFCM commands to be configured by the user.

	Module Information Data			←	→	Device Information Data				
1	2	3	4	5	6	7	8	9	10	11
Enable Code	Internal Address	Poll Interval Time	Count	Swap Code	Node Address	Function Code	Function Parameters			

Function Code #1 - Protected Write (Basic Command Set)

Column	Parameter	Description	Parameter
1	Enable/Type Word	0=Disabled, 1=Continuous and 2=Conditional.	
2	Virtual Database Address	This parameter defines the database address of the first data point to be associated with the command.	
3	Poll Interval	Minimum time in tenths of a second to wait before polling with this command.	
4	Count	Number of data word values to be considered by the function.	
5	Swap Type Code	Swap type code for command: 0=None, 1=Swap words, 2=Swap words & bytes and 3=swap bytes in each word.	
6	Node Address	Address of unit to reach on the data highway.	
7	Function Code = 1	Protected Write Function	
8	Word Address	Word address where to start the write operation.	P1
9 to 11	Not Used	These fields are not used by the command. Values entered in these columns will be ignored.	P2 to P4

This function writes one or more words of data into a limited area of the slave device. This function should work on the following devices: 1774-PLC, PLC-2, PLC-3, PLC-5 and PLC-5/250.

Function Code #2 - Unprotected Read (Basic Command Set)

Column	Parameter	Description	Parameter
1	Enable/Type Word	0=Disabled and 1=Continuous.	
2	Virtual Database Address	This parameter defines the database address of the first data point to be associated with the command.	
3	Poll Interval	Minimum time in tenths of a second to wait before polling with this command.	
4	Count	Number of data word values to be considered by the function.	
5	Swap Type Code	Swap type code for command: 0=None, 1=Swap words, 2=Swap words & bytes and 3=swap bytes in each word.	

Column	Parameter	Description	Parameter
6	Node Address	Address of unit to reach on the data highway.	
7	Function Code = 2	Unprotected Read Function	
8	Word Address	Word address where to start the read operation.	P1
9 to 11	Not Used	These fields are not used by the command. Values entered in these columns will be ignored.	P2 to P4

This function reads one or more words of data from the PLC memory. This function should work on the following devices: 1774-PLC, PLC-2, PLC-3, PLC-5, SLC 500, SLC 5/03, SLC 5/04 and MicroLogix 1000.

Function Code #3 - Protected Bit Write (Basic Command Set)

Column	Parameter	Description	Parameter
1	Enable/Type Word	0=Disabled, 1=Continuous and 2=Conditional.	
	Virtual Database Address	This parameter defines the database address for the data to be associated with the command. The address defined represents a register address and not a bit address. This function will update one or more words of data as defined by the count parameter.	
3	Poll Interval	Minimum time in tenths of a second to wait before polling with this command.	
4	Count	Number of data word values to be considered by the function.	
5	Swap Type Code	Swap type code for command: Always zero (0).	
6	Node Address	Address of unit to reach on the data highway.	
7	Function Code = 3	Protected Bit Write Function	
8	Word Address	Word address where to start the write operation.	P1
9 to 11	Not Used	These fields are not used by the command. Values entered in these columns will be ignored.	P2 to P4

This function sets or resets individual bits within a limited area of the PLC data table. This function should work on the following devices: 1774-PLC, PLC-2, PLC-3, PLC-5 and PLC-5/250.

Function Code #4 - Unprotected Bit Write (Basic Command Set)

Column	Parameter	Description	Parameter
1	Enable/Type Word	0=Disabled, 1=Continuous and 2=Conditional.	
2	Virtual Database Address	This parameter defines the database address for the data to be associated with the command. The address defined represents a register address and not a bit address. This function will update one or more words of data as defined by the count parameter.	
3	Poll Interval	Minimum time in tenths of a second to wait before polling with this command.	
4	Count	Number of data word values to be considered by the function.	
5	Swap Type Code	Swap type code for command: Always zero (0).	
6	Node Address	Address of unit to reach on the data highway.	
7	Function Code = 4	Unprotected Bit Write Function	

Column	Parameter	Description	Parameter
8	Word Address	Word address where to start the write operation.	P1
9 to 11	Not Used	These fields are not used by the command. Values entered in these columns will be ignored.	P2 to P4

This function sets or resets individual bits within a limited area of the PLC data table. This function should work on the following devices: 1774-PLC, PLC-2, PLC-3 and PLC-5.

Function Code #5 - Unprotected Write (Basic Command Set)

Column	Parameter	Description	Parameter
1	Enable/Type Word	0=Disabled, 1=Continuous and 2=Conditional.	
2	Virtual Database Address	This parameter defines the database address of the first data point to be associated with the command.	
3	Poll Interval	Minimum time in tenths of a second to wait before polling with this command.	
4	Count	Number of data word values to be considered by the function.	
5	Swap Type Code	Swap type code for command: 0=None, 1=Swap words, 2=Swap words & bytes and 3=swap bytes in each word.	
6	Node Address	Address of unit to reach on the data highway.	
7	Function Code = 5	Unprotected Write Function	
8	Word Address	Word address where to start the write operation.	P1
9 to 11	Not Used	These fields are not used by the command. Values entered in these columns will be ignored.	P2 to P4

This function writes one or more words of data to the PLC memory. This function should work on the following devices: 1774-PLC, PLC-2, PLC-3, PLC-5, SLC 500, SLC 5/03, SLC 5/04 and MicroLogix 1000.

Function Code #100 - Word Range Write (PLC-5 Command) (Binary Address)

Column	Parameter	Description	Parameter
1	Enable/Type Word	0=Disabled, 1=Continuous and 2=Conditional.	
2	Virtual Database Address	This parameter defines the database address of the first data point to be associated with the command.	
3	Poll Interval	Minimum time in tenths of a second to wait before polling with this command.	
4	Count	Number of data word values to be considered by the function.	
5	Swap Type Code	Swap type code for command: 0=None, 1=Swap words, 2=Swap words & bytes and 3=swap bytes in each word.	
6	Node Address	Address of unit to reach on the data highway.	
7	Function Code = 100	Word Range Write Command.	
8	File Number	PLC-5 file number to be associated with the command. If a value of -1 is entered for the parameter, the field will not be used in the command, and the default file will be used.	P1

Column	Parameter	Description	Parameter
9	Element Number	The parameter defines the element in the file where write operation will start. If a value of -1 is entered for the parameter, the field will not be used in the command, and the default element will be used.	P2
10	Sub-Element Number	This parameter defines the sub-element to be used with the command. Refer to the AB documentation for a list of valid sub- element codes. If the value is set to -1, the default sub-element number will be used.	P3
11	Not Used	This field is not used by the command. Values entered in this column will be ignored.	P4

This function writes one or more words of data to a PLC data table. This function should work on the following devices: PLC-5.

Function Code #101 - Word Range Read (PLC-5 Command) (Binary Address)

Column	Parameter	Description	Parameter
1	Enable/Type Word	0=Disabled and 1=Continuous.	
2	Virtual Database Address	This parameter defines the database address of the first data point to be associated with the command.	
3	Poll Interval	Minimum time in tenths of a second to wait before polling with this command.	
4	Count	Number of data word values to be considered by the function.	
5	Swap Type Code	Swap type code for command: 0=None, 1=Swap words, 2=Swap words & bytes and 3=swap bytes in each word.	
6	Node Address	Address of unit to reach on the data highway.	
7	Function Code = 101	Word Range Write Command.	
8	File Number	PLC-5 file number to be associated with the command. If a value of -1 is entered for the parameter, the field will not be used in the command, and the default file will be used.	P1
9	Element Number	The parameter defines the element in the file where write operation will start. If a value of -1 is entered for the parameter, the field will not be used in the command, and the default element will be used.	P2
10	Sub-Element Number	This parameter defines the sub-element to be used with the command. Refer to the AB documentation for a list of valid sub- element codes. If the value is set to -1, the default sub-element number will be used.	P3
11	Not Used	This field is not used by the command. Values entered in this column will be ignored.	P4

This function reads one or more words of data from a PLC data table. This function should work on the following devices: PLC-5.

Function Code #102 - Read-Modify-Write (PLC-5 Command) (Binary Address)

Column	Parameter	Description	Parameter
1	Enable/Type Word	0=Disabled, 1=Continuous and 2=Conditional.	
2	Virtual Database Address	This parameter defines the database address for the data to be associated with the command.	

Column	Parameter	Description	Parameter
3	Poll Interval	Minimum number of seconds to wait before polling with this command.	
4	Count	Number of data word values to be considered by the function.	
5	Swap Type Code	Swap type code for command: Always zero (0).	
6	Node Address	Address of unit to reach on the data highway.	
7	Function Code = 102	Read-Modify-Write Command.	
8	File Number	PLC-5 file number to be associated with the command. If a value of -1 is entered for the parameter, the field will not be used in the command, and the default file will be used.	P1
9	Element Number	The parameter defines the element in the file where write operation will start. If a value of -1 is entered for the parameter, the field will not be used in the command, and the default element will be used.	P2
10	Sub-Element Number	This parameter defines the sub-element to be used with the command. Refer to the AB documentation for a list of valid sub-element codes. If the value is set to -1, the default sub-element number will be used.	P3
11	Not Used	This field is not used by the command. Values entered in this column will be ignored.	P4

This function writes one or more words of data to a PLC data table. This function should work on the following devices: PLC-5. The command constructed contains an AND mask and an OR mask. Values in the AND mask have the following definitions: 0=Reset and 1=Leave the Same. Values in the OR mask have the following definitions: 0=Leave the Same and 1=Set. The module is responsible for setting the mask values to correctly construct the message from the virtual database values.

Function Code #150 - Word Range Write (PLC-5 Command) (ASCII Address)

Column	Parameter	Description	Parameter
1	Enable/Type Word	0=Disabled, 1=Continuous and 2=Conditional.	
2	Virtual Database Address	This parameter defines the database address of the first data point to be associated with the command.	
3	Poll Interval	Minimum time in tenths of a second to wait before polling with this command.	
4	Count	Number of data word values to be considered by the function.	
5	Swap Type Code	Swap type code for command: 0=None, 1=Swap words, 2=Swap words & bytes and 3=swap bytes in each word.	
6	Node Address	Address of unit to reach on the data highway.	
7	Function Code = 150	Word Range Write Command.	
8	File String	PLC-5 address as specified as an ASCII string. For example, N10:300.	P1
9 to 11	Not Used	These fields are not used by the command. Values entered in these columns will be ignored.	P2 to P4

This function writes one or more words of data to a PLC data table. This function should work on the following devices: PLC-5.

Function Code #151 - Word Range Read (PLC-5 Command) (ASCII Address)

Column	Parameter	Description	Parameter
1	Enable/Type Word	0=Disabled and 1=Continuous.	
2	Virtual Database Address	This parameter defines the database address of the first data point to be associated with the command.	
3	Poll Interval	Minimum time in tenths of a second to wait before polling with this command.	
4	Count	Number of data word values to be considered by the function.	
5	Swap Type Code	Swap type code for command: 0=None, 1=Swap words, 2=Swap words & bytes and 3=swap bytes in each word.	
6	Node Address	Address of unit to reach on the data highway.	
7	Function Code = 151	Word Range Read Command.	
8	File String	PLC-5 address as specified as an ASCII string. For example, N10:300.	P1
9 to 11	Not Used	These fields are not used by the command. Values entered in these columns will be ignored.	P2 to P4

This function reads one or more words of data from a PLC data table. This function should work on the following devices: PLC-5.

Function Code #152 - Read-Modify-Write (PLC-5 Command) (ASCII Address)

Column	Parameter	Description	Parameter
1	Enable/Type Word	0=Disabled, 1=Continuous and 2=Conditional.	
2	Virtual Database Address	This parameter defines the database address for the data to be associated with the command. The first database register is used as the AND mask for the command, and the second is used for the OR mask. Values in the AND mask have the following definitions: 0=Reset and 1=Leave the Same. Values in the OR mask have the following definitions: 0=Leave the Same and 1=Set.	
3	Poll Interval	Minimum time in tenths of a second to wait before polling with this command.	
4	Count	Number of data word values to be considered by the function.	
5	Swap Type Code	Swap type code for command: Always zero (0).	
6	Node Address	Address of unit to reach on the data highway.	
7	Function Code = 152	Read-Modify-Write Command.	
8	File String	PLC-5 address as specified as an ASCII string. For example, N10:300.	P1
9 to 11	Not Used	These fields are not used by the command. Values entered in these columns will be ignored.	P2 to P4

This function writes one or more words of data to a PLC data table. This function should work on the following devices: PLC-5. The command constructed contains an AND mask and an OR mask. Values in the AND mask have the following definitions: 0=Reset and 1=Leave the Same. Values in the OR mask have the following definitions: 0=Leave the Same and 1=Set. The module is responsible for setting the mask values to correctly construct the message from the virtual database values.

Function Code #501 - Protected Typed Logical Read (Two Address Fields)

Column	Parameter	Description	Parameter
1	Enable/Type Word	0=Disabled and 1=Continuous.	
2	Virtual Database Address	This parameter defines the database address of the first data point to be associated with the command.	
3	Poll Interval	Minimum number of seconds to wait before polling with this command.	
4	Count	Number of data word values to be considered by the function.	
5	Swap Type Code	Swap type code for command: 0=None, 1=Swap words, 2=Swap words & bytes and 3=swap bytes in each word.	
6	Node Address	Address of unit to reach on the data highway.	
7	Function Code = 501	Logical Read Command	
8	File Type	SLC file type letter as used in file name string. Valid values for the system are N, S, F, A, ….	P1
9	File Number	SLC file number to be associated with the command.	P2
10	Element Number	The parameter defines the element in the file where write operation will start.	P3
11	Not Used	This field is not used by the command. Values entered in this column will be ignored.	P4

This function reads one or more words of data from a PLC data table.

Function Code #502 - Protected Typed Logical Read (Three Address Fields)

Column	Parameter	Description	Parameter
1	Enable/Type Word	0=Disabled and 1=Continuous.	
2	Virtual Database Address	This parameter defines the database address of the first data point to be associated with the command.	
3	Poll Interval	Minimum number of seconds to wait before polling with this command.	
4	Count	Number of data word values to be considered by the function.	
5	Swap Type Code	Swap type code for command: 0=None, 1=Swap words, 2=Swap words & bytes and 3=swap bytes in each word.	
6	Node Address	Address of unit to reach on the data highway.	
7	Function Code = 502	Logical Read Command	
8	File Type	SLC file type letter as used in file name string. Valid values for the system are N, S, F, A, ….	P1
9	File Number	SLC file number to be associated with the command.	P2
10	Element Number	The parameter defines the element in the file where write operation will start.	P3
11	Sub-Element Number	This parameter defines the sub-element to be used with the command. Refer to the AB documentation for a list of valid sub-element codes.	P4

This function reads one or more words of data from a PLC data table. This function should work on the following devices: SLC 500, SLC 5/03 and SLC 5/04.

Function Code #509 - Protected Typed Logical Write (Two Address Fields)

Column	Parameter	Description	Parameter
1	Enable/Type Word	0=Disabled, 1=Continuous and 2=Conditional.	
2	Virtual Database Address	This parameter defines the database address of the first data point to be associated with the command.	
3	Poll Interval	Minimum time in tenths of a second to wait before polling with this command.	
4	Count	Number of data word values to be considered by the function.	
5	Swap Type Code	Swap type code for command: 0=None, 1=Swap words, 2=Swap words & bytes and 3=swap bytes in each word.	
6	Node Address	Address of unit to reach on the data highway.	
7	Function Code = 509	Logical Write Command	
8	File Type	SLC file type letter as used in file name string. Valid values for the system are N, S, F, A, ….	P1
9	File Number	SLC file number to be associated with the command.	P2
10	Element Number	The parameter defines the element in the file where write operation will start.	P3
11	Not Used	This field is not used by the command. Values entered in this column will be ignored.	P4

This function writes one or more words of data to a PLC data table.

Function Code #510 - Protected Typed Logical Write (Three Address Fields)

Column	Parameter	Description	Parameter
1	Enable/Type Word	0=Disabled, 1=Continuous and 2=Conditional.	
2	Virtual Database Address	This parameter defines the database address of the first data point to be associated with the command.	
3	Poll Interval	Minimum time in tenths of a second to wait before polling with this command.	
4	Count	Number of data word values to be considered by the function.	
5	Swap Type Code	Swap type code for command: 0=None, 1=Swap words, 2=Swap words & bytes and 3=swap bytes in each word.	
6	Node Address	Address of unit to reach on the data highway.	
7	Function Code = 510	Logical Write Command	
8	File Type	SLC file type letter as used in file name string. Valid values for the system are N, S, F, A, ….	P1
9	File Number	SLC file number to be associated with the command.	P2
10	Element Number	The parameter defines the element in the file where write operation will start.	P3
11	Sub-Element Number	This parameter defines the sub-element to be used with the command. Refer to the AB documentation for a list of valid sub-element codes.	P4

This function writes one or more words of data to a PLC data table. This function should work on the following devices: SLC 500, SLC 5/03 and SLC 5/04.

Function Code #511 - Protected Typed Logical Write with Mask (Three Address Fields)

Column	Parameter	Description	Parameter
1	Enable/Type Word	0=Disabled, 1=Continuous and 2=Conditional.	
2	Virtual Database Address	This parameter defines the database address of the data to be associated with the command. The first word of data contains the bit mask and the second word contains the data.	
3	Poll Interval	Minimum time in tenths of a second to wait before polling with this command.	
4	Count	Number of data word values to be considered by the function.	
5	Swap Type Code	Swap type code for command: Always zero (0).	
6	Node Address	Address of unit to reach on the data highway.	
7	Function Code = 511	Logical Write with mask	
8	File Type	SLC file type letter as used in file name string. Valid values for the system are N, S, F, A, ….	P1
9	File Number	SLC file number to be associated with the command.	P2
10	Element Number	The parameter defines the element in the file where write operation will start.	P3
11	Sub-Element Number	This parameter defines the sub-element to be used with the command. Refer to the AB documentation for a list of valid sub-element codes.	P4

This function writes one or more words of data from a PLC data table controlling individual bits in the table. The bit mask used for the command is 0xFFFF. This provides direct manipulation of the data in the device with the internal data of the module. The function requires that all data associated with the command use the same mask.

5.8.3 PLC-5 Processor Specifics

This section contains information specific to the PLC-5 processor with relation to the DF1 command set. The commands specific to the PLC-5 processor contain a sub-element code field. This field selects a sub-element field in a complex data table. For example, to obtain the current accumulated value for a counter or timer, the sub-element field should be set to 2. The tables below show the sub-element codes for PLC-5 complex data tables.

PLC-5 Sub-Element Codes

Timer / Counter

Code	Description
0	Control
1	Preset
2	Accumulated

Control

Code	Description
0	Control
1	Length
2	Position

PD*

Code	Description
0	Control
2	SP
4	Kp
6	Ki
8	Kd
26	PV

*All PD values are floating point values, so they are two words long.

BT

Code	Description
0	Control
1	RLEN
2	DLEN
3	Data file #
4	Element #
5	Rack/Grp/Slot

MG

Code	Description
0	Control
1	Error
2	RLEN
3	DLEN

5.8.4 SLC Processor Specifics

This section contains information specific to the SLC processor based family when used with the DF1 command set. The SLC processor commands support a file type field entered as a single character to denote the data table to interface with in the command. The following table defines the relationship of the file types accepted by the module and the SLC file types:

SLC File Types

File Type	File Type Command Code	Description
S	83	Status
B	66	Bit
T	84	Timer
C	67	Counter
R	82	Control
N	78	Integer
F	70	Floating-point
Z	90	String
A	65	ASCII

The File Type Command Code is the ASCII character code value of the File Type letter. This is the value to enter into the "File Type" parameter of the DF1 Command configurations in the data tables in the ladder logic.

Additionally, the SLC specific functions (502, 510 and 511) support a sub-element field. This field selects a sub-element field in a complex data table. For example, to obtain the current accumulated value for a counter or timer, the sub-element field should be set to 2.

5.8.5 MicroLogix Processor Specifics

This section contains information specific to the MicroLogix processor based family when used with the DF1 command set. The MicroLogix processor commands support a file type field entered as a single character to denote the data table to interface with in the command. This field is the same as that used for a SLC processor. The following table defines the relationship of the file types accepted by the module and the SLC file types:

SLC File Types

File Type	File Type Command Code	Description
S	83	Status
B	66	Bit
T	84	Timer
C	67	Counter
R	82	Control
N	78	Integer
F	70	Floating-point
Z	90	String
A	65	ASCII

The File Type Command Code is the ASCII character code value of the File Type letter. This is the value to enter into the "File Type" parameter of the DF1 Command configurations in the data tables in the ladder logic.

Additionally, the SLC specific functions (502, 510 and 511) support a sub-element field. This field selects a sub-element field in a complex data table. For example, to obtain the current accumulated value for a counter or timer, the sub-element field should be set to 2.

5.8.6 ControlLogix Processor Specifics

This section contains information specific to the ControlLogix processor when used with the DF1 command set. The current implementation of the DF1 command set does not use functions that can directly interface with the ControlLogix Tag Database. In order to interface with this database, the table-mapping feature provided by RSLogix 5000 must be used. The software permits the assignment of ControlLogix Tag Arrays to virtual PLC 5 data tables. The ProSoft module using the PLC 5 command set defined in this document can then reach this controller data.

5.9 DF1 Command List Form

	Module Information Data ←					→ Device Information Data					
Column #	1	2	3	4	5	6	7	8	9	10	11
Function Code	Enable Code	Internal Address	Poll Interval Time	Count	Swap Code	Node Address	Function Code	Function Parameters			

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 107). 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 110)". 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 110) 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

T

U

V

W

Y